and shown the output signal **OutSig** coincides with the output signal of the last inversion unit $\mathbf{400_r}$ in the cascade, this is not at all limitative: the output signal **OutSig** might in fact also be tapped off the output of any other inverter in the cascade.

The circuit of **Figure 4** resembles an LFSR in the so-called Fibonacci configuration (a so-called Fibonacci LFSR); however, differently from the Fibonacci LFSR, the circuit of **Figure 4** has asynchronous inverting delay units instead of synchronous delay units implemented as synchronously clocked D-type flips-flops. Importantly, the logic inversions operated by the inverting delay units make the circuit of **Figure 4** work in a mathematically different way from the Fibonacci LFSR, even if it is assumed to be operated synchronously.

The feedback paths from the outputs of the inversion units in the cascade to the input of the first inversion unit are preferably chosen in order to satisfy the above-mentioned requirement that no fixed points exist in the corresponding state-transition function.

For a mathematical analysis, it is convenient to represent the feedback coefficients by means of a binary polynomial *f(x)*:

$$f(x) = \sum_{i=0}^{r} f_i x^i$$

wherein x^i corresponds to the generic output $\mathbf{405_i}$, with $\mathbf{i} = 0$ to r, the summation is binary (*i.e.*, it is a modulo-2 addition, *i.e.*, an XOR operation), and $\mathbf{f_0} = \mathbf{f_r} = 1$.

Analysis of the state-transition function corresponding to the logic circuit with feedback shown in **Figure 4** is conducted by assuming that the circuit is operated in a synchronous manner, *i.e.,* that the logic inversion units $\mathbf{400_1}$, $\mathbf{400_2}$, ..., $\mathbf{400_r}$ operate on the basis of a timing signal, which (*e.g.* in simulations) can be implemented by introducing a synchronously clocked D-type flip-flop after each logic inversion unit in the cascade.

A mathematical analysis reveals that the circuit of **Figure 4** has no fixed points in the respective state-transition function if and only if the following conditions (in the following simply referred to as "condition C_F") are satisfied:

$$f(x) = (1+x)h(x)$$

and

$$h(1) = 1.$$

The above condition C_F equivalently means that the polynomial *f(x)* is divisible by (1+x), *i.e,.* that $f(1)=0$, and that the quotient polynomial $h(x)$ is not divisible by (1+x). It is

WAVELYNX EX. 1003, Pg. 543
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

emphasized that from the above condition C_F there follows that the degree r of the polynomial $f(x)$ can be odd or even, but necessarily $r \neq 2$.

Further mathematical analysis shows that, provided that the above condition C_F is satisfied, the state-transition diagram of the circuit of **Figure 4**, assumed to be operated in a synchronous manner, contains one short cycle, particularly a cycle of length 2, composed of the two states of "all-zeros" and "all-ones", and a plurality of longer cycles, whose number and lengths depend on the polynomial $h(x)$, in a way known in the theory of linear recurring sequences.

It thus follows that the said short cycle of length 2 is meta-stable in the asynchronous operation, because in practice the transition between these two states is extremely unlikely to happen, since when the outputs of all the inversion units quasi-simultaneously reach the intermediate voltage value, half-way between the voltage values corresponding to binary logic states of "1" and "0", the further transition in the same direction is not sustained any more and hence becomes unpredictable. As a consequence, if by chance the state sequence ever enters this cycle, then it can spend only a very short time in it.

Preferably, the polynomial $h(x)$ is chosen in such a way so as to be a primitive polynomial (*i.e.*, an irreducible polynomial with maximal period equal to 2^{r-1}-1 as in this case, according to the theory of linear recurring sequences, there is only one long cycle, of length 2^r-2. This ensures good pseudo-randomness properties of the oscillating output signal **OutSig**, including a long period and good statistical properties.

In addition to having good pseudo-randomness properties, the oscillating output signal **OutSig** also exhibits randomness properties, as discussed in connection with the more general structure of logic circuit with feedback **315** of **Figure 3**, all the consideration made for that general structure being directly applicable to the specific logic circuit of **Figure 4**. In particular, randomness properties derive from the unpredictable variations in the delay of the inverter logic gates, which get propagated and enhanced through feedback, the possible internal meta-stability events, and the frequent meta-stability events in the sampling D-type flip-flop.

For the logic circuit of **Figure 4**, the frequency of the oscillating output signal **OutSig** is determined by the total delay of the logic inversion units in the shortest feedback loop, that is, in the loop identified by the smallest index i such that $\mathbf{f_i=1}$. Thus, in order to obtain a high-speed oscillating output signal **OutSig**, it is preferable that such an

WAVELYNX EX. 1003, Pg. 544
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

index i is small, *e.g.*, equal to 1 or 2 and that the delay introduced by the individual inverting units is small (in practice, this means that the (odd) number of elementary inverter logic gates forming the logic inversion units is relatively small).

For the logic circuit of **Figure 4**, the internal meta-stability events, which as such may give rise to unpredictable state changes, are mainly characteristic of the internal states containing a constant string of length at least two that starts immediately after the point where the feedback signal is fed back into the cascade (that is to say, at the beginning of the cascade). This is because the state transition where the said constant string should be transformed into its binary complement may not be sustainable by the feedback signal. More precisely, similarly as for the meta-stability of the short cycle of length 2, this happens if the feedback signal has the same binary value and undergoes the same change as the said constant string.

The oscillating output signal **OutSig** can thus have a high speed and possess a significant amount of randomness not only at the digital level, but also at the analog level: the output signal **OutSig**, more than being a two-valued (voltage or current) purely digital signal, appears to be like a sort of analog noise. Altogether, a high-speed raw random binary sequence **RRBS** that contains a high amount of entropy per bit is thus obtained at the output of the sampling unit **325**. Not only are the obtained randomness properties robust with respect to changes in the statistics of the random variations in the delay of the logic inversion units, but also, in the unlikely event of the total absence of these variations, the resulting raw random binary sequence will at worst possess good pseudo-randomness properties.

A second possible practical implementation of the random binary sequence generator **105** is shown in **Figure 5**; similarly to the circuit of **Figure 4**, a plurality of, *e.g.*, *s*, logic inversion units $\mathbf{500_s}$, $\mathbf{500_{(s-1)}}$, ..., $\mathbf{500_2}$, $\mathbf{500_1}$, are connected in cascade one to another. The output $\mathbf{505_1}$ of the last inversion unit $\mathbf{500_1}$ in the cascade is directly fed back and forms the input $\mathbf{I_s}$ to the first inversion unit $\mathbf{500_s}$ (for convenience, the indexes are now given in the reverse order when compared with the configuration depicted in **Figure 4**).

In addition to the direct feedback from the output $\mathbf{505_1}$ of the last inversion unit $\mathbf{500_1}$ to the input $\mathbf{I_s}$ of the first inversion unit $\mathbf{500_s}$, a plurality of feedback paths from the output $\mathbf{505_1}$ of the last inversion unit $\mathbf{500_1}$ to the inputs of the inversion units $\mathbf{500_{(s-1)}}$, ..., $\mathbf{500_2}$, $\mathbf{500_1}$ subsequent (downstream) to the first one in the cascade are provided. In

WAVELYNX EX. 1003, Pg. 545
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

particular, the output signal 505_1 of the last inversion unit 500_1, in the shown example corresponding to the output signal OutSig of the raw binary sequence generator, is combined, according to XOR operations (performed for example by two-inputs XOR gates $510_{(s-1)}$, ..., 510_2, 510_1), with the output signal 505_s, ..., 505_3, 505_2 of the generic inversion unit 500_s, ..., 500_3, 500_2 so as to produce the input signal to the subsequent inversion unit $500_{(s-1)}$, ..., 500_2, 500_1 , respectively. Similarly to the circuit of **Figure 4**, which ones of the output signals 505_s, ..., 505_3, 505_2 are combined in XOR with the output signal 505_1 to obtain the input signal to the subsequent inversion unit is determined by the open/closed state of a plurality of switches $515_{(s-1)}$, ..., 515_2, 515_1; said open/closed state is specified by the assertion/deassertion of open/closed state control signals g_{s-1}, ..., g_2, g_1: when the generic control signal g_i ($i = 1, 2,..., (s-1)$) is a logic "0", the corresponding switch 515_i is open and the output signal 505_{i+1} is not XOR-ed with the output signal 505_1 of the last inversion unit 500_1, being instead directly fed as an input to the subsequent inversion unit 500_i; on the contrary, when the control signal g_i is a logic "1", the corresponding switch 515_i is closed and the output signal 505_{i+1} is XOR-ed with the output signal 505_1 of the last inversion unit 500_1. In other words, the open/closed state of the switches 515_1, 515_2, ..., $515_{(s-1)}$, as specified by open/closed state control signals g_1, g_2, ..., g_{s-1}, determines the type and number of feedback paths from the output of the last inversion unit to the input of the internal inversion units (but the first) in the cascade.

Also in this case, the generic inversion unit 500_i may, for example, be formed by an elementary inverter, *e.g.,* a CMOS inverter, or, alternatively, by an odd number of elementary inverters in a series; this latter construction may be in some circumstances preferable, because in this way the designer may individually adjust the transmission delay of the single inversion unit. As in the previously described invention embodiment, the specific construction of the inversion units is however not limitative to the present invention: in general, any circuit acting as an inverting delay unit can be exploited.

It is observed that, similarly to the embodiment of **Figure 4**, it is not necessary to materially implement (s-1) selectively activatable feedback paths, the activation of which is individually determined by the open/closed state of the respective switch: the open/closed state control signals g_1, g_2, ..., g_{s-1} may in fact be regarded as binary-valued coefficients, each coefficient specifying, depending on the respective value, if the associated feedback path is materially provided or not. However, if all the possible feedback paths are provided for, selectively activatable by controlling the respective

WAVELYNX EX. 1003, Pg. 546
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

switches, the circuit results to be configurable, and greater flexibility is ensured.

Furthermore, it is observed that also in this case the output signal **OutSig** might in principle be tapped off the output of any other inversion unit in the cascade, not necessarily from the last one.

The circuit of **Figure 5** resembles an LFSR in the so-called Galois configuration (so-called Galois LFSR); however, differently from the Galois LFSR, the circuit of **Figure 5** has asynchronous inverting delay units instead of synchronous delay units implemented as synchronously clocked D-type flips-flops. Importantly, the logic inversions operated by the inverting delay units make the circuit of **Figure 5** work in a mathematically different way from the Galois LFSR, even if it is assumed to be operated synchronously.

The feedback paths from the output of the last inversion unit in the cascade to the input of the other inversion units are preferably chosen in order to satisfy the above-mentioned requirement that no fixed points exist in the corresponding state-transition function.

Adopting again the binary polynomial representation of the feedback coefficients in the circuit of **Figure 5**, it is obtained:

$$g(x) = \sum_{i=0}^{s} g_i x^i$$

wherein x^i corresponds to the generic output $\mathbf{505_i}$, with $\mathbf{i} = 0$ to s, the summation is binary (*i.e.*, it is a modulo-2 addition, *i.e.*, an XOR operation), and $\mathbf{g_0} = \mathbf{g_s} = 1$.

Analysis of the state-transition function corresponding to the logic circuit with feedback shown in **Figure 5** is conducted by assuming that the circuit is operated in a synchronous manner, *i.e.,* that the logic inversion units $\mathbf{500_1}$, $\mathbf{500_2}$, ..., $\mathbf{500_s}$ operate on the basis of a timing signal, which can be implemented (*e.g.*, in simulations) by introducing a synchronously clocked D-type flip-flop after each logic inversion unit in the cascade.

A mathematical analysis reveals that the circuit of **Figure 5** has no fixed point in the respective state-transition function if and only if the following conditions (in the following simply referred to as "condition C_G") are satisfied:

$$g(1) = 0$$

and

s is odd.

The above condition C_G equivalently means that the polynomial $g(x)$ is equal to

WAVELYNX EX. 1003, Pg. 547
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

$g(x) = (1+x)k(x)$, for a binary polynomial $k(x)$, and that the degree of the polynomial $g(x)$ is odd.

Further mathematical analysis shows that, provided that the above condition C_G is satisfied, the state-transition diagram of the circuit of **Figure 5**, still assumed to be operated in a synchronous manner, contains exactly one short cycle of length 2 if and only if the further condition $k(1)=1$ is satisfied.

The number and the lengths of the longer cycles depend on the polynomial $k(x)$, in a way known in the theory of linear recurring sequences. Preferably, the polynomial $k(x)$ is chosen to be a primitive polynomial, in which case there is only one long cycle of length 2^s-2. This ensures good pseudo randomness properties of the oscillating output signal **OS**, including a long period and good statistical properties.

By further mathematical analysis, it can be derived that the short cycle of length 2 is composed of the two states of the following form. Let the s inverters be grouped into 2n+1 groups, separated by the effective feedback connections, that is, by the 2n nonzero feedback coefficients (the number has to be even as $g(1) = 0$), and let the inverter output bits define the state vector of the logic circuit with feedback **315**. Then, the two states take the forms $c_{n+1}^* \Lambda_n c_n^* \Lambda_{n-1} \ldots \Lambda_2 c_2^* \Lambda_1 c_1^*$ and $\neg c_{n+1}^* \Lambda_n \neg c_n^* \Lambda_{n-1} \ldots \Lambda_2 \neg c_2^* \Lambda_1 \neg c_1^*$, respectively, where Λ_i is an alternating string, c_i^* is the constant string obtained by repeating a constant c_i, and $\neg c_i^*$ is the constant string defined by the binary complement of c_i. All the strings are uniquely determined by the following rule: if $c_1=0$, then the last bit of A_i equals c_i and the first bit of A_i equals the binary complement of c_{i-1}. It thus follows that the transition between these two states is meta-stable unless all the constant strings have the same odd length, for example, equal to one. This is because the transition between the constituent constant strings and their binary complements is generally meta-stable (for reasons similar to those already explained in the context of the Fibonacci configuration). In particular, to enhance the meta-stability, when the output signal **OutSig** is taken from the last inverter in the cascade, it is thus recommended that the length of the first constant string, c_1^*, should be even, for example, equal to two (which is equivalent to $g_1=0$ and $g_2=1$).

Also for the logic circuit of **Figure 5**, the frequency of the oscillating output signal **OutSig** is determined by the total delay of the logic inversion units in the shortest feedback loop, that is, in the loop identified by the smallest index i such that $g_i=1$. Thus, in order to obtain a high-speed output signal **OutSig**, it is preferable that such an index i

WAVELYNX EX. 1003, Pg. 548
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

is small, *e.g.*, equal to 2 (provided that g_1=0, as suggested above) and that the delay introduced by the individual inverting units is small (in practice, this means that the (odd) number of elementary inverter logic gates forming the logic inversion units is relatively small). In the present embodiment, the inverter delay may be even smaller than in the case of the circuit of **Figure 4**, because in that case the feedback signal is produced with a delay, due to a series of XOR gates connected together, which is not present in the circuit of **Figure 5**.

As for the logic circuit of **Figure 4**, in addition to having good pseudo-randomness properties, the oscillating output signal **OutSig**, also exhibits randomness properties, for the reasons discussed in connection with the general logic circuit with feedback **315** of **Figure 3**, which are all directly applicable to the specific logic circuit of **Figure 5** as well. The arguments include the unpredictable variations in the delay of the inverter logic gates, which get propagated and enhanced through feedback, possible internal meta-stability events, and frequent meta-stability events in the sampling D-type flip-flop. Due to different feedback configurations, the induced randomness properties of the logic circuits shown in **Figures 4** and **5** may be somewhat different, but the main arguments remain the same.

For the logic circuit of **Figure 5**, the internal meta-stability events, which as such may give rise to unpredictable state changes, are mainly characteristic of the internal states containing one or more constant strings of length at least 2 that start immediately after the points where the feedback signal is fed back into the cascade so that each such string and the value of the feedback signal are the same and undergo the same change as in the described meta-stable cycle of length 2. This is because the corresponding state transition where the said constant strings should be transformed into their binary complements is then not sustainable by the feedback signal. In this regard, the circuit of **Figure 5** is preferable to that of **Figure 4**, because it contains more than just one such point and hence contains a larger number of states prone to meta-stability.

It is again emphasized that while in the circuit configuration of **Figure 5** an odd number of logic inversion units is required, in the circuit configuration of **Figure 4** the number of logic inversion units can be either odd or even.

A further possible implementation of the random binary sequence generator **105** is depicted schematically in **Figure 6**. A first and a second oscillators **600a** and **600b** are exploited, one, *e.g.*, the first oscillator **600a**, being of the type shown in **Figure 4** without

WAVELYNX EX. 1003, Pg. 549
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

the sampling unit, the other, *e.g.*, the second oscillator **600b**, being of the type shown in **Figure 5** without the sampling unit. An output sequence **OutSiga** of the first oscillator **600a** and an output sequence **OutSigb** of the second oscillator **600b** are combined together, for example, they are bitwise XOR-ed by a two-input XOR gate **605**. The resulting output sequence **OutSig** is fed to the sampling unit **325**, operated by a clock signal **Ck**, producing the raw binary sequence **RRBS**.

Assuming that the first and second oscillators **600a** and **600b** are mutually independent, the structure of **Figure 6** further enhances the randomness properties and increases their robustness with respect to changes in the statistics of the random variations in the delay of the logic inversion units. For example, in the unlikely event of the total absence of these variations in one of the oscillators (for example, due to the technological reasons), the resulting raw random binary sequence will at worst possess randomness properties resulting from the other oscillator.

Preferably, the lengths of the two oscillators **600a** and **600b**, in terms of the numbers of constituent logic inversion units minus one, should be mutually prime, because in this way the period of the corresponding pseudorandom sequence is maximized and the interlocking or coupling effect is minimized. In particular, the lengths can differ only by one, where the even length corresponds to the oscillator **600a**.

Merely by way of example, in a practical implementation of the solution presented in **Figure 4,** the binary polynomial $f(x)$ may be:

$$f(x) = 1 + x^2 + x^4 + x^5 + x^{12} + x^{13} + x^{15} + x^{16} x^{18} + x^{20}$$

with the generic logic inversion unit $\mathbf{400_1}$, $\mathbf{400_2}$, ..., $\mathbf{400_r}$ being composed of five elementary inverters. The corresponding long cycle then has length 2^{20}-2. For the solution proposed in **Figure 5**, the binary polynomial may be:

$$g(x) = 1 + x^2 + x^3 + x^7 + x^{16} + x^{17} + x^{19} + x^{21}$$

with the generic logic inversion units $\mathbf{500_1}$, $\mathbf{500_2}$, ..., $\mathbf{500_s}$ being composed of three elementary inverters. The corresponding long cycle then has length 2^{21}-2. The long cycle for the XOR combination of the two oscillators then has length $2(2^{19}-1)(2^{20}-1)$. The ring oscillator forming the clock generator **130** that generates the sampling clock **Ck** is composed of eleven elementary inverters.

It is observed that other combinations of the oscillator structures of **Figures 4** and **5** are also possible; more generally, a raw binary sequence generator may be constructed combining one or more oscillator structures of the type shown in **Figure 4** and/or one or

WAVELYNX EX. 1003, Pg. 550
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

more oscillator structures of the type shown in **Figure 5**.

So far, it has been assumed that the PRNG in the post-processing circuit **110** is implemented by means of a Galois LFSR. In an alternative embodiment of the present invention, the PRNG may comprise an LFSR in the so-called Fibonacci configuration (shortly, a Fibonacci LFSR). In brief, a Fibonacci LFSR comprises a cascade of D-type flip-flops, wherein the first flip-flop receives at its input a feedback signal that is formed by XOR-ing the outputs of at least one of the flip-flops in the cascade.

The security level of the PRNG **110** (where by security level there is meant the level of difficulty in predicting the PRNG output sequence **PRBS**) is predominantly determined by the length r of the LFSR. For post-processing applications of a raw random binary sequence produced by a physical source of randomness, the LFSR length is preferably in the range $64 \leq r \leq 128$. Also, the security is affected by the number of non-zero feedback coefficients, that is, by the weight of the feedback polynomial $t(x)$, which is suggested to be relatively close to r/2. An additional condition that is preferably satisfied is that the weight of the polynomial multiples of the feedback polynomial $t(x)$ of moderately large degree is not small (*e.g.*, for a polynomial of degree 64, the weight of its polynomial multiples of degrees smaller than 1000 is not smaller than 7); this avoids so-called "correlation attacks", and also avoids distinguishing output sequence from a purely random sequence.

The security level can be further increased, by using more than one LFSR, either Galois or Fibonacci, possibly implementing the irregular clocking scheme described in the foregoing, and combining their respective outputs by a suitable Boolean function. In particular, different LSFRs might be clocked in either the self-clock-control, or in a mutual-clock-control manner, with the clock-control signal generated in one LSFR being used to establish the clock-control sequence for the other LFSR, and *vice versa*.

Thanks to the present invention, output binary sequences of good statistical properties, particularly purely random output sequences, can be generated starting from raw random binary sequences having statistical weaknesses; in particular, if the raw random binary sequence is a high-speed sequence, the obtained output sequence has a smaller, but sufficiently high speed as well. Alternatively, the obtained output sequence can have a higher speed than the raw random binary sequence, in which case the obtained output sequence is not purely random, but is computationally indistinguishable from a purely random sequence.

WAVELYNX EX. 1003, Pg. 551
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

A random number generator according to an embodiment of the present invention can be rather easily implemented in the practice by using only logic gates in digital semiconductor technology, and is suitable for integration in both FPGA (Field Programmable Gate Array) and ASIC (Application Specific Integrated Circuit) technologies; in particular, standard digital library units can be directly exploited, and the result is practically independent of the fabrication technology.

The random number generator according to the present invention can be advantageously exploited in many applications, in particular, for providing random bits to be used in computer games, in cryptographic algorithms and protocols for secret keys, and for countermeasures against side-channel attacks on microelectronic devices implementing cryptographic algorithms such as the integrated chip cards.

Although the present invention has been disclosed and described by way of some embodiments, it is apparent to those skilled in the art that several modifications to the described embodiments, as well as other embodiments of the present invention are possible without departing from scope thereof as defined in the appended claims.

For example, an adaptive irregular clocking scheme may be adopted, whereby the speed ratio $(\delta 1 + \delta 2)/2$ is caused to be inversely proportional to an estimate of the entropy rate of the input raw random binary sequence **RRBS;** such an estimate may for example be computed on-line, using a statistic such as the bias of individual bits and/or the autocorrelation function on relatively short segments of the input raw random binary sequence **RRBS**.

WAVELYNX EX. 1003, Pg. 552
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

CLAIMS

1. An apparatus for generating random data, comprising a raw random sequence source (**105**) adapted to generate a raw random sequence (**RRBS**) and a digital post-processor (**110**) adapted to process said raw random sequence to generate said random data (**PRBS**), wherein said digital post-processor (**110**) comprises a synchronous finite-state machine (**115**) having at least one input adapted to repeatedly receive a current value of said raw random sequence and at least one output to provide a current output value (**OS**) depending on previous values of said raw random sequence.

2. The apparatus according to claim 1, wherein said digital post-processor (**110**) further comprises a logic circuit (**XOR2**) suitable to receive said current output value (**OS**) and said current value of said raw random sequence and to provide a current output data (**OS'**) as a function of said current output value and said current value of said raw random sequence, the random data being related to the sequence of said current output data.

3. The apparatus according to claim 1 or 2, further comprising a decimation circuit (**145**) suitable to receive said output data and to generate said random data, said finite state machine being suitable to be clocked by a first clock signal (**Ck**), said decimation circuit operating according to a second clock signal that has a period that is k times the period of the first clock signal, k having a value variable in time, said value being calculated each time a random data is generated.

4. The apparatus according to claim 3, in which said value of k at a given time depends on the current state of the finite-state machine.

5. The apparatus according to claim 2, wherein said logic circuit comprises a latin-square type logic circuit.

6. The apparatus according to claim 5, wherein said latin-square type logic circuit comprises an XOR gate.

WAVELYNX EX. 1003, Pg. 553
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

7. The apparatus according to claim 4, wherein said finite-state machine implements a state-transition function and comprises a state-transition sequential circuit (**130**) having a set of inputs (**In**) and a set of outputs (**Out**), said set of outputs being suitable to provide current state values of said state-transition function, said set of inputs including said at least one input of said finite-state machine, said apparatus comprising a further logic circuit (**XOR1**) suitable to receive at least one of said current state values and said current value of said raw random sequence and to provide an input value for said state-transition sequential circuit.

8. The apparatus according to any one of the preceding claims, in which the finite-state machine is such that the output binary sequence is purely random if the input raw random sequence is purely random.

9. The apparatus according to any one of the preceding claims, in which the finite-state machine is such that the output binary sequence is computationally unpredictable even if the raw random sequence is known.

10. The apparatus according to any one of the preceding claims, in which the finite-state machine is such that a change in the first input bit of the input binary sequence induces a computationally unpredictable change in the subsequent output binary sequence.

11. The apparatus according to claim 7, in which said finite-state machine includes a plurality of synchronous delay logic units (**Dr, ..., D1**) connected in a cascade one to another and clocked by the first clock signal, the plurality of synchronous delay logic units comprising a first (**Dr**) and a last (**D1**) synchronous delay logic units in the cascade, said set of outputs comprising an output (**O1**) of the last synchronous delay logic unit and said set of inputs comprising an input (**Ir**) of the first synchronous delay logic unit, said finite-state machine comprising a first feedback path from the output of the last synchronous delay logic unit to the input of the first synchronous delay logic unit, said plurality of synchronous delay logic units further comprising at least one intermediate synchronous delay logic unit (**D(r-1), ..., D2**) in the cascade.

WAVELYNX EX. 1003, Pg. 554
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

12. The apparatus according to claim 11, in which said finite-state machine comprises at least one second feedback path from said output of the last synchronous delay logic unit to an input of a respective synchronous delay logic unit in the cascade, said respective synchronous delay logic unit being chosen in the group consisting of the at least one intermediate synchronous delay logic unit and the last synchronous delay logic unit in the cascade.

13. The apparatus according to claim 12, in which the finite-state machine further comprises an XOR logic combination (X_{r-1}, ..., X_1) of said at least one second feedback path and the output of a preceding synchronous delay logic unit that precedes said respective synchronous delay logic unit where said at least one second feedback path terminates.

14. The apparatus according to claim 13, in which said at lest one second feedback path includes a plurality of feedback paths.

15. The apparatus according to claim 14, in which the finite-state machine has an output signal, said output signal being extracted from a first internal state signal of the finite-state machine according to an irregular clocking.

16. The apparatus according to claim 15, in which the irregular clocking is controlled by a clocking control signal (**CkC**), said clocking control signal being a logic combination of at least a second and a third internal state signals of the finite-state machine.

17. The apparatus according to claim 17, in which the output signal is a logic combination of the first internal state signal with at least a fourth and a fifth internal state values of the finite-state machine.

18. The apparatus according to any one of the preceding claims, in which said raw random sequence source comprises at least one logic circuit (**315**) corresponding to an associated finite-state machine having a state-transition function including states arranged to form cycles of states,

WAVELYNX EX. 1003, Pg. 555
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

wherein:

the at least one logic circuit has a set of logic circuit inputs (InS) and a set of logic circuit outputs (OutS) fed back to said logic circuit inputs;

the associated finite-state machine is autonomous and asynchronous;

the state-transition function is void of loops; and

any of said cycles of states has either a minimum length equal to three states, in case the cycle is stable, or a minimum length of two states, in case the cycle is meta-stable.

19. A method for generating random data, comprising generating a raw random sequence (RRBS) and post-processing the raw random sequence so as to obtain said random data, wherein said post-processing the raw random binary sequence comprises continuously feeding the raw random binary sequence as an input to a synchronous state-transition sequential circuit.

20. The method according to claim 19, in which said feeding the raw random binary sequence comprises:

feeding a current value of the raw random binary sequence to the state-transition sequential circuit of a finite-state machine by using a combination logic and

clocking the state-transition logic circuit to produce a next internal state value, said next internal state value depending on the current raw random binary sequence value and on the internal state value at a current time.

21. The method according to claim 20, further comprising obtaining a combined output value by combining a value related to said current internal state value with said current value of the raw random binary sequence.

22. The method according to claim 21, further comprising obtaining said random data by taking the combined output values at irregular time intervals.

WAVELYNX EX. 1003, Pg. 556
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

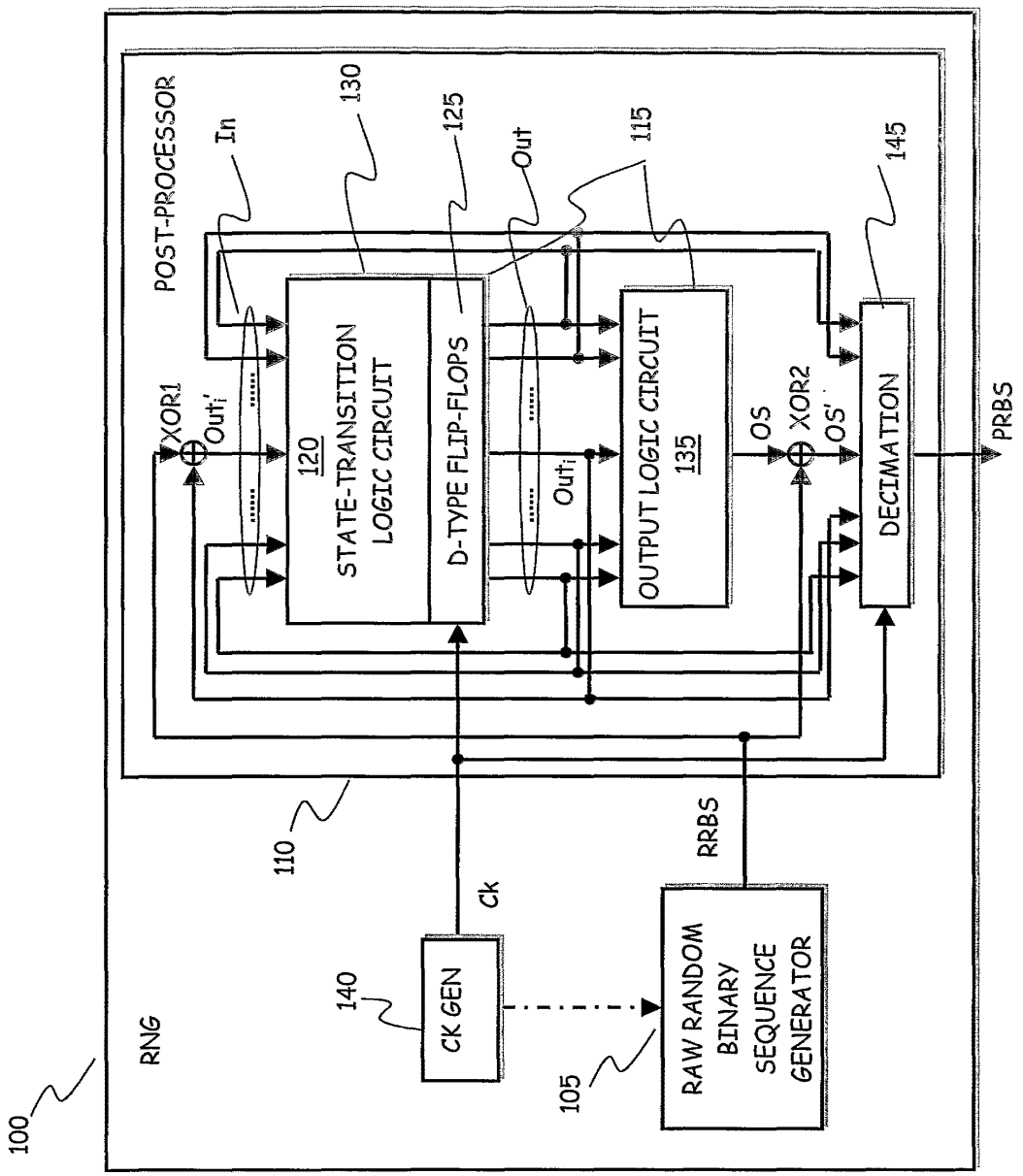

100
RNG
110
POST-PROCESSOR
In
130
125
Out
115
145
XOR1
Out$_i$'
120
STATE-TRANSITION
LOGIC CIRCUIT
D-TYPE FLIP-FLOPS
Out$_i$
OUTPUT LOGIC CIRCUIT
135
OS
XOR2
OS'
DECIMATION
PRBS
Ck
RRBS
140
CK GEN
105
RAW RANDOM
BINARY
SEQUENCE
GENERATOR

FIG. 1

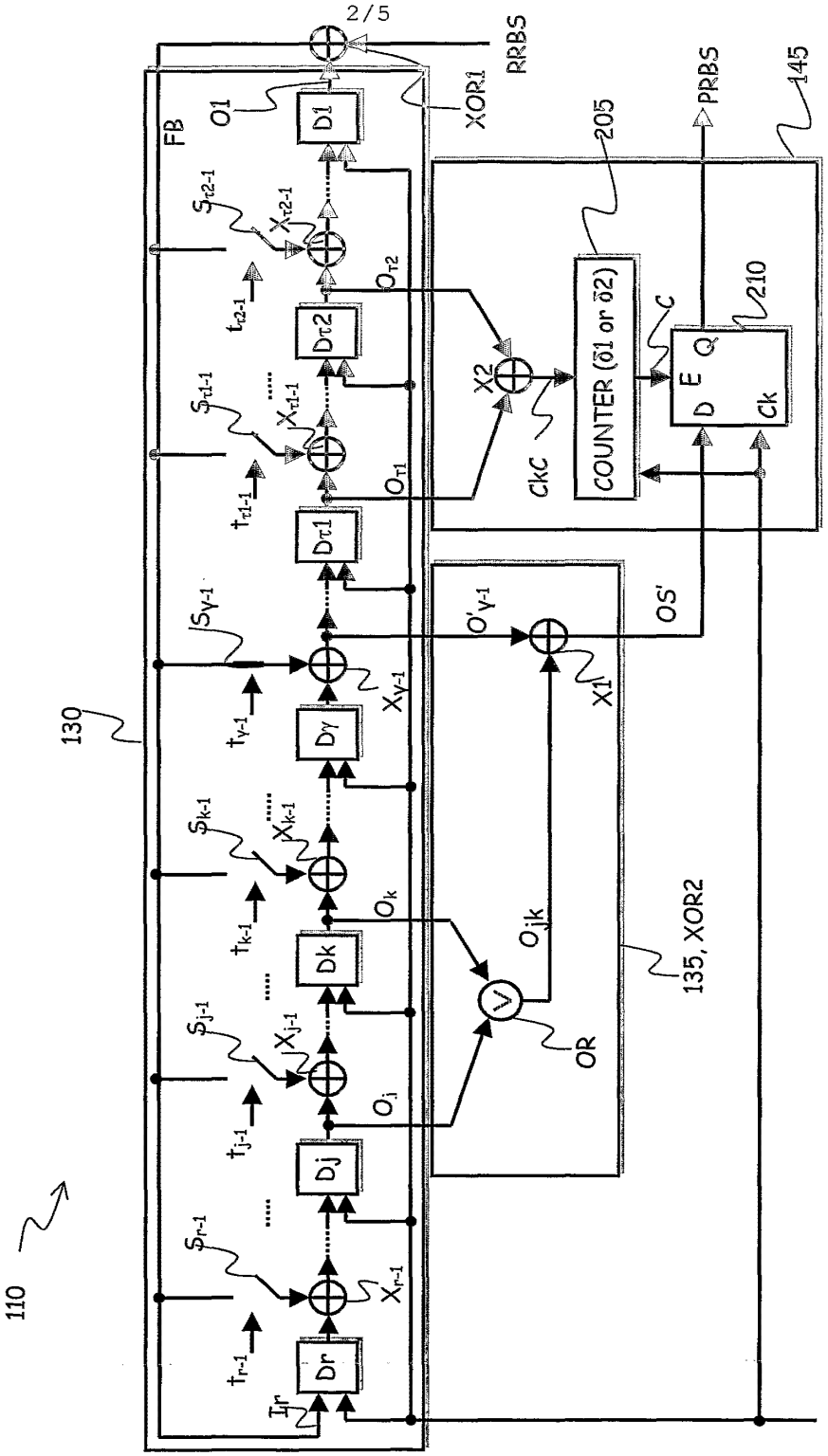

110
130
FB
O1
D1
XOR1
RRBS
205
PRBS
145
Sr-1
Xr-1
Dr
Ir
tr-1
Sj-1
Xj-1
Dj
Oj
tj-1
Sk-1
Xk-1
Dk
Ok
tk-1
Sy-1
Xy-1
Dγ
Oτ1
Dτ1
tτ1-1
Sτ2-1
Xτ2-1
Dτ2
Oτ2
tτ2-1
X2
CkC
COUNTER (δ1 or δ2)
C
210
D E Q
Ck
O'y-1
OS'
X1
Ojk
OR
135, XOR2
Ck

FIG. 2

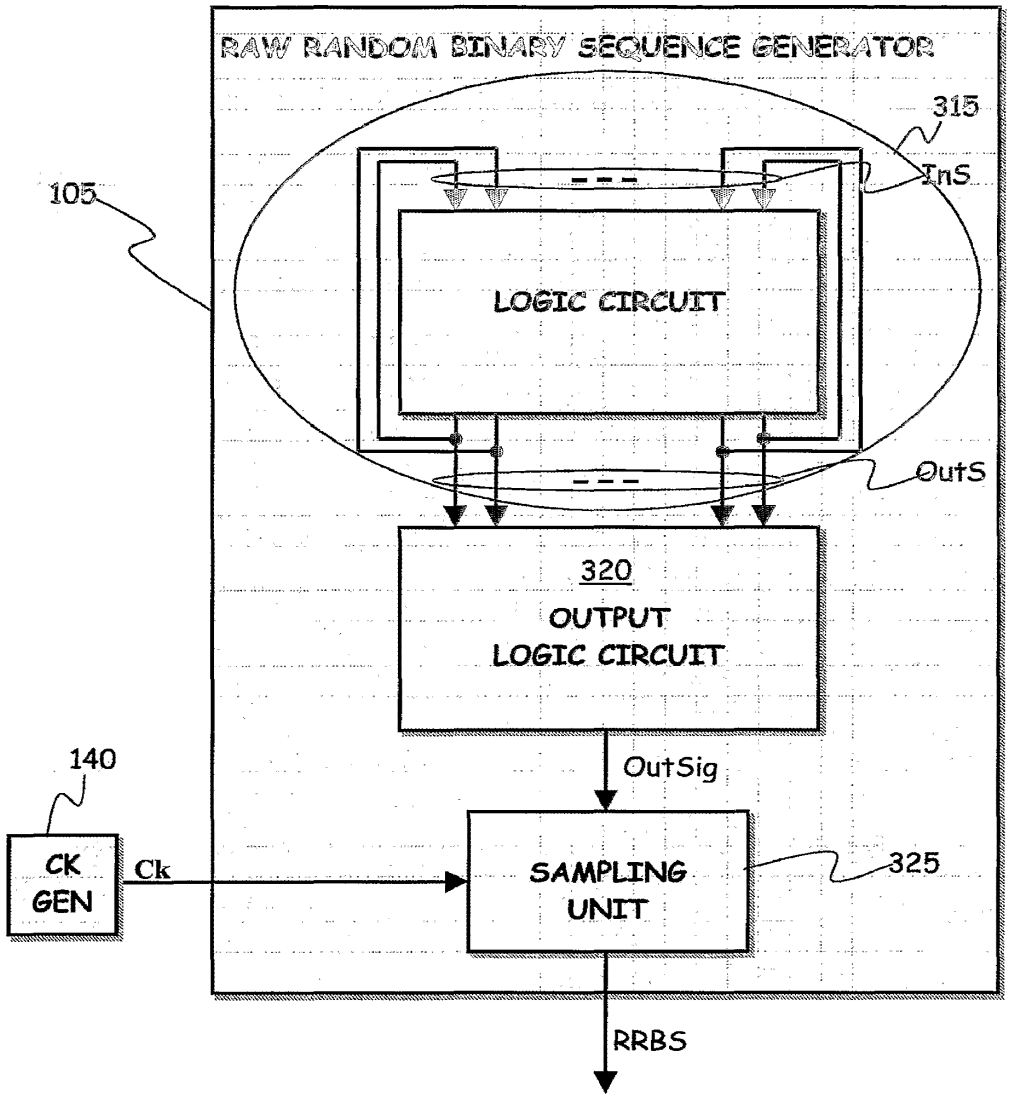

RAW RANDOM BINARY SEQUENCE GENERATOR
315
InS
105
LOGIC CIRCUIT
OutS
320
OUTPUT
LOGIC CIRCUIT
140
OutSig
CK
GEN
Ck
SAMPLING
UNIT
325
RRBS

FIG. 3

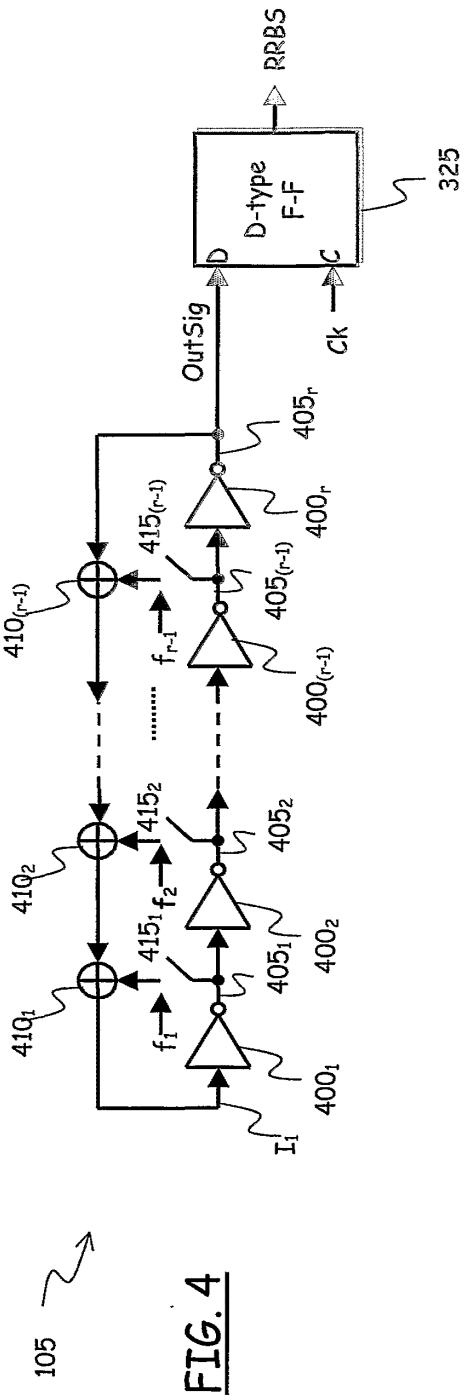

RRBS
D-type F-F
D
C
325
OutSig
Ck
405_r
400_r
$415_{(r-1)}$
$410_{(r-1)}$
$405_{(r-1)}$
f_{r-1}
$400_{(r-1)}$
415_2
410_2
405_2
f_2
415_1
405_1
400_2
410_1
f_1
400_1
I_1
105

FIG. 4

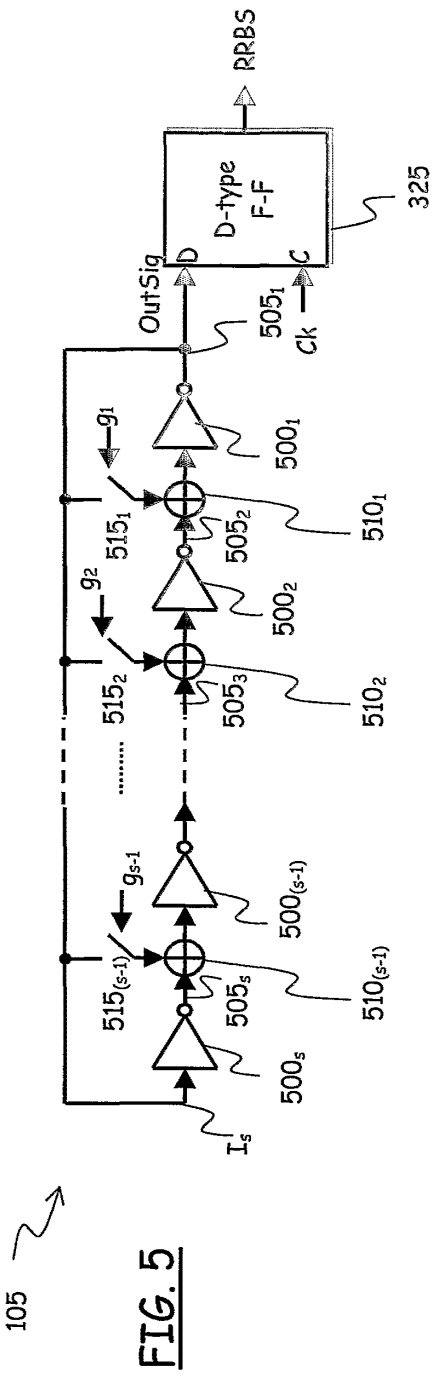

RRBS
D-type F-F
D
C
325
OutSig
Ck
505_1
500_1
g_1
515_1
505_2
510_1
500_2
g_2
515_2
505_3
510_2
$500_{(s-1)}$
g_{s-1}
$515_{(s-1)}$
505_s
$510_{(s-1)}$
500_s
I_s
105

FIG. 5

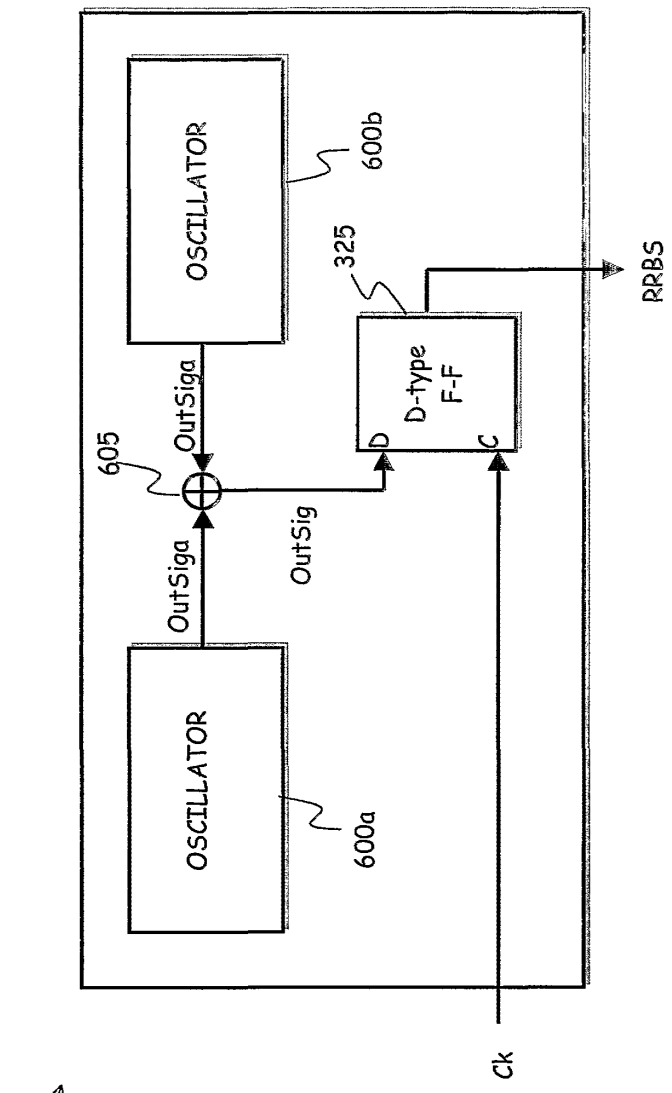

OSCILLATOR
600b
325
RRBS
D-type
F-F
D
C
OutSiga
605
OutSig
OutSiga
OSCILLATOR
600a
Ck
105

FIG. 6

INTERNATIONAL SEARCH REPORT

International Application No
PCT/EP2004/051747

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F7/58

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 782 069 A (NEC CORPORATION) 2 July 1997 (1997-07-02)	1,2, 5-10, 18-21
Y	column 9, line 6 - line 57; figures 3,7 column 11, line 40 - line 52; figure 10	3,4, 11-17,22
Y	KENCHENG ZENG: "PSEUDORANDOM BIT GENERATORS IN STREAM-CIPHER CRYPTOGRAPHY" COMPUTER, IEEE COMPUTER SOCIETY, LONG BEACH., CA, US, US, vol. 24, no. 2, 1 February 1991 (1991-02-01), pages 8-17, XP000219462 ISSN: 0018-9162 page 15 - page 16	3,4, 15-17,22
Y	figure 5	11-17

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 March 2005	06/04/2005

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Cohen, B

Form PCT/ISA/210 (second sheet) (January 2004)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/EP2004/051747

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 3 875 528 A (HIRSCH ET AL) 1 April 1975 (1975-04-01) abstract; figure 1 column 2, line 3 - line 6	1,19
A	US 3 790 768 A (CHEVALIER P,CA ET AL) 5 February 1974 (1974-02-05) figure 1	1,3,19
A	figure 2	18
A	US 2003/059045 A1 (RUEHLE MICHAEL D) 27 March 2003 (2003-03-27) paragraph '0022!; figure 2	11-14

Form PCT/ISA/210 (continuation of second sheet) (January 2004)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/EP2004/051747

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0782069 A	02-07-1997	JP 9179726 A	11-07-1997
		AU 7539696 A	03-07-1997
		CA 2193196 A1	26-06-1997
		EP 0782069 A1	02-07-1997
US 3875528 A	01-04-1975	FR 2172459 A5	28-09-1973
		DE 2306850 A1	16-08-1973
		GB 1406013 A	10-09-1975
		IT 978863 B	20-09-1974
		JP 48094337 A	05-12-1973
		NL 7301609 A	14-08-1973
US 3790768 A	05-02-1974	NONE	
US 2003059045 A1	27-03-2003	EP 1430391 A2	23-06-2004
		WO 03027832 A2	03-04-2003

Form PCT/ISA/210 (patent family annex) (January 2004)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 March 2006 (30.03.2006)

PCT

(10) International Publication Number
WO 2006/032941 A1

(51) International Patent Classification[7]: G06F 7/58, H04L 9/22, G11C 19/00

(21) International Application Number: PCT/IB2004/003085

(22) International Filing Date: 22 September 2004 (22.09.2004)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 ESPOO (FI).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **HEMMING, Erwin** [DE/DE]; Koenigstrasse 13, 44651 Herne (DE).

(74) Agent: **KURIG, Thomas**; Becker, Kurig, Straus, Bavariastrasse 7, 80336 München (DE).

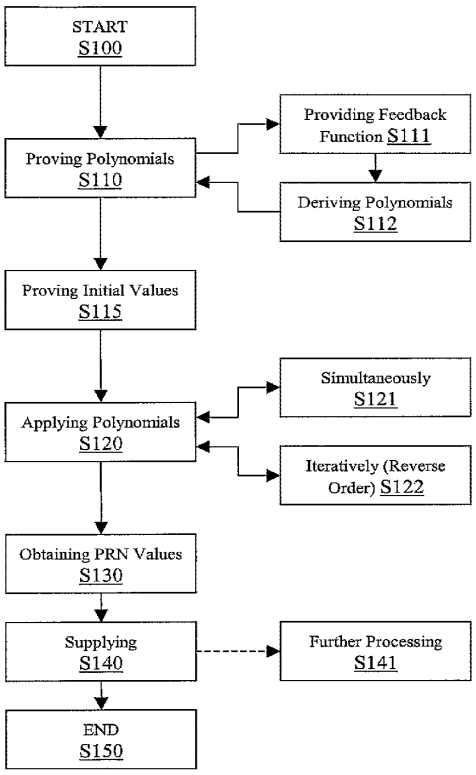
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS FOR GENERATING PSEUDO RANDOM NUMBERS



(57) Abstract: The present invention proposes a methodology implementable in form of a hardware or software module for generating a pseudo random number. The pseudo random number corresponds to a pseudo random sequence of bits, which form the pseudo random number. A plurality of m polynomials is provided. The polynomials are derived from an original polynomial, which defines a feedback function of a linear feedback shift register capable for generating the pseudo random number. The polynomials are functions of n bits, which serve as initial bits and seed bits, respectively. Then, the polynomials are applied on the initial bits for generating the pseudo random number, which comprises at least m bits resulting from the m polynomials. Due to the fact that the polynomials are independent from each other, i.e. the initial bits serve as input values to the polynomials, the polynomials can be applied substantially simultaneously or in any other sequence.

WO 2006/032941 A1

METHOD AND APPARATUS FOR GENERATING PSEUDO RANDOM NUMBERS

The present invention relates to the field of pseudo random numbers (PRN) and in particular to methods and apparatus for generating such pseudo random numbers (PRN) in an efficient manner.

A huge number of technical applications require pseudo random numbers (PRN). Typical applications are spreading codes in CDMA (code division multiple access) transceivers, pseudo random function for scrambling (data encryption), pseudo random number bit generators as known from conventional programming languages, pseudo random number bit generators for test pattern generation used in IC (integrated circuit) production testing, codes used in GPS (global positioning systems), signature test pattern generation, generic algorithm programs etc, just for the way of illustration.

One approach for generating truly random numbers is to measure some kind of continuous natural phenomena such as the noise power level in a radio frequency receiver. Such noise power level appears to be random because the power level at any instant in time depends on an undeterminable number of variables. Nevertheless, for most applications pseudo random numbers (PRN) are sufficient and are obtainable simpler. A set of values or elements that is statistically random over long periods (dependent on the specific application) but derived from a known starting point, i.e. follow predicable patterns, is designated as pseudo random number (PRN).

Several algorithms are known and applied for generating such pseudo random numbers. One conceptually simple way to generate long sequences of pseudo random numbers uses a linear feedback shift register (LFSR). Due to their simplicity linear feedback shift registers (LFSR) are widely applied, particularly for synchronizing sending and receiving devices in a spread spectrum transmission. Figs. 1a and Fig. 1b illustrate block diagrams showing an example circuit for

CONFIRMATION COPY

WAVELYNX EX. 1003, Pg. 566
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

implementing a shift register and a circuit diagram depicting an exemplary linear feedback mechanism for generating one or more pseudo random numbers.

The block diagram of Fig. 1a illustrates an example circuit composed of several flip-flops connected in series to realize a simple shift register. A flip-flop or bistable multivibrator is a pulsed digital circuit capable of serving as a one-bit memory used in electronics and computing. Typically, a flip-flop includes zero, one or two input terminals, a clock terminal, and an output terminal, even though commercial flip-flops additionally provide complement output terminal supplying the complementary output signal. Naturally, supplementary terminals required for operation of a flip-flop are included such as power supply terminal and ground terminal. By supplying a pulsing or strobing clock signal to the corresponding clock terminal, the flip-flop is caused to either change or retain its output signal being based upon the values of the input signals and the characteristic equation of the flip-flop. More precise, the wording strobing or pulsing clock signal is a simplified view. Any change of output state actually coincides with either the leading edge or the trailing edge of the clock pulse, and, to further complicate matters, may correspond to either a low-to-high or a high-to-low transition of the clock signal. Four types of flip-flops are commonly applied in clocked sequential systems, which are T ("toggle") flip-flops, S-R ("set-reset") flip-flops, J-K flip-flops, and D ("delay") flip-flops. The behavior of the flip-flops is described by their specific characteristic equation, which is a function of the input signal(s) (e.g. R and S, or J and K, or D etc.) and the current output signal Q and which results in the next output signal Q_{next} served by the output terminal after the next clock pulse.

In particular, the D ("delay") flip-flop takes one input signal present at the input terminal D, which input signal the D flip-flop conveys as output signal at the output terminal Q when the clock is strobed. Regardless of the current value or state of the output, it will assume a value "1" if D = 1 when the flip-flop is strobed or a value "0" if D = 0 when the flip-flop is strobed. This D flip-flop can be interpreted as a primitive delay line or zero-order hold, since the data (a one-bit data information) is posted at the output one clock cycle after the data arrives at the input. The characteristic equation denotes as following:

$$Q_{next} = D,$$

and the corresponding truth table is:

D	Q	Q_{next}
0	X	0
1	X	1

The D flip-flop is suitable to realize registers to store numbers. Consequently, the shift register depicted in Fig. 1a for the way of illustration is composed of D flip-flops, wherein the first D flip-flop 0 is supplied with an input signal DS and the input terminal D_i (i = 0,..., n) of each succeeding D flip-flop FF_1 to FF_n is connected to the output terminal Q_i (i = 0,..., n) of the preceding D flip-flop FF_0 to FF_{n-1}.

Those skilled in the art will appreciate that the circuit depicted in Fig. 1a enables a shift register being composed of a number of n registers, i.e. the D flip-flops FF_0 to FF_n, set up in a linear fashion, which have their inputs and outputs connected together in such a way that the register element states are shifted with each clock pulse from the register elements / flip-flops having low numbers to the register elements / flip-flops having high numbers. Conventionally, shift registers may have a combination of serial and parallel inputs and outputs, including serial-in, parallel-out and parallel-in, serial-out types. There are also types of shift registers that have both serial and parallel input and types of shift registers with serial and parallel output. There are also bi-directional shift registers, which allow you to vary the shift direction of the shift register. The depicted shift register shows a serial input designated as DS and a serial output designates as Q_n. A parallel output can be realized by taping the outputs D_i (i = 0,..., n).

Those skilled in the art will appreciate that the description of shift registers given above and the illustrated implementation of a shift register of the basis of D flip-flops is neither conclusive nor complete since the realization of shift registers is out of the scope of the present invention. However, the description stated above enables the skilled reader for understanding the concept of

the invention provided in detail below. Firstly, a short introduction to feedback mechanism implementable on the basis of shift registers will be given complementary.

The block diagram of Fig. 1b illustrates a shift register with several register elements x^i, which shift register input is supplied with a feedback signal in accordance with a feedback logic / function. Each register element x^i or register element or register cell is realized exemplary by a flip-flop as aforementioned, whereas the designation x^i shall refer to an individual register element, the totality of which forms the shift register, and simultaneously represent a content variable of the individual register element, which is capable of storing a one-bit informational content, i.e. $x^i = 0$ or $x^i = 1$. This means, each register element of the shift register serves as a one-bit storage cell. For the way of illustration, the illustrated shift register includes ten register elements x^1 to x^{10}; consequently, the shift register is capable of storing an informational content of 10 bits.

Upon each clock cycle, the content of the register element x^i is shifted to the succeeding register element x^{i+1}, which carries the shifted content after completion of the clock cycle. This means, an output signal of the shift register is available at the shift register output, which corresponds to the content of the last register element x^{10}. With each clock cycle the content of the last register element x^{10} is overwritten with the content of the last but one register element x^9 provided at the shift register output. The input of the shift register is supplied with input signal caused by feedback function. The feedback function is supplied with signals resulting from two or more tapped register elements of the shift register. In an exemplary fashion, the outputs of the register elements x^3 and x^{10} (i.e. the output signal currently provided at their output terminals) are tapped and supplied to the feedback function, which is herein for simplicity an exclusive-OR (XOR) function combining the tapped content values x^3 and x^{10}. In general, the outputs that influence the input are called taps. The following truth table illustrates the results of the XOR function:

XOR "⊕"	0	1
0	0	1

1	1	0

Assuming that the shift register has initially a non-zero initial values and seed value, respectively, the shift register outputs a stream of output signals with each clock cycle, which forms a pseudo random number sequence of 1-bit values. The maximal periodicity of the outputted pseudo random number sequence is predefined by the number of different states of the shift register, which number is $2^n - 1$, where n is the number of registers. Assuming again the number of 10 registers, the number of different states results to $2^{10} - 1 = 1023$. The state, which is characterized by the register values each being equal zero, is stable and has to be prevented. In particular, the real periodicity, which is equal or smaller than the obtainable maximal periodicity depends on the taps supplied to the feedback function as well as the feedback function itself. In case of a linear feedback function as illustrated with reference to Fig. 1b, the periodicity can be determined by the means of mathematical methods.

The description above has been introduced into the field of shift registers and their specific application as pseudo random number generators by the use of linear feedback shift registers (LFSR) with adequate feedback functions. Nevertheless, those skilled in the art will appreciate immediately that the illustrated methodology implies various deficiencies. The skilled reader understands that the aforementioned algorithm for realizing pseudo random number generators on the basis of linear feedback shift registers (LFSR) represents a recursive algorithm. This means, upon each iteration of the algorithm or clock cycle only one (single) bit results from the shift register and is provided at its output. For generating a final pseudo random sequence of m bits the aforementioned algorithm requires m iterations and clock cycles, respectively.

Some applications require pseudo random numbers of numerous bits with high data rates for processing. To comply with such requirements, the clock cycle rate may be increased, which implies the implementation of adequate shift registers and flip-flops operating at such increased data rates, or alternatively deep combinatorial logic enabling of performing several iterations in a single clock cycle. Shift registers and flip-flops adapted to high data rates are expensive and deep combinatorial logic includes a reduction of the clock cycle rate due to long delays caused by the deep combinatorial logic.

WAVELYNX EX. 1003, Pg. 570
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

It is now invented a methodology for generating pseudo random numbers on the basis of the aforementioned shift register feedback logic, which overcomes the deficiencies of the state of the art stated above. In particular, an embodiment of the present invention provides a methodology for generating pseudo random numbers at a high rate factor in comparison with the iterative methodology. More particularly, the inventive methodology allows a generation of the bits of pseudo random numbers in reverse order.

The objects of the present invention are solved by the subject matter defined in the accompanying independent claims.

According to a first aspect of the present invention, a method for generating a pseudo random number is provided. The pseudo random number corresponds to a pseudo random sequence of bits, which form the pseudo random number. A plurality of m polynomials is provided. The polynomials are derived from an original polynomial, which defines a feedback function of a linear feedback shift register capable for generating the pseudo random number. The polynomials are functions of n bits, which serve as initial bits and seed bits, respectively. Then, the polynomials are applied on the initial bits for generating the pseudo random number, which comprises at least m bits resulting from the m polynomials. Due to the fact that the polynomials are independent to each other, i.e. the initial bits serve as input values to the polynomials, the polynomials can be applied substantially simultaneously or in any other sequence.

According to an embodiment of the present invention, the plurality of m polynomials are derived from the original polynomial by stepwise defining the polynomials each representing a feedback function at a defined iteration and reducing the polynomials to obtain polynomials, which are functions of the initial bits.

According to another embodiment of the present invention, the polynomials are applied substantially simultaneously. Further, according to another embodiment of the present invention, the polynomials are applied in reverse order to generate firstly lower bits of the pseudo random number and subsequently higher bits of the pseudo random number.

WAVELYNX EX. 1003, Pg. 571
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

According to yet another embodiment of the present invention, the polynomials represent logic relationships for combining at least two values each having a domain of one bit.

According to a further embodiment of the present invention, the m polynomials are static polynomials, i.e. the polynomials are (pre-)determined by the original polynomial defining the feedback function such that the polynomials can be provided fixedly for application. According to another embodiment of the present invention, the m polynomials are dynamic polynomials, i.e. the polynomials are derived from the provided original polynomials by reduction to the order of the original polynomial such that the polynomials are functions of the initial values. The dynamical provision of the polynomials allows for providing differing original polynomials as the basis for the generation of the sequence of pseudo random bits enabling the forming of a pseudo random number thereof.

According to yet a further embodiment of the present invention, the maximal number m_{max} of the polynomials is less or equal to $2^{np} - (n + 1)$. According to an additional embodiment of the present invention, an order of said polynomials is equal or less to np, where np represents an order of the original polynomial.

According to yet an additional embodiment of the present invention, the pseudo random number comprising the m bits resulting from the polynomials and the n initial bits is identical with a sequence obtained from the linear feedback shift register used as a pseudo random number generator and applying the feedback function corresponding to the original polynomial.

According to a second aspect of the present invention, a module for generating a pseudo random number is provided. The module comprises an initial storage having a number of n initial storage cells. Each initial storage cell serves as a one-bit storage. The module includes additionally a result storage having a number of m result storage cells. Each result storage cell serves also as a one-bit storage. A combinatorial logic of the module is selectively coupled to output terminals of said n initial storage cells and is selectively coupled to input terminals of said m result storage cells. The combinatorial logic implements a number of m polynomials, which are derived from an original polynomial defining a feedback function of a linear feedback shift register operable as

WAVELYNX EX. 1003, Pg. 572
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

a pseudo random number generator. The polynomials are functions of the n bits serving as initial bits and stored in the initial storage cells.

According to an embodiment of the present invention, the combinatorial logic is implemented as a software module, which comprises code sections. When executed on a processing unit, the code sections perform logical relationship operations in accordance with the combinatorial logic defined on the basis of the polynomials.

According to another embodiment of the present invention, the combinatorial logic is implemented by the means of a plurality of logic components. Each logical component has at least two input terminals for receiving one-bit inputs and each having at least one output terminal for providing a one-bit output. The output result is defined by the predefined combinatorial logic relationship.

According to a third aspect of the present invention, an electronic apparatus is provided, which comprises an initial storage, a result storage and a combinatorial logic. The initial storage has a number of n initial storage cells and the result storage has a number of m result storage cells. Each initial or result storage cell serves also as a one-bit storage. The combinatorial logic of the module is selectively coupled to output terminals of said n initial storage cells and is selectively coupled to input terminals of said m result storage cells. The combinatorial logic implements a number of m polynomials, which are derived from an original polynomial defining a feedback function of a linear feedback shift register operable as a pseudo random number generator. The polynomials are functions of the n bits serving as initial bits and stored in the initial storage cells.

According to a fourth aspect of the present invention, a system for generating a pseudo random number is provided. The system includes a number of n initial states representing n initial 1-bit values, and a number of m result states representing m result 1-bit values. A combinatorial logic being also comprised by the system is selectively supplied with said n initial states and supplies selectively said m result states. The combinatorial logic implements a number of m polynomials, which are derived from an original polynomial. The original polynomial defines a feedback function of a linear feedback shift register. The polynomials are functions of n 1-bit states, which states serve as initial states defining the initial values.

According to a fifth aspect of the present invention, a computer program product for generating a pseudo random number is provided, which comprises program code sections stored on a machine-readable medium for carrying out the steps of the method according to any aforementioned embodiment of the invention, when the computer program product is run on a processor-based device, a computer, a terminal, a network device, a mobile terminal, or a mobile communication enabled terminal.

According to a sixth aspect of the present invention, a computer program product for generating a pseudo random number is provided, comprising program code sections stored on a machine-readable medium for carrying out the steps of the aforementioned method according to an embodiment of the present invention, when the computer program product is run on a processor-based device, a computer, a terminal, a network device, a mobile terminal, or a mobile communication enabled terminal.

According to a seventh aspect of the present invention, a software tool is provided. The software tool comprises program portions for carrying out the operations of the aforementioned methods when the software tool is implemented in a computer program and/or executed.

According to an eighth aspect of the present invention, a computer data signal embodied in a carrier wave and representing instructions is provided which when executed by a processor causes the steps of the method according to an aforementioned embodiment of the invention to be carried out.

Advantages of the present invention will become apparent to the reader of the present invention when reading the detailed description referring to embodiments of the present invention, based on which the inventive concept is easily understandable.

The accompanying drawings are included to provide a further understanding of the invention, and are incorporated in and constitute a part of this specification. The drawings illustrate embodiments of the present invention and together with the description serve to explain the principles of the invention. In the drawings,

WAVELYNX EX. 1003, Pg. 574
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Fig. 1a shows a block diagram illustrating an example circuit composed of several flip-flops connected in series to realize a simple shift register;

Fig. 1b shows a block diagram illustrating a shift register with several registers x^i, which shift register input is supplied with linear feedback signal;

Fig. 2 illustrates a schematic sequence diagram enabling generation of random bits in accordance with embodiments of the present invention;

Fig. 3a shows a block diagram illustrating a logic diagram for generating simultaneously a number of n random bits according to an embodiment of the present invention;

Fig. 3b shows a block diagram illustration a schematic general block diagram of an input storage and output storage being linked together by a combinatorial logic for generating a pseudo random sequence according to an embodiment of the present invention;

Fig. 4a illustrates schematically a block diagram of a base station CDMA transmitter and

Fig. 4b illustrates schematically a block diagram of a GPS receiver.

Reference will be made in detail to the embodiments of the invention examples, which are illustrated in the accompanying drawings. Wherever possible same reference numbers are used throughout drawings and description to refer to similar or like parts. The following description relates to various embodiments based on which the skilled reader will understand the principle inventive concept of the present invention. Nevertheless, the skilled reader will appreciate that the inventive concept is likewise applicable to further embodiments, which are covered by the scope of the accompanying claims.

The starting point of the description of the inventive concept shall be based on the known linear feedback shift register illustrated in Fig. 1b and described above in detail. For simplicity and for the way of illustration, the linear feedback function is given by combining the two tapped register

elements x^3 and x^{10} in accordance with the XOR relationship, which is defined above by the means of a truth table. The following embodiment is based on this exemplary linear feedback function. It shall be noted that the present invention is not limited to this specific linear feedback function. However, based on the teaching, which will be given below with reference to this specific linear feedback function, a skilled reader will appreciate the inventive concept and will be able to generalize the teaching, which is applicable to numerous feedback functions implementable with shift registers in the aforementioned general manner. The scope of the invention is solely defined in the accompanying claims and not limited by any specific or particular embodiments of the detailed description.

The feedback function, illustrated in the logic diagram of Fig. 1b, can be denoted mathematically in form of a recursive polynomial, which is applied at each iterative cycle of the shift register. Assume that the shift register and its register elements are filled with original bit values or seed bit values, which serve as initial content for the generation of the pseudo random bit sequence in accordance with the feedback function such as described above with reference to Fig. 1b. Note that the stable state (i.e. all register elements are equal zero) has to be prevented. The mathematical denotation of the feedback function and original polynomial, respectively, can be written as

$$x^3 \oplus x^{10},$$

where the (bit) content of the third register element x^3 and the (bit) content of the tenth register element x^{10} is combined by the XOR relationship. Consequently, a new bit value x^0, which represents the current result of the feedback function and which will become content of the first register element x^1 upon one clock cycle and iteration, respectively, can be predicted and can be calculated by the following polynomial:

$$x^0 = x^3 \oplus x^{10}.$$

It should be noted that in the following description reference to an order of the polynomials will be given. The order of the polynomial x^0 as defined above results from the maximum order of the logically associated elements, herein the element x^3 having the order 3 and the element x^{10} having the order 10. This means, an element x^i should have assigned an order i. In consequence to the definition above, the polynomial x^0 has an order 10. It should be also noted that the number of elements, herein the ten elements x^1 to x^{10}, typically corresponds to the order of the original polynomial due as further elements do not contribute to the feedback function and thus such further elements are needless.

Conclusively, further new / future bit values x^{-i}, which represents the result of the feedback function after i clock cycles or iterations and which will become content of the first register element x^1 upon i+1 clock cycles or iterations, can be predicted and calculated in analogy. Exemplarily, the polynomial for the new bit values x^{-1} and x^{-2} can be written as:

$$x^{-1} = x^2 \oplus x^9, \text{ and}$$

$$x^{-2} = x^1 \oplus x^8.$$

It should be noted that predicted new / future bit values x^{-i} are denoted with a negative exponent including the exponent 0. The order of the corresponding polynomial for predicting the new / future bit values x^{-i} should be defined on the basis of its elements. This means, the order of the polynomial x^{-1} is equal to 9 and the order of the polynomial x^{-2} is equal to 8.

Due to the shift register functionality, which shifts the content of the register elements x^i to the register elements x^{i+1} upon each clock cycle or iteration, the new (future) bit value x^{-1} is obtainable from the register elements x^2 and x^9, because x^0 has to be calculated firstly and shifted into the shift register such that the contents of the register elements x^2 and x^9 is present in the register elements x^3 and x^{10} when x^{-1} is to be calculated. Hence, one intermediate shift cycle has to be considered. The same argumentation applies to the calculation of the new (future) bit value x^{-2}, but there has to be considered two intermediate shift cycles.

WAVELYNX EX. 1003, Pg. 577
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Polynomials can be modeled in the aforementioned manner for each new (future) bit value x^{-i} (i = 1, 2, 3,..., n). An exemplary selection of polynomials is given below:

$$x^{-3} = x^0 \oplus x^7 = \left(x^3 \oplus x^{10}\right) \oplus x^7 = x^3 \oplus x^7 \oplus x^{10},$$

$$x^{-4} = x^{-1} \oplus x^6 = \left(x^2 \oplus x^9\right) \oplus x^6 = x^2 \oplus x^6 \oplus x^9,$$

$$x^{-5} = x^{-2} \oplus x^5 = \left(x^1 \oplus x^8\right) \oplus x^5 = x^1 \oplus x^5 \oplus x^8,$$

$$x^{-6} = x^{-3} \oplus x^4 = \left(x^3 \oplus x^7 \oplus x^{10}\right) \oplus x^4 = x^3 \oplus x^4 \oplus x^7 \oplus x^{10},$$

$$x^{-7} = x^{-4} \oplus x^3 = \cdots,$$

$$\vdots$$

$$x^{-33} = x^{-30} \oplus x^{-23} = \cdots = x^1 \oplus x^2 \oplus x^3 \oplus x^4 \oplus x^5 \oplus x^9 \oplus x^{10},$$

$$\vdots.$$

With reference to x^{-3}, the polynomial defining the feedback value obtained after three intermediate shift cycles is determined of the values of the register element x^0 and the register element x^7. Skilled persons appreciate that the content / value of the register element x^0 is unknown because the register element x^0 is not part of the initial values. By including the above definition of the polynomial for x^0, the polynomial x^{-3} can be reduced to a function of initial values and a function, which has an order within a range of orders from 1 to the order of the original polynomial. The skilled reader will appreciate on the basis of the exemplary selection of polynomials above that known polynomials of lower order (i.e. higher index –i) are included into polynomials of higher order (i.e. lower index –i) such that the polynomials defining the new (future) bit values x^{-i} become functions equal or less than the order of the original polynomial, i.e. a function of the content / values of the register elements x^1 to x^{10} only.

The aforementioned exemplary selection of polynomials can be expanded to any number of desired polynomials serving as a sequence of polynomials to generate a pseudo random bit sequence of any number of bits. Such a pseudo random bit sequence may be interpreted as a

random number having the corresponding bit length. Note that the aforementioned periodicity and number of differing states may have to be considered, respectively.

With reference to Fig. 2, an operation sequence block diagram enabling the generation of pseudo random bits and one or more pseudo random numbers (which pseudo random numbers are composed of the pseudo random bits) according to embodiments of the invention is depicted.

In an operation S100, the operational sequence starts. In accordance with the aforementioned description of the inventive concept, a number of m polynomials are provided in an operation S110, which polynomials are obtained from an original polynomial suitable for defining a feedback function of a feedback shift register as defined above. The polynomials are obtained in an operation successively applying the original polynomial and reducing the polynomials to functions which include orders in a range of one ("1") to the maximal order of the original polynomial, which is herein ten in accordance with the order of the exemplary original polynomial. The polynomials may be statically defined or may be dynamically derived from the original polynomial. In case of statically defined polynomials, the polynomials are based on a pre-defined original polynomial, from which the polynomials have been derived, and may be provided directly or fixedly. In case of dynamical defined polynomials, the polynomials are derived from an original polynomial operable with a linear feedback shift register as a feedback function, which function or original polynomial is provided in an operation S111. The polynomials are derived form the original polynomial in an operation S112 in accordance with the basic inventive concept illustrated above on the basis of the exemplary polynomial.

In an operation S120, the initial values are provided. The provision of the initial values enables the application of the polynomials thereon, which results in an operation S130 in obtaining a sequence of pseudo random values, i.e. pseudo random bits, formed of the result values of each of the polynomials. Consequently in step S140, the obtained sequence of pseudo random bits can be supplied for further processing in step S141 requiring the sequence of pseudo random bits or a pseudo random number to be formed thereof. If no further processing is carried out the method comes to an end at step S150.

The application of the polynomials on the initial values may be performed in varying embodiments. With reference to an operation S121, the provided polynomials enable to essentially obtain simultaneously all pseudo random values from the polynomials, which realization of the essentially simultaneous obtainment may address a hardware implementation such as described exemplary with reference to Fig. 3a.

In contrast to operation S121, in an operation S122 an iterative application of the polynomials on the initial values can be performed. In order to illustrate the advantage of such an iterative process, reference shall be given to Fig. 1b, which illustrates a feedback shift register for generating a sequence of pseudo random values (bits). Assume that a sequence of pseudo random bits generated by the feedback shift register of Fig. 1b forms a pseudo random number having a most significant bit (MSB) and a least significant bit (LSB). Conventionally, the most significant bit is the first bit shifted to the output of the shift register, whereas the least significant bit is formed of the last bit shifted to the output of the shift register. Each sifting step corresponds to an iterative cycle of the shift register. Some applications, especially those described with reference to Figs. 4a and 4b, accept a pseudo random number bitwise in sequence. But these applications assume to receive firstly the least significant bit (LSB) and lastly the most significant bit (MSB). With reference to the state of the art generating of pseudo random number by the means of a feedback shift register, all required iterative cycles have to be operated before the least significant bit is available, whereas the number of required cycles depend of the number of bits of the sequence of pseudo random bits, i.e. the length of the pseudo random number. It should be noted that the sequence of pseudo bits may comprise the initial values / bits or alternatively the initial values / bits are neglected.

With reference to the operation S122 according to an embodiment of the invention, the polynomials are applied in reverse order, which means that the least significant bit of the aforementioned sequence of pseudo random bits is obtainable firstly and the most significant bit of the aforementioned sequence of pseudo random bits is obtainable lastly. The sequence of the pseudo random bits is obtained in reverse order in comparison with the state of the art generation. The advantage of the reverse order obtainment of the sequence of the pseudo random bits, the generating procedure of the sequence of pseudo random bits and the further processing procedure, which requires the sequence of pseudo random bits for processing, can be interwoven

or combined such that at each cycle a pseudo random bits is obtained and supplied to the further processing procedure, which processes the supplied pseudo random bit in the next cycle. Those skilled in the art will appreciate the possibility of interweaving the generation process according to an embodiment of the present invention, which results in a signification improvement of the total processing time.

Fig. 3a shows a block diagram, which illustrates a logic diagram for generating simultaneously a number of n random bits or a corresponding random number according to an embodiment of the present invention. The illustrated logic diagram including a plurality of XOR logic elements or XOR relationship operators can serve as a basis for realizing a corresponding logic circuit for instance forming a part of an application specific integrated circuit (ASIC). Alternatively, the logic circuit may be implemented by the means of a programmable logic component. The illustrated logic diagram including a plurality of XOR relationship operators can alternatively serve as a basis for implementing a corresponding software-based method carrying out operations in accordance with the logic diagram. The logic diagram of Fig. 3a illustrates a selection of logical combination of the original bit values or seed bit values x^i with i = 1, 2,..., 10, required for the illustratively depicted predicted (new / future) bit values x^{-i}, i = 0, 1, 2, 3, 4,..., 33. The seed bit values x^i with i = 1, 2, ..., 10 may be provided as valid signals or may be obtained by tapering storage cells storing the seed bit values x^i, i = 1, 2,..., 10. The same applies to the predicted (new / future) bit values x^{-i}, i = 0, 1, 2, 3, 4,..., 33, which may be available as valid signals for further processing or which may be supplied to storage cells to be stored therein and provided to be read out allowing further processing thereof. The logic circuits reproduce the polynomials obtained by reduction to the order of the original polynomial described above. For instance with reference to x^{-33}, the storage cells, which stores the seed values x^1, x^2, x^3, x^4, x^5, x^9 and x^{10}, are tapped from and supplied to XOR logic elements in accordance with the polynomial x^{-33}.

The implementation of the logic diagram shown in Fig. 3a in form of an integrated circuit or as a program comprising code sections for performing the illustrated logic rules is part of the knowledge of those skilled in the specific art.

Conclusively, the inventive concept provides a methodology to parallelize at least partly the iterative and recursive pseudo random number generation being based on linear feedback shift registers. From mathematical view, the pseudo random number generation defined by one polynomial, which is applied iteratively and recursively on an initial sequence of bits (the seed bit sequence or seed number) in order to obtain pseudo random number comprising sequence of bits caused by the applied iterative and recursive methodology. The inventive concepts purposes to derive several polynomials from the one polynomial describing the feedback function, which polynomials are a function of the initial sequence of bits (the seed bit sequence or seed number) and which application results in the pseudo random number, which would be also obtainable by the aforementioned iterative and recursive methodology.

With reference to Fig. 3b, a generalized embodiment of the present invention is illustrated. Assuming a number of n bits forming the initial sequence of bits (the seed bit sequence or seed number) depicted as initial storage 1, the resulting total random number can comprise a maximum of $2^{np} - 1$ bits before repetition of the bit sequence occurs, wherein np denotes the order of the original polynomial. Consequently, $2^{np} - (n + 1)$ (10 bits are seed or initial bits) static polynomials can be derived in accordance with the methodology proposed above. Hence, the result storage 2 illustrated in Fig. 3b comprises m one-bit storage cells; the maximum m_{max} of cells is equal to the maximum of derivable polynomials; i.e. $m_{max} = 2^{np} - (n + 1)$. Applying these general remarks to the embodiment shown in Fig. 3a, the initial bit sequence comprises n = 10 bits in correspondence with the order of the original polynomial. This means, $2^{10} - 11 = 1013$ static polynomials can be derived and can be calculated parallel to obtain the complete sequence of 1023 bits (comprising the ten initial bits). As aforementioned, the combinatorial logic 10 of Fig. 3b can be implemented as a hardware combinatorial logic or as a software module comprising combinatorial logic relationships in accordance with the polynomials defined above.

The application of the aforementioned polynomials to obtain pseudo random numbers enables to generate the pseudo random numbers in reverse order in comparison with the iterative and recursive method. Firstly, lower bits of the pseudo random numbers can be obtained and subsequently higher bits of the pseudo random numbers are gained.

The integration of a combinatorial logic within an integrated circuit as exemplary illustrated in Fig. 3a enables to simultaneously obtain a sequence of bits for forming a pseudo random number comprising this sequence and eventually the initial sequence of bits (i.e. the seed bits). The integrated circuit comprises for instance a number of n storage cells each capable for storing a one-bit value. The n storage cells serve to store the initial bit values or seed values, which n storage cells are represented exemplarily in Fig. 3a by the cells x^1 to x^{10}. The integrated circuit comprises additionally for instance a number of m storage cells each also capable for storing a one-bit value, which m storage cells are represented exemplarily in Fig. 3a by the cells x^0 to x^{-33}. Each input of the m storage cells is coupled to a combinatorial logic in accordance with the polynomials described above. Each combinatorial logic receives its input values from output terminals of the corresponding selection of the n storage cells defined by the corresponding polynomial.

With reference to the embodiment illustrated in Fig. 3a, the value of the cell x^0 is determined by the values of the cells x^3 and x^{10}, which values are combined by a XOR logic relationship or XOR logic component. The cell x^{-33} is defined by the values of the cell x^{10}, x^9, x^5, x^4, x^3, x^2, and x^1, which are each combined by XOR logic relationship or XOR logic component. Corresponding combinatorial logics are obtained from the further polynomials. Those skilled in the art will appreciate that the maximal number m_{max} of storage cells is equal to the maximal number of derivable polynomials, which is equal to $2^n - (n + 1)$ as defined above. In case the number of m storage cells is smaller that the maximal number m_{max}, the cell values of the lower cells, i.e. the cells having the lower indices, may be used as initial values and seed values, respectively. Hence, the maximal sequence of bits can be always obtained even when implementing a limited number of storage cells and combinatorial logic elements.

Those skilled in the art will appreciate that the embodiments illustrated in Figs. 3a and 3b and described with reference thereto may be modified. In particular, the aforementioned embodiments include storage cells to store the initial (seed) values and the result values caused by the combinatorial logic. A modification of these embodiments may manage the inventive concept without storage cells. This means, an alternative (modified) embodiment comprises a combinatorial logic as described above, which is supplied with a number of n signals and states, respectively, representing the initial (seed) values and being used as such. Consequently, the

WAVELYNX EX. 1003, Pg. 583
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

combinatorial logic causes a number of m signals and states, respectively, which represent the result values caused by the combinatorial logic fed with the initial states.

As briefly mentioned in the introduction to this invention, pseudo random numbers are widely applied. In particular, GPS receivers and CDMA transmission technology make extensive use of pseudo random numbers.

Figs. 4a and 4b illustrate schematically block diagrams of a base station CDMA transmitter and a GPS receiver, which both take use of pseudo random number generators for operation.

With reference to Fig. 4a, each voice conversation is converted into digital code (with the help of an analog-to-digital converter ADC 110) and encoded by the means of a voice encoder or vocoder 120. The vocoder output is supplied to a convolutional encoder 130 that adds redundancy for error correction and each bit is replicated 64 times (not shown). Further, the resulting bit sequence is XOR-ed with a Walsh code provided by a Walsh code generator 140, which Walsh code is used to identify that call from the rest and output of the XOR-ed bit sequence is again XOR-ed with a string of pseudo random bits from a pseudo random generator 150, which string of bits is used to identify all the calls within a particular cell sector. All the calls at the base station are combined and modulated onto a carrier frequency at a combiner and modulator 160 and transmitted via the antenna 170 to the mobile stations. At a receiving mobile station, the received signals are quantized and fed through a Walsh code and pseudo random number sequence correlation receiver to reconstruct the transmitted bits of the original signals dedicated to the corresponding mobile station.

With reference to Fig. 4b, a GPS signal on the L1 (1575.42 MHz) or GPS signals on the L1 and L2 (1227.60 MHz) carrier frequencies is received via the antenna 200 and supplied to an amplifier 210. The GPS signal on the L2 carrier frequency is conventionally scrambled and dedicated for exclusive military use. The GPS signal carried on the L1 carrier frequency is known as a coarse acquisition signal and is composed of a 1.023 MHz pseudo random sequence signal and a 50 Hz navigation and system data signal both modulated onto the L1 carrier frequency.

The 1 MHz pseudo random sequence signal is used for determining the time of flight of the GPS signal emitted by a GPS satellite and received by the GPS receiver. The GPS receiver is informed about the seed value used by the GPS satellite for generating the 1.023 MHz pseudo random sequence signal. The C/A code generator represents a pseudo random number generator 230 by the means of which the GPS receiver generates the same pseudo random sequence signal employing the known seed value. Due to the finite time of flight of the GPS signal the received pseudo random sequence signal and the generated pseudo random sequence signal differ about a phase shift in relation to each other. The phase shift can be determined in form of a code of chip shift such that a pseudo range can be obtained, which approximates the distance between GPS satellite and GPS receiver.

Both examples, i.e. the CDMA transmitter as well as the GPS receiver, use pseudo random generators for operation. In case of the CDMA transmitter, the generation rate required for the employed pseudo random generator is defined by the data rate of the bit sequence to be XOR-ed with the pseudo random sequence. In case of the GPS receiver, the generation rate required for the employed pseudo random generator is defined by the pseudo random sequence signal generated and emitted by the GPS satellite. Those skilled in the art will appreciate that these two examples illustrate only two of many applications, which require fast and economical pseudo random generators. Especially, increasing data rates in mobile data communication technologies employing CDMA technology drives the requirement for such fast and economical pseudo random generators, which can be realized on the basis of the inventive concept illustrated in detail above in view of an embodiment.

Even though the invention is described above with reference to embodiments according to the accompanying drawings, it is clear that the invention is not restricted thereto but it can be modified in several ways within the scope of the appended claims.

Claims

1. Method for generating a pseudo random number, comprising:
   - providing a plurality of m polynomials, wherein said polynomials are derived from an original polynomial defining a feedback function of a linear feedback shift register, wherein said polynomials are functions of n bits serving as initial bits;
   - applying said polynomials on said initial bits for generating said pseudo random number comprising at least m bits resulting from said polynomials.

2. Method according to claim 1, comprising:
   - deriving said plurality of m polynomials from said original polynomial by
   - stepwise defining polynomials representing said feedback function at each iteration; and
   - reducing said polynomials to obtain polynomials being functions of said initial bits.

3. Method according to claim 1 or 2, comprising:
   - applying said polynomials substantially simultaneously.

4. Method according to claim 1 or 2, comprising:
   - applying said polynomials in reverse order to generate firstly lower bits of said pseudo random number and subsequently higher bits of said pseudo random number.

5. Method according to anyone of the preceding claims, wherein said polynomials represent logic relationships for combining at least two values each having a domain of one bit.

6. Method according to anyone of the claims 1 to 5, wherein said m polynomials are static polynomials, which can be provided fixedly.

7. Method according to anyone of the claims 1 to 5, wherein said m polynomials are dynamic polynomials, which are obtainable from said original polynomial by reduction of the orders of said polynomials equal or less than an order of said original polynomial.

8. Method according to anyone of the preceding claims, wherein said maximal number m_{max} of said polynomials is less or equal to $2^{np} - (n + 1)$, where np represents the order of the original polynomial.

9. Method according to anyone of the preceding claims, wherein an order of said polynomials is equal or less to np, wherein np represents the order of the original polynomial.

10. Method according to anyone of the preceding claims, wherein said pseudo random number comprising said m bits resulting from said polynomials and said n initial bits is identical with a sequence obtained from a linear feedback shift register used as a pseudo random number generator and applying a feedback function corresponding to said original polynomial.

11. Module for generating a pseudo random number, comprising:
    - a number of n initial storage cells each serving as a one-bit storage;
    - a number of m result storage cells each serving as a one-bit storage; and
    - a combinatorial logic being selectively coupled to output terminals of said n initial storage cells and being selectively coupled to input terminals of said m result storage cells, wherein said combinatorial logic implements a number of m polynomials, wherein said polynomials are derived from an original polynomial defining a feedback function of a linear feedback shift register, wherein said polynomials are functions of n bits serving as initial bits.

12. Module according to claim 11, wherein said combinatorial logic is implemented as a software module comprising code sections, which when executed on a processing unit perform logical relationship operations in accordance with said combinatorial logic.

13. Module according to claim 11, wherein said combinatorial logic is implemented by the means of a plurality of logic components, each having at least two input terminals for receiving one-bit inputs and each having at least one output terminal for providing a one-bit output, which output is defined by a predefined combinatorial logic relationship.

14. Electronic apparatus enabled for generating a pseudo random number, comprising at least a module for generating a pseudo random number, said module comprising:
    - a number of n initial storage cells each serving as a one-bit storage;
    - a number of m result storage cells each serving as a one-bit storage; and
    - a combinatorial logic being selectively coupled to output terminals of said n initial storage cells and being selectively coupled to input terminals of said m result storage cells, wherein said combinatorial logic implements a number of m polynomials, wherein said polynomials are derived from an original polynomial defining a feedback function of a linear feedback shift register, wherein said polynomials are functions of n bits serving as initial bits.

15. System for generating a pseudo random number, comprising:
    - a number of n initial states representing n initial 1-bit values;
    - a number of m result states representing m result 1-bit values; and
    - a combinatorial logic being selectively supplied with said n initial states and supplying selectively said m result states, wherein said combinatorial logic implements a number of m polynomials, wherein said polynomials are derived from an original polynomial defining a feedback function of a linear feedback shift register, wherein said polynomials are functions of n 1-bit states serving as initial states.

16. Computer program product for generating a pseudo random number, comprising program code sections for carrying out the steps of anyone of claims 1 to 10, when said program is run on a processor-based device, a terminal device, a network device, a portable terminal, a consumer electronic device, or a mobile communication enabled terminal.

17. Computer program product for generating a pseudo random number, comprising program code sections stored on a machine-readable medium for carrying out the steps of anyone of claims 1 to 10, when said program product is run on a processor-based device, a terminal device, a network device, a portable terminal, a consumer electronic device, or a mobile communication enabled terminal.

WAVELYNX EX. 1003, Pg. 588
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

18. Software tool for generating a pseudo random number, comprising program portions for carrying out the operations of any one of the claims 1 to 10, when said program is implemented in a computer program for being executed on a processor-based device, a terminal device, a network device, a portable terminal, a consumer electronic device, or a mobile communication enabled terminal.

19. Computer data signal embodied in a carrier wave and representing instructions, which when executed by a processor cause the steps of anyone of claims 1 to 10 to be carried out.

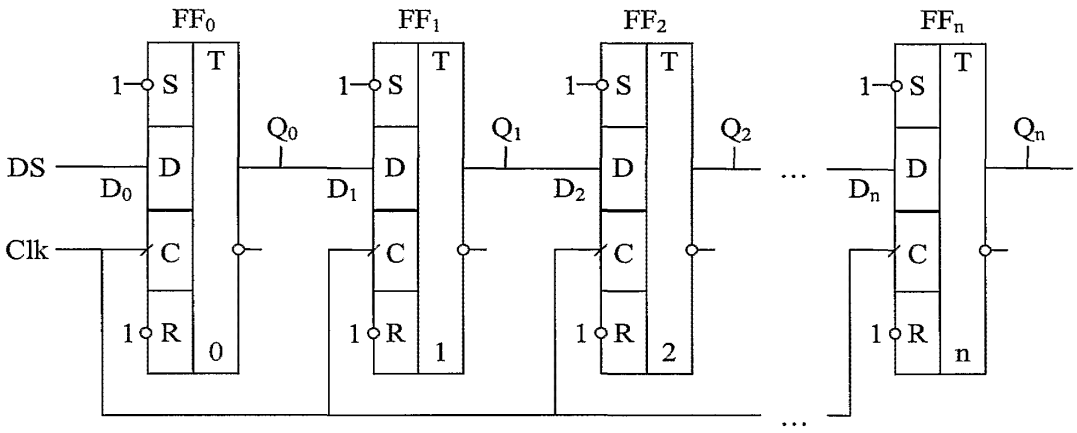

FF0
FF1
FF2
FFn
T
S
D
C
R
1
DS
Clk
Q0
Q1
Q2
Qn
D0
D1
D2
Dn
0
1
2
n
...

Fig. 1a

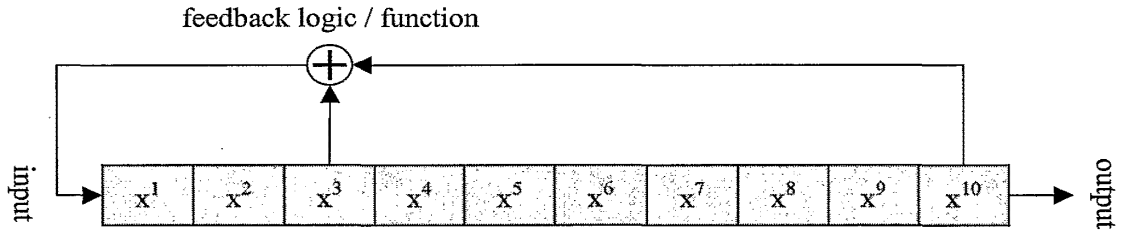

feedback logic / function
input
output
x1
x2
x3
x4
x5
x6
x7
x8
x9
x10

Fig. 1b

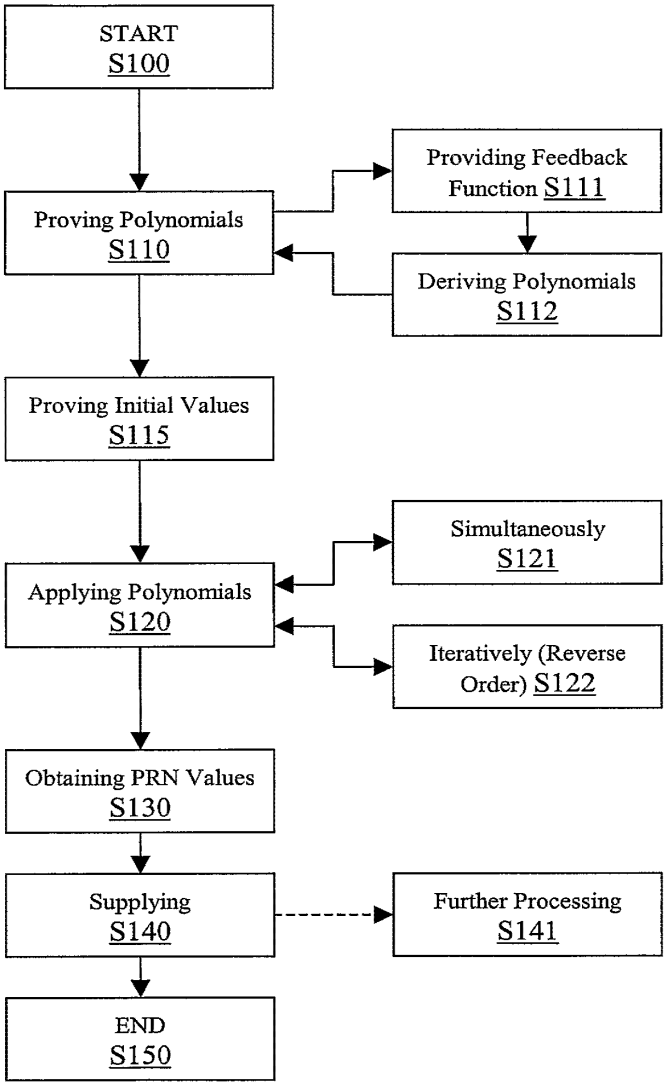

START
S100
Proving Polynomials
S110
Providing Feedback Function S111
Deriving Polynomials
S112
Proving Initial Values
S115
Applying Polynomials
S120
Simultaneously
S121
Iteratively (Reverse Order) S122
Obtaining PRN Values
S130
Supplying
S140
Further Processing
S141
END
S150

Fig. 2

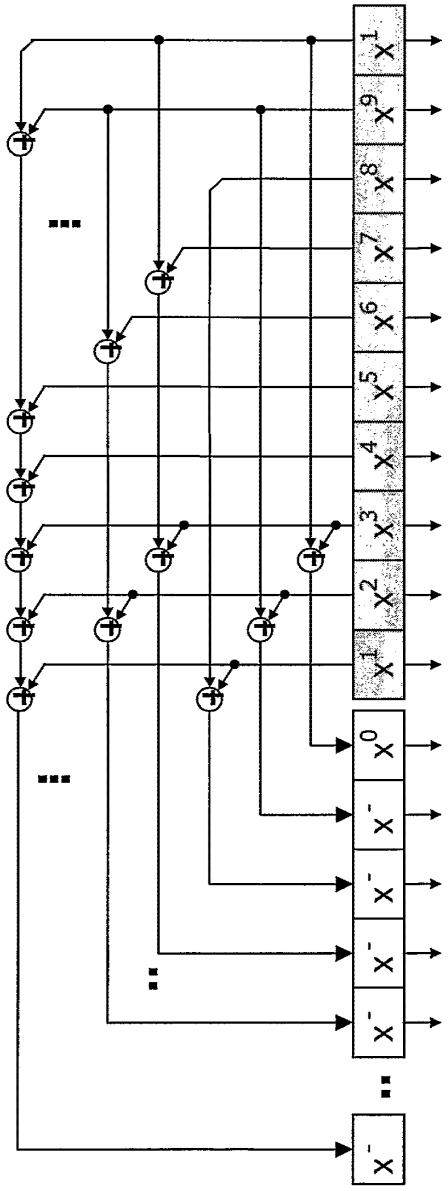


Fig. 3a

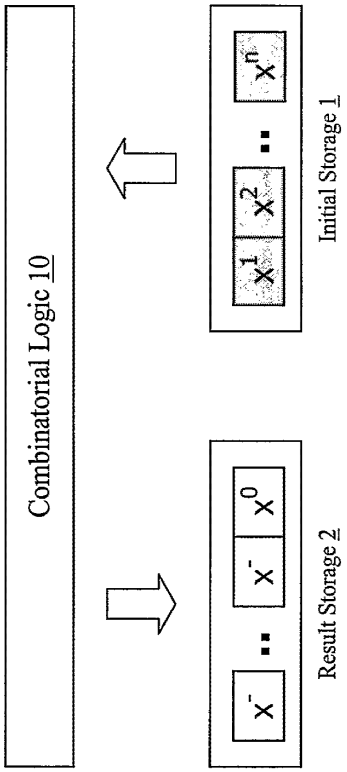


Fig. 3b

WAVELYNX EX. 1003, Pg. 592
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

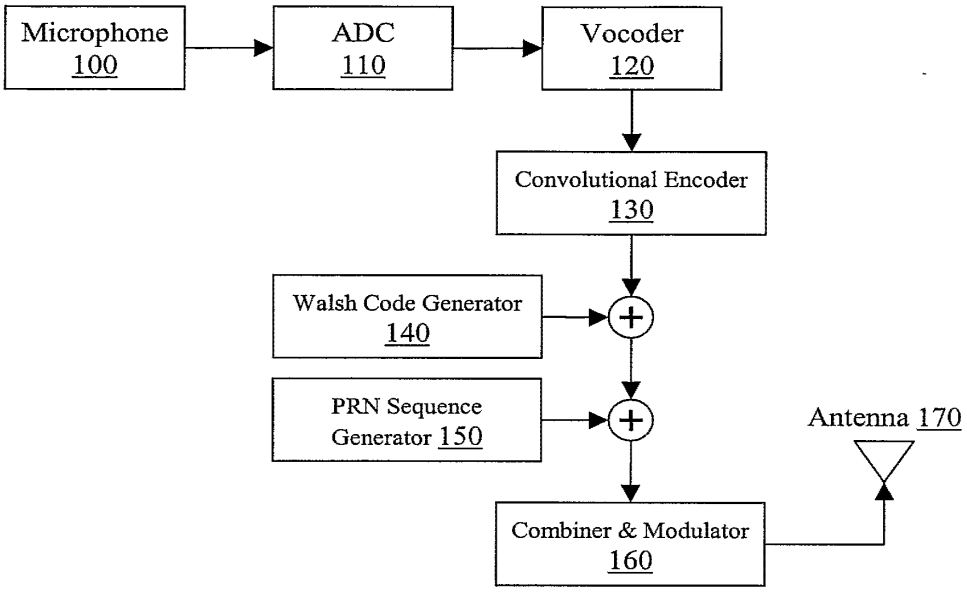

Microphone 100
ADC 110
Vocoder 120
Convolutional Encoder 130
Walsh Code Generator 140
PRN Sequence Generator 150
Antenna 170
Combiner & Modulator 160

Fig. 4a

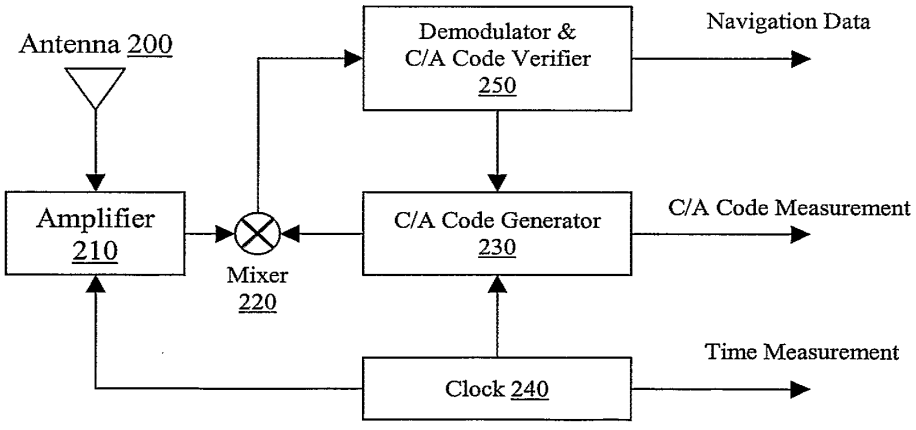

Antenna 200
Demodulator & C/A Code Verifier 250
Navigation Data
Amplifier 210
Mixer 220
C/A Code Generator 230
C/A Code Measurement
Time Measurement
Clock 240

Fig. 4b

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB 2004/003085

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: G06F 7/58, H04L 9/22, G11C 19/00
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: G06F, H04L, G11C

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI DATA, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 4965881 A (DILLEY, J E), 23 October 1990 (23.10.1990), column 5, line 49 - column 7, line 35	1-3,5-19
	--	
A	US 5046036 A (TEZUKA, S), 3 Sept 1991 (03.09.1991), column 1, line 9 - column 2, line 32	1-19
	--	
A	US 4493046 A (WATANABE, T), 8 January 1985 (08.01.1985), column 1, line 6 - column 2, line 53	1-19
	--	
A	US 5910907 A (CHEN, C-K ET AL), 8 June 1999 (08.06.1999), column 1, line 5 - column 2, line 29	1-19
	--	

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
15 April 2005	22-01-2005
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Antonio Farieta /LR Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (January 2004)

US	4965881	A	23/10/1990	NONE			
US	5046036	A	03/09/1991	DE	3580151	D	00/00/0000
				EP	0178432	A,B	23/04/1986
				JP	1660832	C	19/05/1992
				JP	3026859	B	12/04/1991
				JP	61097746	A	16/05/1986
US	4493046	A	08/01/1985	DE	3266649	D	00/00/0000
				EP	0066768	A,B	15/12/1982
				JP	57191974	U	06/12/1982
				JP	57194621	A	30/11/1982
US	5910907	A	08/06/1999	NONE			

WAVELYNX EX. 1003, Pg. 595
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 April 2006 (06.04.2006)

PCT

(10) International Publication Number
WO 2006/036521 A1

(51) International Patent Classification:
H04Q 7/32 (2006.01)

(21) International Application Number:
PCT/US2005/032337

(22) International Filing Date:
6 September 2005 (06.09.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/608,305	8 September 2004 (08.09.2004)	US
Not furnished	2 September 2005 (02.09.2005)	US

(71) Applicant *(for all designated States except US)*: **QUALCOMM INCORPORATED** [US/US]; 5775 Morehouse Drive, San Diego, CA 92121 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **SEMPLE, James** [CA/GB]; 7 Queensgate Place, Flat 4, London South Wales SW7 5NU (GB). **ROSE, Gregory Gordon** [AU/US]; 3234 North Star Drive, San Diego, CA 92117 (US). **PADDON, Michael** [AU/AU]; 31 Armine Way, Kellyville, NSW 2155 (AU). **HAWKES, Philip Michael** [AU/AU]; 18-20 Knocklayde St, Unit 30, Ashfield, NSW 2131 (AU).

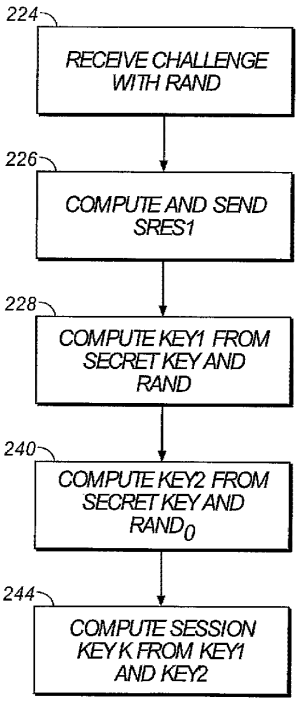
(74) Agents: **WADSWORTH, Philip R.** et al.; 5775 Morehouse Drive, San Diego, CA 92121 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

[Continued on next page]

(54) Title: BOOTSTRAPPING AUTHENTICATION USING DISTINGUISHED RANDOM CHALLENGES



(57) Abstract: A communications system and method of bootstrapping mobile station authentication and establishing a secure encryption key are disclosed. In one embodiment of the communications network, a distinguished random challenge is reserved for generation of a secure encryption key, wherein the distinguished random challenge is not used for authentication of a mobile station. The distinguished random challenge is stored at a mobile station's mobile equipment and used to generate a secure encryption key, and a bootstrapping function in the network uses a normal random challenge to authenticate the mobile station and the distinguished random challenge to generate the secure encryption key.

WO 2006/036521 A1

European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

BOOTSTRAPPING AUTHENTICATION USING DISTINGUISHED RANDOM CHALLENGES

Claim of Priority under 35 U.S.C. § 119

[0001] This application claims priority to U.S. Provisional Patent Application No. 60/608,305 entitled "BOOTSTRAPPING GSM AUTHENTICATION AND DISTINGUISHED RANDS" and filed on September 8, 2004. The disclosure of the above-described filed application is hereby incorporated by reference in its entirety.

BACKGROUND

Field

[0002] The application relates generally to authentication in cellular communication networks, and more particularly to the derivation of encryption keys for application security.

Background

[0003] Mobile communication applications generally share a need for authentication of a subscriber (user equipment or mobile station) by a communication server before communication is initiated or a transaction is carried out. One authentication mechanism is based on a secret shared between the communicating entities, and there are a number of authentication protocols that rely on this pre-shared secret.

[0004] In a mobile communications network based on the Global System for Mobile Communications (GSM), for example, the identity of a subscriber is authenticated before the subscriber is allowed to access the communications network. In order for a subscriber's mobile station (or user equipment UE) to establish a communication session with a network element, the mobile station authenticates itself to the network element by responding to a random number challenge. The random number challenge and a shared secret key are used to establish a session encryption key for encrypting communication transmissions between the mobile station and the network element.

[0005] The communications system features described herein can be implemented in a variety of communications networks requiring authentication and encrypted communication between communicating entities. Figure 1 is a block diagram of the communication network entities involved in authentication of a subscriber in a GSM

network. A subscriber's mobile station 30 comprises a secure IC 32 and mobile equipment (ME) 34 (e.g., a cellular telephone handset). The mobile equipment 34 includes a processor 36 configured to perform authentication functions at the mobile station 30 in conjunction with the secure IC 32.

[0006] Stored on the secure IC 32 is subscriber identity and subscription related information, information for performing authentication functions with the communications network, an International Mobile Subscriber Identity (IMSI), preferred language, and IC card identification. The secure IC may be referred to as a SIM card or a smart card. Also stored at the secure IC 32 is a secret key Ki 38 which is used to authenticate the mobile station 30 to a network element 40 of the serving network for access to the network. The secret key Ki 38 is also stored at the mobile subscriber's home network at an authentication center (AuC) 42. The authentication center 42 uses the secret key Ki 38 to generate authentication data specific to the subscriber using the secret key Ki 38, and sends the authentication data to the network element 40.

[0007] An authentication and key generation process for mobile station authentication and encrypted communication is illustrated in Figures 1-3, wherein Figure 2 is a flow diagram illustrating a method of authentication and encryption key generation at the mobile station 30, and Figure 3 is a signal flow diagram illustrating a method of mobile station authentication and encryption key generation in the communications network. In reference to Figure 3, the mobile station 30 requests a communication session with a network element 40 in a step 102. If the network element 40 does not already have security information stored for that subscriber to authenticate the mobile station 30, the network element 40 sends a request for security information to the authentication center 42 in the mobile station's home network in a step 104. In response to the security information request, the authentication center 42 generates one or more authentication vectors comprising a random number challenge RAND, an expected authentication response XRES, and a encryption key Kc. The expected response XRES and the encryption key Kc are determined based on the RAND and the secret key Ki 38. In a step 108, the authentication center 42 sends the authentication vector(s) (RAND, XRES, Kc) to the network element 40.

[0008] The network element 40 selects an authentication vector (RAND, XRES, Kc) to use in authenticating the identity of the mobile station 30 and sends the random challenge RAND of the selected authentication vector to the mobile station 30 in a step

112. Referring to Figure 2, the mobile station 30 receives the authentication challenge with the challenge RAND in step 112, and computes and sends an authentication response in a step 114. The mobile station 30 also computes a session key in a step 115 using the secret key Ki 38 and RAND.

[0009] To produce the response and the session key, the mobile equipment 34 at the mobile station 30 passes the RAND to the secure IC in a step 113. In steps 114 and 115, the secure IC 32 computes a set of one or more values using the received random challenge RAND and the stored secret key Ki. These values generally include an authentication response SRES as shown in step 114. In step 115, the secure IC 32 computes a second value comprising a session encryption key Kc using the received random challenge RAND, the stored secret key Ki 38. In a step 116, the secure IC 32 sends the generated response SRES and the encryption key Kc to the mobile equipment 34 in a step 116. The mobile equipment 34 sends the generated authentication response SRES to the network element 40 in a step 117, and stores the key Kc at the mobile equipment in a step 118. The network element 40 compares the mobile station generated authentication response SRES to the expected response XRES of the selected authentication vector in a step 119. If the authentication parameters do not match, the authentication procedure is terminated. If the parameters do match, the mobile station 30 is considered authenticated in a step 120 and the network element 40 begins communication with the mobile unit using the encryption key Kc in step 122.

[0010] GSM authentication and key agreement procedures are subject to replay and cryptanalytic attack. For example, the conventional algorithms used by the GSM system to encrypt communications are weak. Methods have been devised to determine the encryption key Kc and determine the contents of a subscriber's communications. There is therefore a need in the art for a method of improving application security using the current capabilities of deployed mobile stations, especially as mobile communications become used for more sensitive data or require stronger authentication.

SUMMARY

[0011] In one aspect, the invention includes a mobile station configured for communicating in a wireless communications network. The mobile station comprises a receiver configured to receive at least one authentication data parameter from the wireless communications network and a memory storing a fixed authentication data

parameter. A first processing circuit is configured to generate a first key based on the at least one received authentication data parameter, and to generate a second key based on the fixed authentication data parameter. A second processing circuit is configured to generate a third key using at least the first and second keys.

[0012] In another aspect, a mobile element of a wireless communications network is provided. The wireless communications network comprises a plurality of mobile elements and a plurality of network elements communicating with the mobile elements. The mobile element is configured to authenticate itself to the communications network by responding to a challenge value presented to the mobile element by a network element of the communications network during an authentication procedure. Furthermore, the mobile element comprises a memory storing a reserved challenge value that is not used to authenticate mobile elements in authentication procedures between any network element and any mobile element.

[0013] In another aspect, the invention includes a method of communication between a mobile station and a communications network element. The method includes selecting an authentication challenge at a network element and transmitting the authentication challenge to a mobile station. The method further includes generating a first value comprising an authentication response at the mobile station using at least the authentication challenge and a stored key; generating a second value at the mobile equipment using at least the authentication challenge and the stored key; generating a third value at the mobile equipment using at least a fourth value different from the authentication challenge and the stored key; and generating a key using at least the second and third values.

[0014] In another aspect, a method of creating keys in a communication network that uses a challenge-response authentication procedure comprises reserving at least one challenge value for use in generating session keys for use in communication between mobile units and network elements within the communication network. The reserved challenge value is not used for mobile unit authentication.

[0015] In another aspect, a method of generating a key at a mobile station for securing communication between the mobile station and a network element is provided. In this aspect, the method includes receiving an authentication challenge value from a network element at the mobile station and sending the authentication challenge value to a processing circuit. The method further includes generating a first set of one or more

values using at least the authentication challenge value, sending at least one value from the first set to the network element for authentication. The method continues by sending a second authentication challenge value to the processing circuit and generating a second set of one or more values using at least the second authentication challenge. A key is generated using at least one value of the first set and at least one value of the second set.

[0016] In another aspect, a mobile station in a communications network includes means for receiving an authentication challenge value from the communications network, means for generating a first set of values in response to the received authentication challenge, means for generating a second set of values in response to a distinguished authentication challenge value, and means for generating a key using at least one of the first set of values and at least one of the second set of values.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] FIG. 1 is a block diagram of the communication network entities involved in authentication of a subscriber for communication in a GSM network;

[0018] FIG. 2 is a flow chart illustrating an authentication and key generation process performed at a mobile station according to GSM;

[0019] FIG. 3 is a signal flow diagram illustrating an authentication and key agreement procedure for authentication of a subscriber to a network element in GSM;

[0020] FIG. 4 is a block diagram of one embodiment of the communication network entities involved in authentication of a subscriber using distinguished authentication data;

[0021] FIG. 5 is a flow chart illustrating one embodiment of an authentication and distinguished key generation process performed at a mobile station using distinguished authentication data; and

[0022] FIG. 6 is a signal flow diagram illustrating one embodiment of a method of establishing a secure communication session between a mobile station and a communications network using distinguished authentication data.

DETAILED DESCRIPTION

[0023] As discussed above, the GSM encryption algorithms A5/1 and A5/2 are subject to attack, and methods have been found to obtain knowledge of an encryption key and thereby obtain unauthorized information from the mobile station 30. Thus, an improved authentication and key generation procedure is herein described, wherein the authentication and key generation procedure is implemented in one embodiment wherein the functions performed by a mobile subscriber's secure IC 32 remain the same as in the procedure illustrated in Figures 2-3, but the functions performed by the mobile equipment ME are different. Specifically, embodiments of the authentication and key generation procedure described herein can be implemented in new mobile station terminals using already deployed secure IC's 32, in order to derive keys for use in application security which are not compromised by the weaknesses of the GSM radio interface encryption.

[0024] Figure 4 is a block diagram of one embodiment of the communication network entities involved in authentication of a subscriber using distinguished authentication data. The communications network illustrated in Figure 4 comprises a mobile station 202, which is similar to the mobile station 30 of Figure 1, wherein the mobile station 202 of Figure 3 comprises the secure IC 32 which stores the secret key 38. However, mobile equipment 204 of the mobile station 202 is different from the mobile equipment 34 of the mobile station 30 of Figure 1 in that mobile equipment 204 stores distinguished or reserved authentication data in its memory, such as a distinguished random number challenge RAND 206. The mobile equipment 204 also comprises a processor 208.

[0025] The mobile equipment 204 uses the distinguished RAND 206 to generate a second set of values in addition to the set of values produced in response to a RAND received from the network as part of an authentication process. The mobile station computes a "distinguished" session key K using values produced from the challenge RAND received from the network and values produced from the distinguished RAND stored in the mobile station. The distinguished RAND has a predetermined fixed value known to the network and the mobile device. It may, for example, have an all-zero value, and is designated herein as $RAND_0$. The authentication center 42 also stores the distinguished RAND so that the network can also compute the distinguished key K. The distinguished key K can be used for a variety of purposes after it is generated, including

encrypting or keying a message authentication code in future communications, transactions, or the like. It may be used to secure communications between the mobile station 202 and a network element for applications requiring increased security, such as banking applications, over a variety of bearers such as GPRS, Bluetooth or WLAN. The distinguished RAND is reserved by the system for use in generating the distinguished key K, and is not used for initial authentication procedures, so that neither $RAND_0$ or the signed response to $RAND_0$ ($SRES_0$) are transferred over the wireless communication link.

[0026] Figures 5-6 illustrate an authentication and secure key generation process for the network entities of Figure 4, wherein Figure 5 is a flow diagram illustrating one embodiment of a method of authentication and secure key generation at the mobile station 202, and Figure 6 is a signal flow diagram illustrating one embodiment of an authentication and secure key generation process for establishing secure communication in a network.

[0027] In reference to Figure 6, the network element 40 obtains authentication data specific to the subscriber from the authentication center (AuC) 42 in the subscriber's home network. In a step 214, the authentication center 42 uses a random challenge RAND and the secret key Ki to generate one or more authentication vectors (RAND, XRES, Kc). The authentication center 42 also generates one or more distinguished authentication vectors ($RAND_0$, $XRES_0$, Kc_0) using the distinguished random challenge $RAND_0$ 206 and the secret key Ki. In a step 216, the authentication center computes a distinguished session key K using Kc, Kc_0, XRES, and $XRES_0$. In one embodiment, the distinguished key K is a hash of Kc, Kc_0, XRES, and $XRES_0$. In a step 220, the authentication center 42 sends both the authentication vector(s) and distinguished authentication vector(s) to the network element 40. With this information, the network element can also compute the distinguished key K based on the information provided by the authentication center.

[0028] As will be appreciated by those skilled in the art, the distinguished key K may be generated based on a plurality of combinations of values and is not limited to those described herein. For example, the distinguished key K may be generated based on RAND and $RAND_0$ in addition to or in place of Kc, Kc_0, XRES, and $XRES_0$. Also, a variety of variants may be used to provide the network element with the information necessary to communicate with the mobile station using the key K. The network

element 40 may directly receive the hash value forming the distinguished key K from the authentication center rather than the above described distinguished authentication vector. Alternately, the network element could keep a database of $XRES_0$ and Kc_0 for different subscriber identities (e.g. the IMSI).

[0029] Figure 4 is a signal flow diagram illustrating one embodiment of a bootstrapped method of authenticating a mobile subscriber using distinguished authentication data. Some of the procedures performed by the network elements according to the method illustrated in Figure 4 are similar to the procedures performed according to the method illustrated in Figure 2.

[0030] For authentication of the mobile station and generation of a session key, the network element 40 sends an authentication request to the mobile subscriber's mobile equipment 204 in a step 220, wherein the authentication request comprises only the random number challenge RAND, and the distinguished $RAND_0$ is not transmitted over the radio network from the network element 40 to the mobile station 202. In reference to Figure 5, the authentication and session key generation process performed at the mobile station 202 begins in step 224 in which the mobile equipment 204 receives the authentication challenge with the challenge RAND. In a step 226, the secure IC 32 computes the authentication response SRES1 and the mobile equipment 204 sends the response to the network element 40. In a step 228, the secure IC computes a first key KEY1 using the stored secret key Ki and the challenge RAND. In a step 240, the secure IC 32 computes a second key KEY2 using the distinguished challenge $RAND_0$ stored at the mobile equipment 204 and the secret key Ki stored at the secure IC 32. In a step 244, the mobile equipment 204 computes a session key K from KEY1 and KEY2. This key may be used in future communications or transactions.

[0031] The authentication and key generation process performed at the secure IC 32 and mobile equipment 204 is illustrated in more detail in reference to the network element 40 in Figure 6. Upon receipt of the random number challenge RAND, the mobile equipment 204 sends RAND to the secure IC 32 in step 224. In step 226, the secure IC generates the authentication or signed response SRES using RAND and the secret key Ki 38, and the secure IC computes the cipher key Kc in step 228 using the RAND and the secret key Ki. The secure IC 32 sends both the authentication response SRES and the cipher key Kc to the mobile equipment 204 in a step 230, and the mobile equipment 204 transmits the authentication response SRES to the network element 40 in a step 230.

The network element 40 compares the authentication response SRES from the mobile equipment 204 to the expected response XRES in the selected authentication vector in a step 234, similar to step 119 in Figure 3.

[0032] In a step 236, the mobile equipment 204 sends the distinguished random challenge $RAND_0$ 206, stored at the mobile equipment 204, to the secure IC 32, which computes a distinguished authentication response $SRES_0$ based on the distinguished $RAND_0$ in step 238, similar to step 226, using the secret key Ki. The secure IC 32 also computes the distinguished cipher key Kc_0 in step 240 using the secret key Ki. The secure IC 32 then transmits the distinguished authentication response $SRES_0$ and distinguished cipher key Kc_0 to the mobile equipment 204 in a step 242. Thus, the same secure IC used in the authentication process of Figure 3 can be used to generate the distinguished cipher key Kc_0 according to the embodiment of the invention illustrated in Figure 6.

[0033] In response to receipt of the distinguished authentication response $SRES_0$ and distinguished cipher key Kc_0, the mobile equipment 204 generates a distinguished session key K in step 244. In one embodiment, the distinguished key K is generated based on the cipher key Kc and authentication response SRES generated by the secure IC 32 in steps 226 and 228 using RAND, and the distinguished cipher key Kc_0 and distinguished authentication response $SRES_0$ generated by the secure IC 32 in steps 238 and 240 using $RAND_0$. The mobile equipment stores the distinguished key K in a step 246. With the distinguished key K stored at both the mobile equipment 204 and the network element 40, the key K can be used in future communications and transactions. In some embodiments, the mobile equipment 204 is configured to reject an authentication request including the distinguished RAND value to ensure that the signed response to the reserved RAND value is never sent over the wireless communication link and the resulting enciphering key Kc0 is not used to encrypt over the radio link.

[0034] Thus, according to the authentication and key generation process illustrated in Figures 4-6, a key K is agreed for use by applications, re-using the existing GSM SIM, Authentication Center, and interface between the mobile terminal and SIM, but the key is not exposed by security weaknesses in the GSM air interface.

[0035] In one embodiment, the mobile equipment 204 is configured to generate a distinguished authentication response DRES to replace the authentication response SRES for the authentication of the identity of the mobile station 202 to the network

element 40. For example, the mobile equipment 204 may be configured to generate a distinguished key DRES based on XRES, XRES0, Kc, Kc0. In such an embodiment the network element 40 either receives an expected distinguished authentication response DRES which is generated at the authentication center 42, or the network element 40 is configured to generate the expected distinguished response DRES based on the received parameters SRES, SRES0, Kc and Kc0. The network element 40 is further configured to compare the distinguished authentication data DRES generated by the mobile equipment 204 to the expected distinguished response for authentication of the mobile station 202.

[0036] In some embodiments, the authentication and key generation process discussed in reference to Figures 4-6 further employs a bootstrapping function in the mobile stations' home network for bootstrapping the authentication and key generation process. The bootstrapped process may be used for authentication and key generation in connection with communication sessions requiring heightened security, such as between the mobile station and an e-commerce network application function. In such embodiments, the mobile station 202 performs the authentication and key generation process in connection with the bootstrapping function instead of the network element 40, wherein the bootstrapping function receives the authentication vectors and session key K from the authentication center 42. Following authentication of the mobile station, the bootstrapping function then sends the session key K to the e-commerce network application function for encrypting communications with the mobile station.

[0037] In a communications network employing the bootstrapped process, the mobile equipment 204 may be configured to perform the authentication and key generation process with the secure IC 32 illustrated in Figures 2-3 for voice calls, wherein the cipher key Kc is used to encrypt communications with a network element. The mobile equipment 204 may further be configured to recognize a communication session type requiring increased security, such as e-commerce, and accordingly perform the authentication and key generation process with the secure IC 32 illustrated in Figures 5-6, wherein the session key K is used to encrypt communications. Regardless of the authentication and key generation process performed by the mobile equipment 204, the process performed by the secure IC 32 remains the same by receiving a random challenge and computing both a signed response and a cipher key based on the random challenge and the stored secret key Ki.

[0038] An exemplary implementation of the authentication and key generation process illustrated in Figures 4-6 is a communication session between a mobile station and a banking institution, wherein a mobile subscriber desires to exchange sensitive information with a network application and therefore desires increased communication security. In the present example, a mobile station requests communication with the bank network application function by transmitting a request to the bootstrapping function in the mobile station's home network. The bootstrapping function obtains standard (RAND, XRES, Kc) and distinguished authentication vectors ($RAND_0$, $XRES_0$, Kc_0) and a session key K from the authentication center for use in authenticating the identity of the mobile station requesting communication. The bootstrapping function sends the random challenge RAND to the mobile station's mobile equipment. The mobile equipment sends the random challenge RAND to its secure IC for computation of the response SRES and cipher key Kc. In response to receipt of the response SRES and cipher key Kc from the secure IC, the mobile equipment sends the response SRES to the bootstrapping function, which determines whether the key used to generate SRES is the same as the key used to generate the expected response XRES by comparing SRES to XRES. If the two parameters do match, the authentication of the mobile station is considered successful and the bootstrapping function sends the session key to the network application function (bank).

[0039] After sending the generated response SRES to the bootstrapping function, the mobile equipment sends the distinguished $RAND_0$, stored at the mobile equipment, to the secure IC for computation of the distinguished response $SRES_0$ and distinguished cipher key Kc_0. The mobile equipment then uses Kc, SRES, Kc_0, $SRES_0$ to compute the distinguished session key K. The mobile station and the network application function can then begin secure communications using the distinguished session key K to encrypt their communication transmissions.

[0040] As will be appreciated by those skilled in the art, the above-described systems and methods are directed to only a few specific embodiments, and the invention can be practiced in many ways. Those of skill in the art would understand that information and signals may be represented using any of a variety of different technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description may be

represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof.

[0041] Those of skill would further appreciate that the various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present invention.

[0042] The various illustrative logical blocks, modules, and circuits described in connection with the embodiments disclosed herein may be implemented or performed with a general purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general purpose processor may be a microprocessor, but in the alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

[0043] The steps of a method or algorithm described in connection with the embodiments disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary storage medium is coupled to the processor such the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the

processor. The processor and the storage medium may reside in an ASIC. The ASIC may reside in a user terminal. In the alternative, the processor and the storage medium may reside as discrete components in a user terminal.

[0044] The previous description of the disclosed embodiments is provided to enable any person skilled in the art to make or use the present invention. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without departing from the spirit or scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

What is claimed is:

CLAIMS

1. A mobile station for communicating in a wireless communications network, comprising:

a receiver configured to receive at least one authentication data parameter from said wireless communications network;

a memory storing a fixed authentication data parameter;

a first processing circuit configured to generate a first key based on said at least one received authentication data parameter, and to generate a second key based on said fixed authentication data parameter; and

a second processing circuit configured to generate a third key using at least said first and second keys.

2. The mobile station of Claim 1, wherein said second processing circuit is configured to encrypt communication transmissions using said third key.

3. The mobile station of Claim 1, wherein the at least one received authentication data parameter comprises a random or pseudo-random number that changes for different communication sessions between said mobile station and said communications network.

4. The mobile station of Claim 3, wherein the mobile equipment is configured to reject an authentication request containing said fixed authentication data parameter.

5. The mobile station of Claim 1, wherein the first processing circuit is configured to generate an authentication response using at least one received authentication data parameter and a secret key.

6. The mobile station of Claim 5, wherein the mobile station is configured to send the authentication response over said wireless communication channel for authentication of the mobile station.

7. The mobile station of Claim 1, wherein the first processing circuit comprises a secure integrated circuit.

8. The mobile station of Claim 7, wherein the first integrated circuit comprises a subscriber identity module (SIM).

9. A mobile element of a wireless communications network, said wireless communications network comprising a plurality of mobile elements and a plurality of network elements communicating with said mobile elements, wherein said mobile element is configured to authenticate itself to said communications network by responding to a challenge value presented to the mobile element by a network element of said communications network during an authentication procedure, and wherein said mobile element comprises a memory storing a reserved challenge value that is not used to authenticate mobile elements in authentication procedures between any network element and any mobile element.

10. The mobile element of Claim 9, comprising a processing circuit configured to generate sets of one or more values in response to challenge values using at least a stored key.

11. The mobile element of Claim 10, wherein said processing circuit is configured to generate a signed response and an encryption key in response to a challenge value.

12. The mobile element of Claim 11, wherein said processing circuit comprises a subscriber identity module (SIM).

13. The mobile element of Claim 12, wherein said mobile unit is configured to communicate in a GSM communications network.

14. The mobile element of Claim 12, wherein said mobile unit is configured to communicate in a local area network.

15. A method of communication between a mobile station and a communications network element, comprising:

selecting an authentication challenge at a network element;

transmitting the authentication challenge to a mobile station;

generating a first value comprising an authentication response at the mobile station using at least the authentication challenge and a stored key;

generating a second value at the mobile equipment using at least the authentication challenge and said stored key;

generating a third value at the mobile equipment using at least a fourth value different from said authentication challenge and said stored key; and

generating a key using at least said second and third values.

16. The method of Claim 15, wherein the first authentication challenge comprises a random or pseudo-random number.

17. The method of Claim 15, comprising transmitting said authentication challenge from an authentication center to said network element.

18. The method of Claim 15, comprising transmitting said key from an authentication center to said network element.

19. The method of Claim 15, wherein said fourth value comprises a reserved authentication challenge.

20. The method of Claim 15, further comprising rejecting an authentication request at the mobile station when the authentication challenge is the same as said fourth value.

21. A method of creating keys in a communication network that uses a challenge- response authentication procedure, said method comprising reserving at least one challenge value for use in generating session keys for use in communication between mobile units and network elements within the communication network, wherein the reserved challenge value is not used for mobile unit authentication.

22. A method of generating a key at a mobile station for securing communication between said mobile station and a network element, said method comprising:

receiving an authentication challenge value from a network element at said mobile station;

sending said authentication challenge value to a processing circuit;

generating a first set of one or more values using at least said authentication challenge value;

sending at least one value from said first set to the network element for authentication;

sending a second authentication challenge value to said processing circuit;

generating a second set of one or more values using at least said second authentication challenge;

generating said key using at least one value of said first set and at least one value of said second set.

23. The method of Claim 22, wherein a value of said first set of values sent to a network element comprises a signed response (SRES).

24. The method of Claim 22, wherein one of said first set of values comprises an encryption key.

25. The method of Claim 22, wherein one of said second set of values comprises an encryption key.

26. The method of Claim 22, wherein comprising generating, with said processing circuit, a signed response and an encryption key for each authentication challenge value sent to said processing circuit.

27. The method of Claim 22, wherein the first authentication challenge is a random or pseudo-random number.

28. A mobile station in a communications network, said mobile station comprising:

means for receiving an authentication challenge value from said communications network;

means for generating a first set of values in response to said received authentication challenge;

means for generating a second set of values in response to a distinguished authentication challenge value; and

means for generating a key using at least one of said first set of values and at least one of said second set of values.

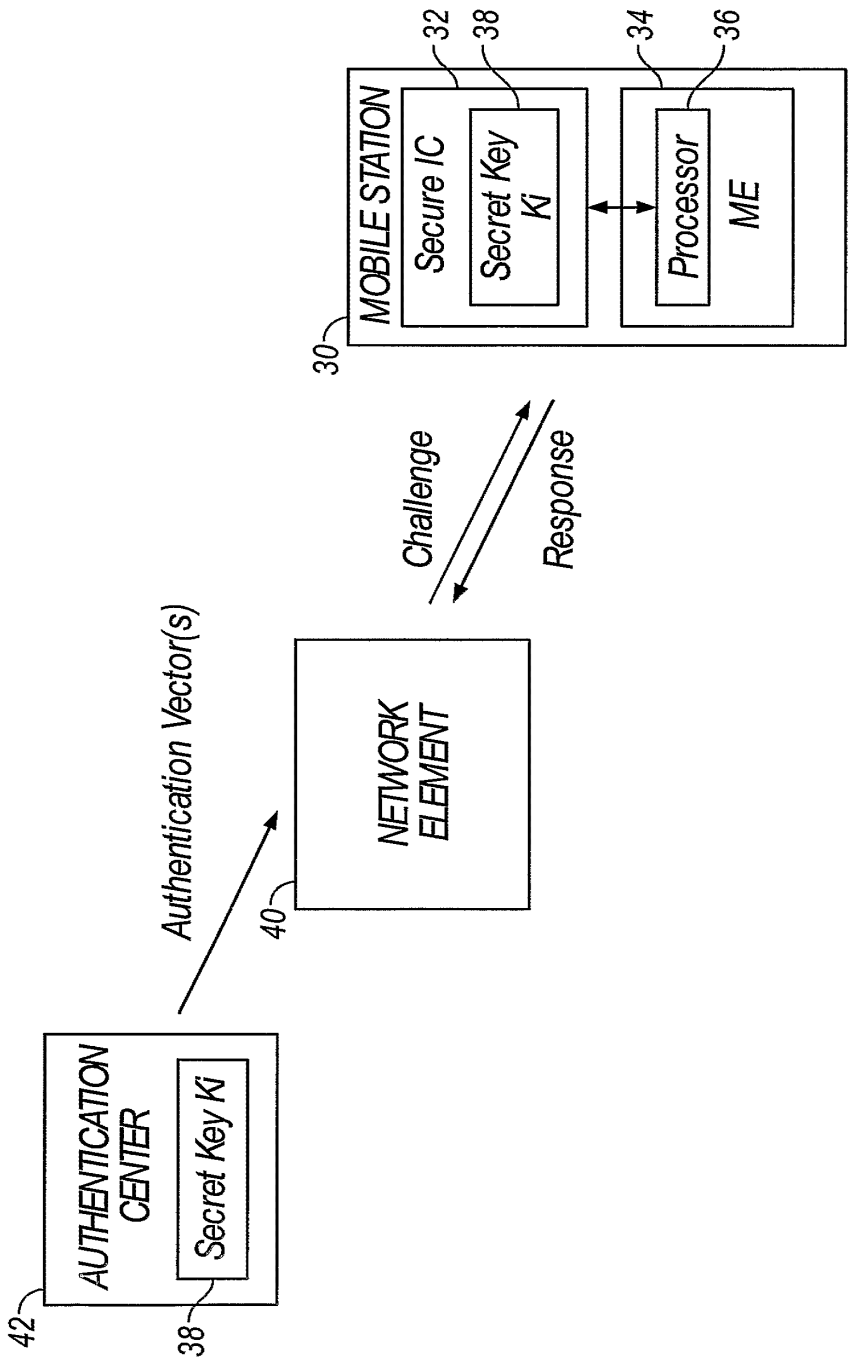

MOBILE STATION
Secure IC
Secret Key Ki
Processor
ME
32
38
34
36
30
Challenge
Response
Authentication Vector(s)
NETWORK ELEMENT
40
AUTHENTICATION CENTER
Secret Key Ki
42
38

FIG. 1

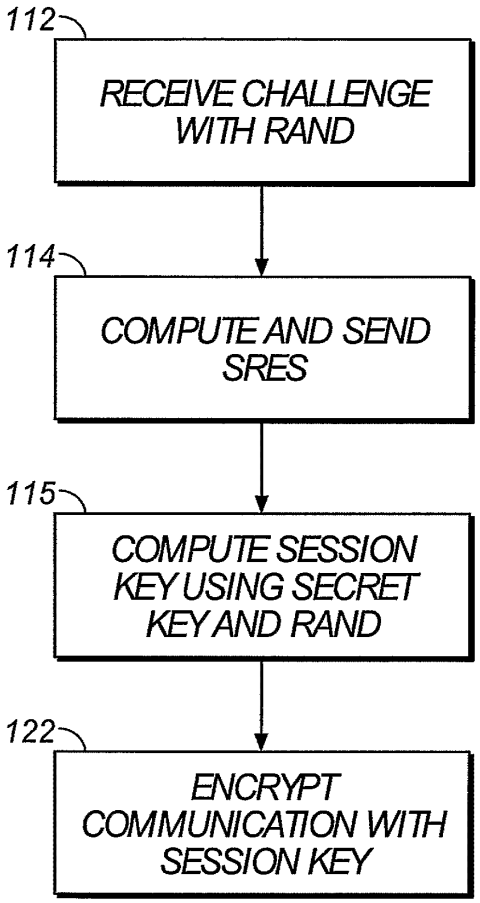

112
RECEIVE CHALLENGE WITH RAND
114
COMPUTE AND SEND SRES
115
COMPUTE SESSION KEY USING SECRET KEY AND RAND
122
ENCRYPT COMMUNICATION WITH SESSION KEY

FIG. 2

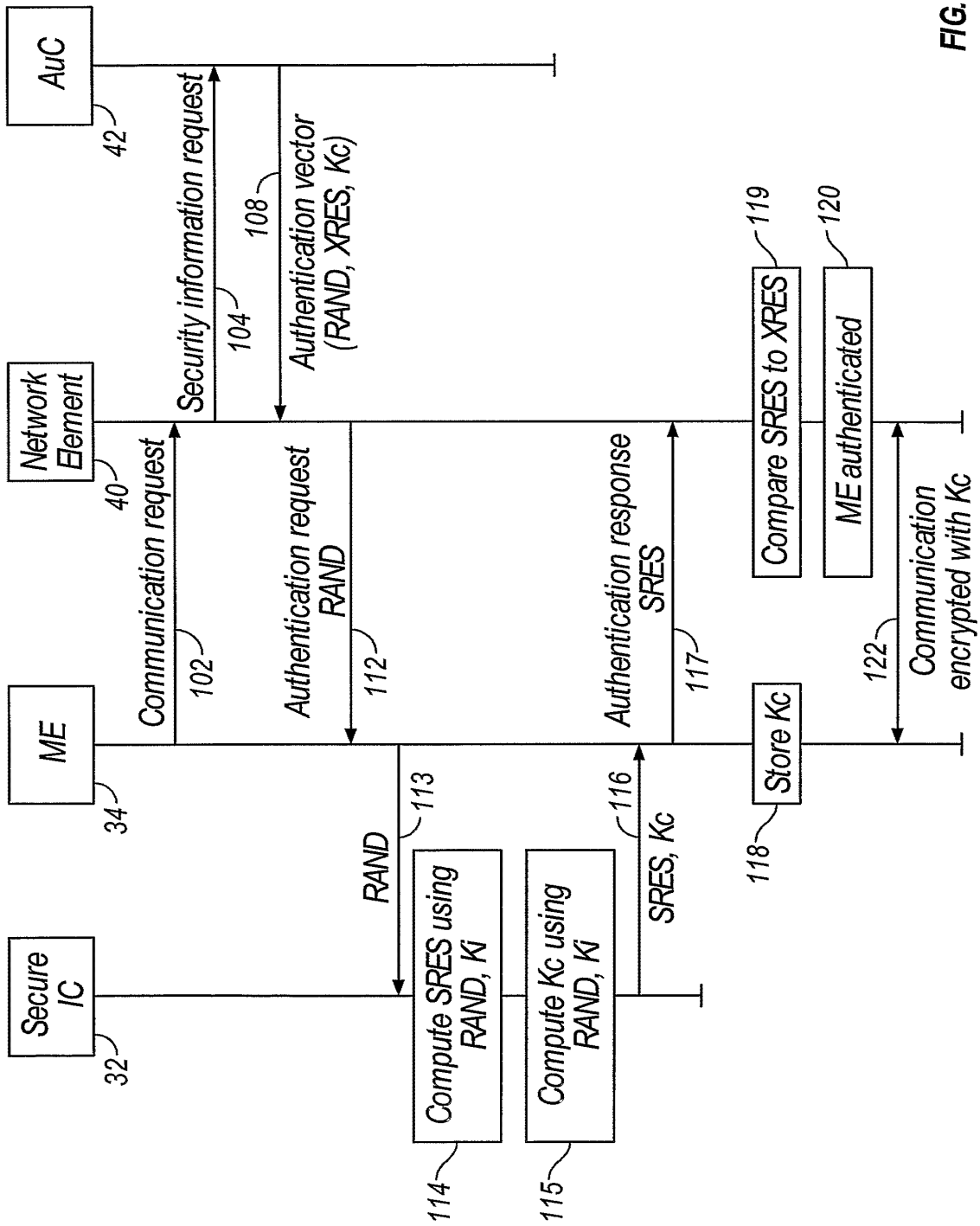

AuC
42
Network Element
40
ME
34
Secure IC
32
Security information request
104
Authentication vector (RAND, XRES, Kc)
108
Communication request
102
Authentication request RAND
112
RAND
113
Compute SRES using RAND, Ki
114
Compute Kc using RAND, Ki
115
SRES, Kc
116
Authentication response SRES
117
Store Kc
118
Compare SRES to XRES
119
ME authenticated
120
Communication encrypted with Kc
122

FIG. 3

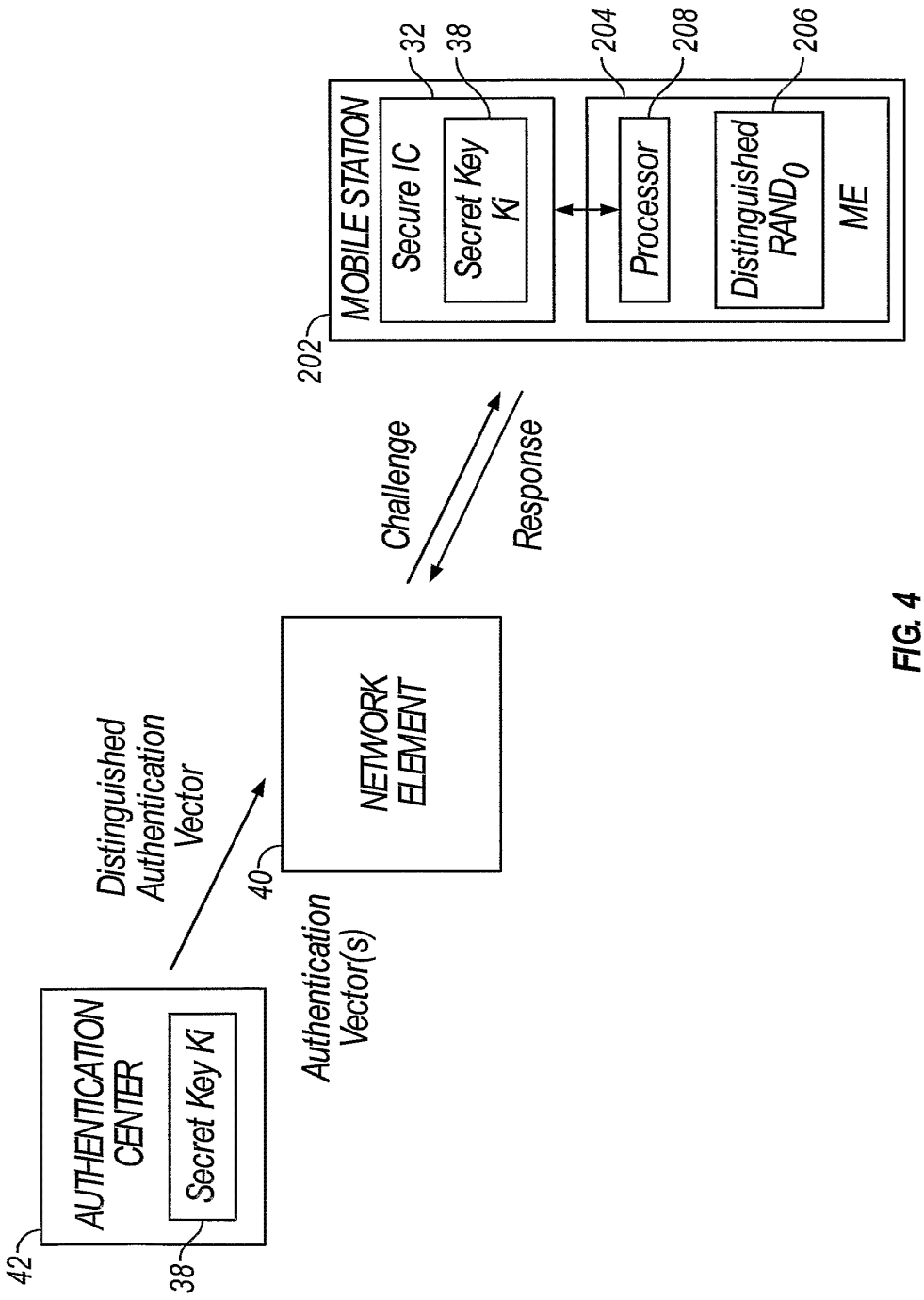

MOBILE STATION
Secure IC
Secret Key Ki
Processor
Distinguished RAND0
ME
32
38
204
208
206
202
Challenge
Response
NETWORK ELEMENT
40
Distinguished Authentication Vector
Authentication Vector(s)
AUTHENTICATION CENTER
Secret Key Ki
42
38

FIG. 4

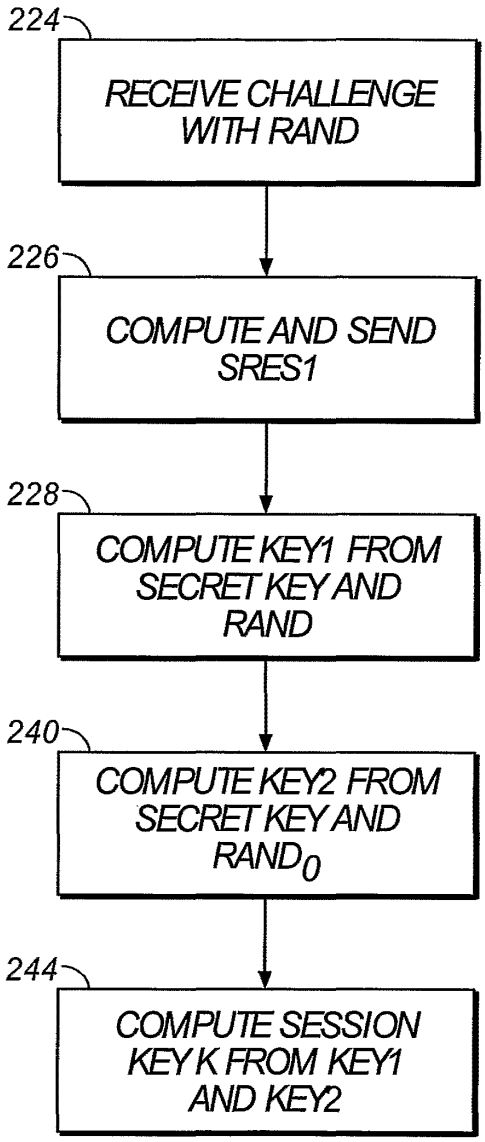

224
RECEIVE CHALLENGE WITH RAND
226
COMPUTE AND SEND SRES1
228
COMPUTE KEY1 FROM SECRET KEY AND RAND
240
COMPUTE KEY2 FROM SECRET KEY AND RAND0
244
COMPUTE SESSION KEY K FROM KEY1 AND KEY2

FIG. 5

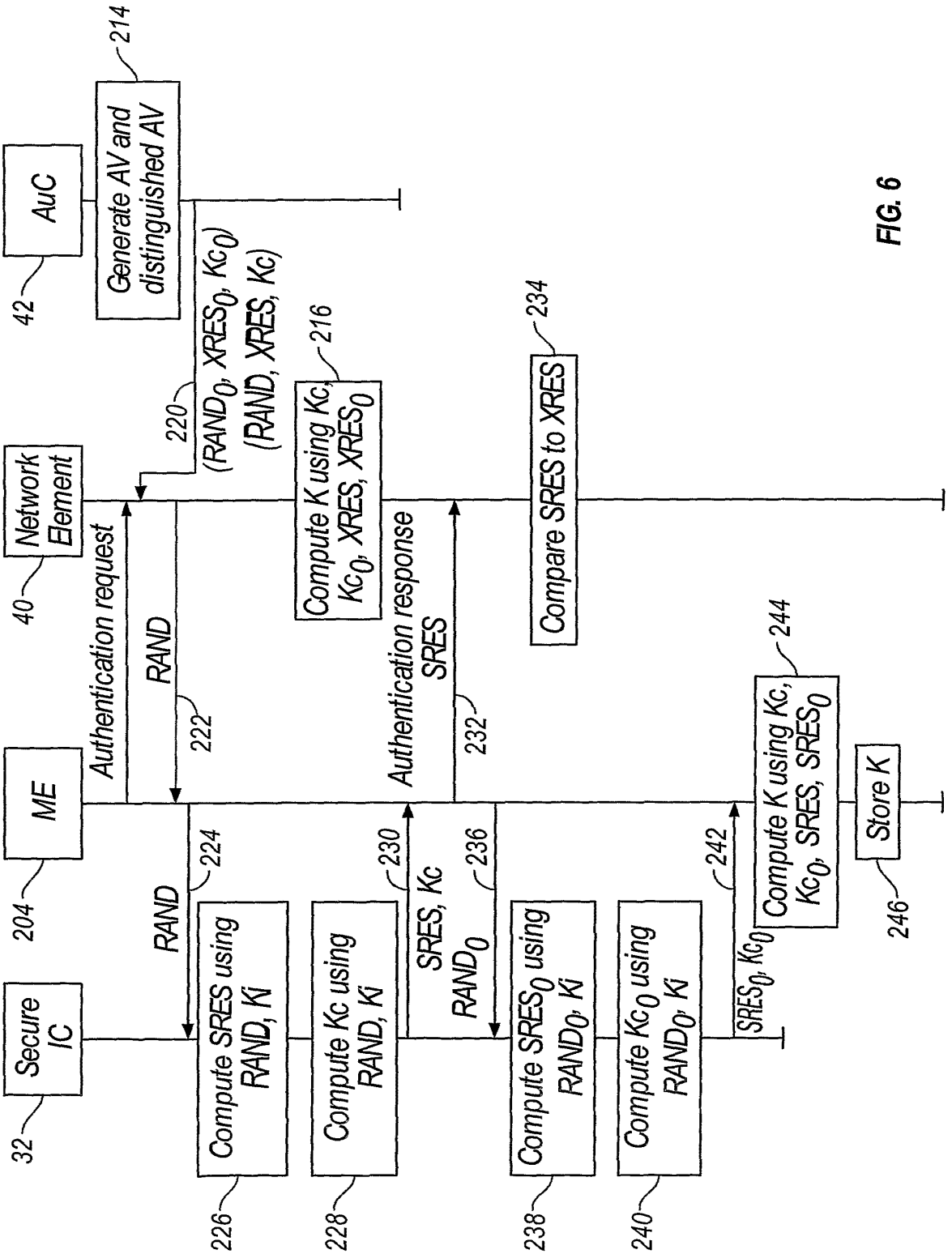

AuC
42
Generate AV and distinguished AV
214
Network Element
40
220
(RAND0, XRES0, Kc0)
(RAND, XRES, Kc)
Compute K using Kc, Kc0, XRES, XRES0
216
Compare SRES to XRES
234
ME
204
Authentication request
RAND
222
Authentication response
SRES
232
Secure IC
32
RAND
224
Compute SRES using RAND, Ki
226
Compute Kc using RAND, Ki
228
230
SRES, Kc
RAND0
236
Compute SRES0 using RAND0, Ki
238
Compute Kc0 using RAND0, Ki
240
242
SRES0, Kc0
Compute K using Kc, Kc0, SRES, SRES0
244
Store K
246

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No
/US2005/032337

A. CLASSIFICATION OF SUBJECT MATTER
H04Q7/32

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04Q H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	DE 101 28 300 A1 (GIESECKE & DEVRIENT GMBH) 9 January 2003 (2003-01-09) paragraph '0022! - paragraph '0032! paragraph '0040! paragraph '0042! paragraph '0048! - paragraph '0051! paragraph '0054! figure 3	1-8, 15-20, 22-28
A	US 6 711 400 B1 (AURA TUOMAS) 23 March 2004 (2004-03-23) column 5, line 21 - line 51 column 6, line 16 - column 7, line 46 figure 4	1-28

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
9 February 2006	16/02/2006

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Müller, N

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

tional application No
/US2005/032337

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 00/14895 A (DETEMOBIL DEUTSCHE TELEKOM MOBILNET GMBH; HAKE, JENS; THELEN, JOERG) 16 March 2000 (2000-03-16) page 2, line 6 - page 3, line 26 page 4, line 9 - line 29	1-28
A	US 5 661 806 A (NEVOUX ET AL) 26 August 1997 (1997-08-26) column 1, line 58 - column 2, line 50 column 4, line 40 - line 65 figure 2	1-28
P,X	WO 2005/048638 A (QUALCOMM INCORPORATED; ROSE, GREGORY G; PADDON, MICHAEL; HAWKES, PHILI) 26 May 2005 (2005-05-26) paragraph '0035! paragraph '0039! figures 5a,6	15-18, 22-28

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2005/032337

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
DE 10128300 A1	09-01-2003	WO 02102103 A2	19-12-2002
		EP 1400142 A2	24-03-2004
US 6711400 B1	23-03-2004	AU 6733198 A	24-11-1998
		EP 0976219 A2	02-02-2000
		FI 971620 A	17-10-1998
		WO 9849855 A2	05-11-1998
WO 0014895 A	16-03-2000	AU 1259200 A	27-03-2000
		CA 2343180 A1	16-03-2000
		DE 19840742 A1	09-03-2000
		EP 1112666 A2	04-07-2001
		PL 347024 A1	11-03-2002
		US 6934531 B1	23-08-2005
US 5661806 A	26-08-1997	EP 0675615 A1	04-10-1995
		FR 2718312 A1	06-10-1995
		JP 8008899 A	12-01-1996
WO 2005048638 A	26-05-2005	US 2005100165 A1	12-05-2005

WAVELYNX EX. 1003, Pg. 624
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2006 年11 月30 日 (30.11.2006)

PCT

(10) 国際公開番号
WO 2006/126668 A1

(51) 国際特許分類:
G09C 1/00 (2006.01) *H04L 9/32* (2006.01)

(21) 国際出願番号: PCT/JP2006/310534

(22) 国際出願日: 2006 年5 月26 日 (26.05.2006)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2005-155496 2005 年5 月27 日 (27.05.2005) JP

(71) 出願人 (米国を除く全ての指定国について): 日本電気株式会社 (NEC CORPORATION) [JP/JP]; 〒1080014 東京都港区芝五丁目７番１号 Tokyo (JP).

(72) 発明者; および

(75) 発明者/出願人 (米国についてのみ): 寺西 勇 (TERANISHI, Isamu) [JP/JP]; 〒1080014 東京都港区芝五丁目７番１号 日本電気株式会社内 Tokyo (JP).

(74) 代理人: 山下 穣平 (YAMASHITA, Johei); 〒1050001 東京都港区虎ノ門五丁目１３番１号虎ノ門４０ＭＴビル 山下国際特許事務所 Tokyo (JP).

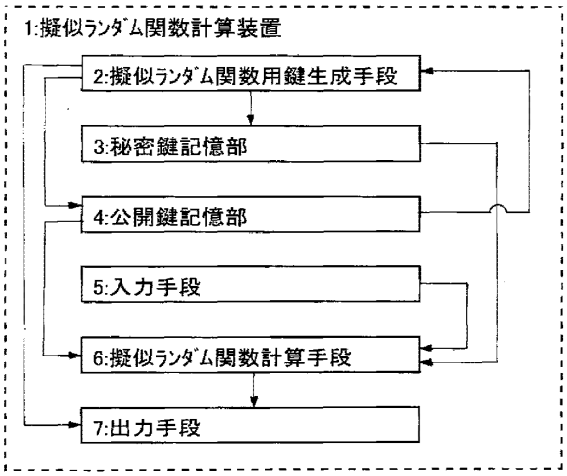
(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

[続葉有]

WO 2006/126668 A1

(54) **Title:** PSEUDO-RANDOM FUNCTION CALCULATING DEVICE AND METHOD, AND NUMBER LIMITED ANONYMOUS AUTHENTICATION SYSTEM AND METHOD

(54) 発明の名称: 擬似ランダム関数計算装置及び方法、並びに回数制限匿名認証システム及び方法



1 PSEUDO-RANDOM FUNCTION CALCULATING DEVICE
2 PSEUDO-RANDOM FUNCTION KEY CREATING MEANS
3 SECRET KEY STORAGE UNIT
4 PUBLIC KEY STORAGE UNIT
5 INPUT MEANS
6 PSEUDO-RANDOM FUNCTION CALCULATING MEANS
7 OUTPUT MEANS

(57) **Abstract:** Realized are an efficient pseudo-random function and an efficient number-limited anonymous authentication system using that function. A pseudo-random function calculation device comprises key creating means and pseudo-random function calculating means. The key creating means creates a public key made of a group having first and second elements as components composing the elements of a finite group, and a secret key made of an integer, and the secret key is secretly stored in a secret key storage unit whereas the public key is laid open. The pseudo-random function calculating means outputs, when an integer is inputted, elements of finite groups as the function values of the pseudo-random function. At this time, the output is the product of a first element having a value obtained by calculating a power-residue with the first element of the public key as a base and with the inputted integer as an exponent, and a second element obtained by calculating a power-residue with a second element of the public key as a base and with a reciprocal of a finite of a sum of the secret key and the inputted integer as an exponent.

(57) 要約: 効率的な擬似ランダム関数及びそれを用いた効率的な回数制限匿名認証方式を実現する。擬似ランダム関数計算装置は、鍵生成手段と、擬似ランダム関数計算手段とを有する。鍵生成手段は、有限群の元を成す要素として第一及び第二の要素を持つ組からなる公開鍵と、整数からなる秘密鍵とを生成し、秘密鍵を秘密鍵記憶部に秘密裡に保管し、公開鍵を公開する。擬似ランダム関数計算手段は、整数が入力されると、擬似ランダム関数の関数値として有限群の元を出力する。このとき、公開鍵の前記第一の要素を底とし、入

[続葉有]

WAVELYNX EX. 1003, Pg. 625
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

力された整数を指数として、べき乗剰余を計算して得られる値からなる第一の元と、公開鍵の第二の要素を底とし、秘密鍵と入力された整数との和の有限体での逆数を指数として、べき乗剰余を計算して得られる値からなる第二の元との積を出力する。

明 細 書

擬似ランダム関数計算装置及び方法、並びに回数制限匿名認証システム及び方法

技術分野

[0001] 本発明は、擬似ランダム関数計算装置及び方法、並びに回数制限匿名認証システム及び方法に係り、特に効率的な擬似ランダム関数及びそれを用いた効率的な回数制限匿名認証方式に関する。

背景技術

[0002] 一方向性関数を用いて擬似ランダム関数を実現する方法が知られている(例えば、非特許文献5参照)が、この方法で実現した擬似ランダム関数fは、y=f(x)を満たすxの知識のゼロ知識証明を効率的に行うことはできない。y=f(x)を満たすxの知識のゼロ知識証明を効率的に行うことができれば、様々な暗号プロトコルを効率的に実現できるので、擬似ランダム関数で効率的なものが望まれる。

[0003] 一方、電子投票、電子現金、電子クーポン、回数制限付き視聴といった多くのアプリケーションでは、ユーザのプライバシーを守るため、ユーザが匿名でこれらのアプリケーションを使える必要がある。しかも、これらのアプリケーションでは、これと同時に、ユーザがアプリケーションを使用できる回数を制限する必要がある。

[0004] 回数制限匿名認証方式(例えば、非特許文献4参照)は、このようなアプリケーションを実現するのに適した方式である。ユーザがアプリケーションを利用する度に、アプリケーション提供者(AP)がユーザをこの方式に従って認証することで、回数制限を守っている正直なユーザには匿名性を提供し、そうでないユーザは皆に特定されてしまうようにできる。

[0005] 特に非特許文献4では、タグ・メカニズムという、ユーザの認証回数を匿名のままカウントする仕組みを提案し、ACJTグループ署名方式(例えば、非特許文献1参照)のメンバー追加プロシージャとタグ・メカニズムとを組み合わせることで、回数制限匿名認証方式を実現している。

[0006] しかし、上記の回数制限匿名認証方式で用いられるタグ・メカニズムは、効率が悪く

WAVELYNX EX. 1003, Pg. 627
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

、認証時には、APとユーザは、ともに制限回数kに比例した回数の巾乗剰余を計算する必要がある。例えば、電子クーポンや回数制限つき視聴の場合には、制限回数が10を越えることも十分考えられる。このため、上記の非特許文献4の方式は、これらに応用するには効率が悪い。

非特許文献1:G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A Practical and Provably Secure Coalition-Resistant Group Signature Scheme", In Advances in Cryptology --- CRYPTO 2000, vol. 1880 of LNCS, pp. 255-270, Springer-Verlag, 2000

非特許文献2:P. S. L. M. Barreto, H. Y. Kim, B. Lynn, M. Scott, "Efficient Algorithms for Pairing-Based Cryptosystems", In Advances in Cryptology -- Crypto'2002, vol. 2442 of LNCS, pp. 354-368, Springer-Verlag, 2002

非特許文献3:Rafael Pass, "On Deniability in the Common Reference String and Random Oracle Model", In Advances in Cryptology --- CRYPTO 2003, vol. 2729 of LNCS, pp. 316-337, Springer-Verlag, 2003

非特許文献4:Isamu Teranishi, Jun Furukawa, and Kazue Sako, "k-Times Anonymous Authentication (Extended Abstract)", In Advances in Cryptology --- ASIACRYPT 2004, vol. 3329 of LNCS, pp. 308-322, Springer-Verlag, 2004

非特許文献5:Oded Goldreich, "Foundation of Cryptography, Basic Tools", Cambridge University Press, ISBN 0-521-79172-3, USA, 2001, pp. 148-169

発明の開示

発明が解決しようとする課題

[0007] 既存の擬似ランダム関数の問題点は、擬似ランダム関数fは、y=f(x)を満たすxの知識のゼロ知識証明を効率的に行うことはできないことにある。その理由は、fの計算方法が複雑なためである。

[0008] 既存の回数制限匿名認証方式の問題点は、認証時におけるユーザの計算量が回数制限kに比例することである。

[0009] 本発明は、このような従来の事情を考慮してなされたもので、効率的な擬似ランダム関数及びそれを用いた効率的な回数制限匿名認証方式を実現することを目的とす

WAVELYNX EX. 1003, Pg. 628
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

る。

課題を解決するための手段

[0010] 上記目的を達成するため、本発明に係る擬似ランダム関数計算装置は、有限群の元を構成する要素として少なくとも第一の要素及び第二の要素を持つ組からなる公開鍵と、整数からなる秘密鍵とを生成し、生成された前記秘密鍵を記憶装置に秘密裡に保管し、生成された前記公開鍵を公開する鍵生成手段と、整数が入力されると、擬似ランダム関数の関数値として、前記有限群の元を出力する擬似ランダム関数計算手段とを有し、前記擬似ランダム関数計算手段は、前記有限群の元として、前記公開鍵の前記第一の要素を底とし、前記入力された整数を指数として、べき乗剰余を計算して得られる値からなる第一の元と、前記公開鍵の前記第二の要素を底とし、前記秘密鍵と前記入力された整数との和の有限体での逆数を指数として、べき乗剰余を計算して得られる値からなる第二の元と、の積を出力することを特徴とする。

[0011] 本発明の別の側面に係る擬似ランダム関数計算装置は、整数からなる秘密鍵を生成し、生成された前記秘密鍵を記憶装置に秘密裡に保管する鍵生成手段と、ビット列と整数との組が入力されると、擬似ランダム関数の関数値として、有限群の元を出力する擬似ランダム関数計算手段とを有し、前記擬似ランダム関数計算手段は、前記有限群の元として、前記入力された値で決まる値を底とし、前記入力された整数を指数として、べき乗剰余を計算して得られる値からなる第一の元と、前記入力された値で決まる値を底とし、前記秘密鍵と前記入力された整数の和の逆数を指数として、べき乗剰余を計算して得られる値を指数として、べき乗剰余演算により得られる値からなる第二の元と、の積を出力することを特徴とする。本発明において、前記底は、前記入力された値のハッシュ値であってもよい。

[0012] 本発明に係る回数制限匿名認証システムは、上記いずれかに記載の擬似ランダム関数計算装置を用いた回数制限匿名認証システムであって、識別子と、整数k、i、y、及びlと、有限群の元tとを入力するための入力手段と、前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、前記有限群に値をとる擬似ランダム関数の関数値を計算する第一タグ計算手段と、前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、前記有限群に値をとる擬似ラン

ダム関数の関数値を計算し、計算された擬似ランダム関数の関数値をl乗した値にtを乗じた値を計算する第二タグ計算手段と、前記第一タグ計算手段の計算結果と前記第二タグ計算手段の計算結果との組を出力する出力手段とを有するタグ生成手段を備えたことを特徴とする回数制限匿名認証システム。

[0013] 本発明において、整数kを入力するための入力手段と、電子署名方式の公開鍵及び秘密鍵ペアを選ぶ電子署名用鍵生成手段と、k個の整数を選ぶ平文選択手段と、前記k個の整数各々に対する署名文を前記公開鍵及び秘密鍵ペアを用いて計算する電子署名計算手段と、前記電子署名方式の公開鍵と前記k個の整数と前記k個の署名文とからなる組を、前記タグ生成手段の計算に使用されるタグ用公開鍵として出力する出力手段と、を有するタグ用鍵生成手段を更に備えてもよい。

[0014] 前記電子署名計算手段は、平文として整数を入力するための手段と、平文と整数の和の有限体での逆元を計算する手段と、計算された前記逆元を指数としてべき乗剰余を計算し、前記べき乗剰余の計算結果を含む組を前記タグ用公開鍵として出力する手段とを有してもよい。

[0015] 前記電子署名用鍵生成手段は、有限群から元を選ぶ手段と、整数を選ぶ手段と、前記元を底とし、前記整数を指数として、べき乗剰余を計算する手段と、選ばれた前記有限群の元と前記べき乗剰余の計算結果とからなる組を出力する手段とを有してもよい。

[0016] 本発明において、前記タグ生成手段に整数lを入力して計算される計算結果をτとし、前記タグ生成手段に整数l'を入力して計算される計算結果をτ'としたとき、前記τ、l、τ'、l'の四つのデータを入力するための入力手段と、前記τを前記τ'で割った値を底とし、前記lから前記l'を減じた値の有限体での逆数を指数として、べき乗剰余を計算する計算手段と、前記べき乗剰余の計算結果を出力する出力手段と、を有するメンバー特定情報抽出手段を更に備えてもよい。

[0017] 本発明において、グループメンバーとしての公開鍵及び秘密鍵ペアと、アプリケーション提供者(以下、AP)装置の公開鍵と、前記AP装置の識別子と、整数k、i、及びlとを入力するための入力手段と、前記グループメンバーとしての秘密鍵から整数yを作り、前記AP装置の識別子と、前記k、i、l、及びyとを入力として、前記タグ生成手

WAVELYNX EX. 1003, Pg. 630
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

段によりタグを成すデータを計算する手段と、前記タグの正当性証明文を計算する正当性証明手段と、前記タグと前記正当性証明文とを出力する出力手段と、を有するグループ証明手段を更に備えてもよい。

[0018] 本発明において、有限群の元τと有限群の元μと整数lと証明文pとを要素とする第一の組と、有限群の元τ'と有限群の元μ'と整数l'と証明文p'とを要素とする第二の組とを入力するための入力手段と、前記τと前記τ'とが同一であるか否かを判定する第一判定手段と、前記lと前記l'とが同一であるか否かを判定する第二判定手段と、前記証明文pが正当であるか否かを判定する第三判定手段と、前記証明文p'が正当であるか否かを判定する第四判定手段と、予め設定された対応表に基づいて、前記メンバー特定情報抽出手段の計算結果と対応する識別子を取得する識別子取得手段と、を有するトレース手段を更に有してもよい。

[0019] 本発明に係る擬似ランダム関数計算方法は、有限群の元を構成する要素として少なくとも第一の要素及び第二の要素を持つ組からなる公開鍵と、整数からなる秘密鍵とを生成し、生成された前記秘密鍵を記憶装置に秘密裡に保管し、生成された前記公開鍵を公開する鍵生成ステップと、整数が入力されると、擬似ランダム関数の関数値として、前記有限群の元を出力する擬似ランダム関数計算ステップとを有し、前記擬似ランダム関数計算ステップは、前記有限群の元として、前記公開鍵の前記第一の要素を底とし、前記入力された整数を指数として、べき乗剰余を計算して得られる値からなる第一の元と、前記公開鍵の前記第二の要素を底とし、前記秘密鍵と前記入力された整数との和の有限体での逆数を指数として、べき乗剰余を計算して得られる値からなる第二の元と、の積を出力することを特徴とする。

[0020] 本発明の別の側面に係る擬似ランダム関数計算方法は、整数からなる秘密鍵を生成し、生成された前記秘密鍵を記憶装置に秘密裡に保管する鍵生成ステップと、ビット列と整数との組が入力されると、擬似ランダム関数の関数値として、有限群の元を出力する擬似ランダム関数計算ステップとを有し、前記擬似ランダム関数計算ステップは、前記有限群の元として、前記入力された値で決まる値を底とし、前記入力された整数を指数として、べき乗剰余を計算して得られる値からなる第一の元と、前記入力された値で決まる値を底とし、前記秘密鍵と前記入力された整数の和の逆数を指

WAVELYNX EX. 1003, Pg. 631
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

数として、べき乗剰余を計算して得られる値を指数として、べき乗剰余演算により得られる値からなる第二の元と、の積を出力することを特徴とする。本発明において、前記底は、前記入力された値のハッシュ値であってもよい。

[0021] 本発明に係る回数制限匿名認証方法は、上記いずれかに記載の擬似ランダム関数計算方法を用いた回数制限匿名認証方法であって、識別子と、整数k、i、y、及びlと、有限群の元tとを入力するためのステップと、前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、前記有限群に値をとる擬似ランダム関数の関数値を計算する第一タグ計算ステップと、前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、前記有限群に値をとる擬似ランダム関数の関数値を計算し、計算された擬似ランダム関数の関数値をl乗した値にtを乗じた値を計算する第二タグ計算ステップと、前記第一タグ計算ステップの計算結果と前記第二タグ計算ステップの計算結果との組を出力するステップとを有するタグ生成ステップを備えたことを特徴とする。

[0022] 本発明において、整数kを入力するための入力ステップと、電子署名方式の公開鍵及び秘密鍵ペアを選ぶ電子署名用鍵生成ステップと、k個の整数を選ぶ平文選択ステップと、前記k個の整数各々に対する署名文を前記公開鍵及び秘密鍵ペアを用いて計算する電子署名計算ステップと、前記電子署名方式の公開鍵と前記k個の整数と前記k個の署名文とからなる組を、前記タグ生成手段の計算に使用されるタグ用公開鍵として出力する出力ステップと、を有するタグ用鍵生成ステップを更に備えてもよい。

[0023] 前記電子署名計算ステップは、平文として整数を入力するためのステップと、平文と整数の和の有限体での逆元を計算するステップと、計算された前記逆元を指数としてべき乗剰余を計算し、前記べき乗剰余の計算結果を含む組を前記タグ用公開鍵として出力するステップとを有してもよい。

[0024] 前記電子署名用鍵生成ステップは、有限群から元を選ぶステップと、整数を選ぶステップと、前記元を底とし、前記整数を指数として、べき乗剰余を計算するステップと、選ばれた前記有限群の元と前記べき乗剰余の計算結果とからなる組を出力するステップとを有してもよい。

WAVELYNX EX. 1003, Pg. 632
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0025] 本発明において、前記タグ生成ステップにて整数lを入力して計算される計算結果をτとし、前記タグ生成ステップにて整数l'を入力して計算される計算結果をτ'としたとき、前記τ、l、τ'、l'の四つのデータを入力するためのステップと、前記τを前記τ'で割った値を底とし、前記lから前記l'を減じた値の有限体での逆数を指数として、べき乗剰余を計算するステップと、前記べき乗剰余の計算結果を出力するステップと、を有するメンバー特定情報抽出ステップを更に備えてもよい。

[0026] 本発明において、グループメンバーとしての公開鍵及び秘密鍵ペアと、アプリケーション提供者(以下、AP)装置の公開鍵と、前記AP装置の識別子と、整数k、i、及びlとを入力するためのステップと、前記グループメンバーとしての秘密鍵から整数yを作り、前記AP装置の識別子と、前記k、i、l、及びyとを入力として、前記タグ生成ステップにてタグを成すデータを計算するステップと、前記タグの正当性証明文を計算するステップと、前記タグと前記正当性証明文とを出力するステップと、を有するグループ証明ステップを更に備えてもよい。

[0027] 本発明において、有限群の元τと有限群の元μと整数lと証明文pとを要素とする第一の組と、有限群の元τ'と有限群の元μ'と整数l'と証明文p'とを要素とする第二の組とを入力するためのステップと、前記τと前記τ'とが同一であるか否かを判定するステップと、前記lと前記l'とが同一であるか否かを判定するステップと、前記証明文pが正当であるか否かを判定するステップと、前記証明文p'が正当であるか否かを判定するステップと、予め設定された対応表に基づいて、前記メンバー特定情報抽出ステップの計算結果と対応する識別子を取得するステップと、を有するトレースステップを更に備えてもよい。

発明の効果

[0028] 本発明によれば、効率的な擬似ランダム関数及びそれを用いた効率的な回数制限匿名認証方式を実現できる。

[0029] 即ち、擬似ランダム関数は、少ない回数の逆数計算とべき乗剰余とで関数値を計算できる。関数値の計算アルゴリズムが単純単純になり、これにより、y=f(x)を満たすxの知識のゼロ知識証明を効率的に行うことができ、既存の擬似ランダム関数の問題点を解決できる。

WAVELYNX EX. 1003, Pg. 633
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0030] また、回数制限匿名認証では、既存の方式とは異なり、ユーザが計算すべきデータ個数がO(log k)個である。よって、認証時におけるユーザの計算量は制限回数kに比例しない。

図面の簡単な説明

[0031] [図1]本発明の実施例5、6による回数制限匿名認証システムの全体構成を示す図である。

[図2]本発明の実施例1による擬似ランダム関数計算装置の全体構成を示す図である。

[図3]図2に示す擬似ランダム関数用鍵生成手段と擬似ランダム関数計算手段の処理手順を説明する概略フローチャートである(実施例1)。

[図4]本発明の実施例2による擬似ランダム関数計算装置の全体構成を示す図である。

[図5]図4に示す擬似ランダム関数用鍵生成手段と擬似ランダム関数計算手段の処理手順を説明する概略フローチャートである(実施例2)。

[図6]本発明の実施例3による回数制限匿名認証システムの全体構成を示す図である。

[図7]図6に示す回数制限匿名認証システムの各手段の処理手順を説明する概略フローチャートである(実施例3)。

[図8]図7に示すタグ用鍵生成手段で用いる電子署名用鍵生成手段と電子署名手段の処理手順を説明する概略フローチャートである(実施例3)。

[図9]本発明の実施例4による回数制限匿名認証システムの全体構成を示す図である。

[図10]図9に示すタグ計算手段の処理手順を説明する概略フローチャートである(実施例4)。

[図11]図1に示すGMセットアップ手段とAPセットアップ手段の処理手順を説明する概略フローチャートである(実施例5)。

[図12]図1に示す発行手段と参加手段の処理手順を説明する概略フローチャートである(実施例5)。

WAVELYNX EX. 1003, Pg. 634
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図13]図1に示すグループ証明手段とグループ検証手段の処理手順を説明する概略フローチャートである(実施例5)。

[図14]図1に示すトレース手段の処理手順を説明する概略フローチャートである(実施例5)。

[図15]図1に示すトレース手段とリスト記憶部の処理手順を説明する概略フローチャートである(実施例5)。

[図16]図1に示すGMセットアップ手段で用いる発行者用鍵生成手段の処理手順を説明する概略フローチャートである(実施例5)。

[図17]図1に示す発行手段及び参加手段で用いる第一発行手段及び第一参加手段の処理手順を説明する概略フローチャートである(実施例5)。

[図18]図1に示す発行手段及び参加手段で用いる第二発行手段、第二参加手段、及びファイ計算手段の処理手順を説明する概略フローチャートである(実施例5)。

[図19]図1に示すユーザ装置による証明文の作成方法の処理手順を説明する概略フローチャートである(実施例5)。

[図20]図1に示すユーザ装置による証明文の作成方法の処理手順を説明する概略フローチャートである(実施例5)。

[図21]図1に示すAP装置による証明文の正当性検証方法の処理手順を説明する概略フローチャートである(実施例5)。

[図22]図1に示すユーザ装置による証明文の作成方法の処理手順を説明する概略フローチャートである(実施例6)。

[図23]図1に示すユーザ装置による証明文の作成方法の処理手順を説明する概略フローチャートである(実施例6)。

[図24]図1に示すAP装置による証明文の正当性検証方法の処理手順を説明する概略フローチャートである(実施例6)。

符号の説明

[0032] 1 擬似ランダム関数計算装置

2 擬似ランダム関数用鍵生成手段

3 秘密鍵記憶部

WAVELYNX EX. 1003, Pg. 635
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

4 公開鍵記憶部

5 入力手段

6 擬似ランダム関数計算手段

7 出力手段

10 メンバー特定情報生成装置

11 秘密鍵生成手段

12 公開情報記憶部

13 メンバー特定情報生成手段

14 通信装置

15 書き込み手段

20 乱数生成装置

21 公開情報記憶部

22 乱数選択手段

23 通信手段

30 タグ生成装置

31 公開情報記憶部

32 タグ生成手段

33 入力手段

34 通信装置

35 タグ計算手段

40 メンバー特定情報抽出装置

41 公開情報記憶部

42 メンバー特定情報抽出手段

43 一致性判定手段

44 出力手段

45 通信装置

50 鍵生成装置

51 入力手段

WAVELYNX EX. 1003, Pg. 636
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

52 公開情報記憶部

53 タグ用鍵生成手段

54 通信手段

100 GM装置

101 GMセットアップ手段

102 発行手段

103 秘密情報記憶部

104 公開情報記憶部

105 通信手段

200 リスト記憶装置

201 リスト記憶部

202 通信手段

300 ユーザ装置

301 参加手段

302 グループ証明手段

303 秘密情報記憶部

304 公開情報記憶部

305 通信手段

400 AP装置

401 APセットアップ手段

402 グループ検証手段

403 公開情報記憶部

404 履歴記憶部

405 通信手段

500 トレース装置

501 トレース手段

502 公開情報記憶部

503 通信手段

発明を実施するための最良の形態

[0033] 以下、本発明に係る擬似ランダム関数計算装置及び方法、並びに回数制限匿名認証システム及び方法を実施するための最良の形態について、添付図面を参照して説明する。

実施例 1

[0034] 本実施例は、擬似ランダム関数計算装置に適用したものである。

[0035] 図2を参照して、本実施例の装置構成を説明する。図2に示す擬似ランダム関数計算装置1は、機能上、擬似ランダム関数用鍵生成手段2、秘密鍵記憶部3、公開鍵記憶部4、入力手段5、擬似ランダム関数計算手段6、及び出力手段7を有している。

[0036] 擬似ランダム関数計算装置1は、一例としてコンピュータ機のCPU、記憶装置、その他の各種入出力装置によって実現される。この例では、擬似ランダム関数用鍵生成手段2及び擬似ランダム関数計算手段6は、コンピュータ機のCPUが記憶装置上のコンピュータプログラムの命令を実行することにより実現される。ここで用いられるコンピュータプログラムは、各手段2、6の処理アルゴリズム(後述参照)に応じて予め設定される。また、秘密鍵記憶部3及び公開鍵記憶部4は、コンピュータ機の記憶装置により実装される。さらに、入力手段5及び出力手段7は、コンピュータ機の入出力装置に対応する。

[0037] 以下の説明では、ωをセキュリティ・パラメータとし、G__1を有限巡回群でG__1の位数がωビットの整数であるものとし、qをG__1の位数とする。本実施例の擬似ランダム関数計算装置1は、なんらかの方法でG__1のパラメータ、ω、およびqを事前に入手しており、G__1のパラメータ、ω、およびqが公開鍵記憶部4に書き込まれているものとする。G__1のパラメータ、ω、およびqを入手する方法は、どのようなものでもよい。例えば、外部からの入力として得る、装置の回路にハードウェア的に書き込んでおく、といった方法がある。

[0038] 次に、擬似ランダム関数用鍵生成手段2及び擬似ランダム関数計算手段6の処理手順を図3に従って説明する。図3に示す処理手順は、コンピュータ機の記憶装置上に格納されるコンピュータプログラムとして実現され、コンピュータ機のCPUにより実行される。

WAVELYNX EX. 1003, Pg. 638
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0039] 図3において、擬似ランダム関数用鍵生成手段2は、次の処理を実行する。まず、G__1の元g、hをランダムに選ぶ(ステップS1)。次に、Z__qの元yをランダムに選ぶ(ステップS2)。次に、yを秘密鍵記憶部3に、(g, h)を公開鍵記憶部4にそれぞれ書き込む(ステップS3)。最後に、(g, h)を出力する(ステップS4)。

[0040] 図3において、擬似ランダム関数計算手段6は、次の処理を実行する。まず、G__1の元iを入力として受け取る(ステップS5)。次に、f(i)＝g^{i}h^{1／(y＋i)}を計算する(ステップS6)。最後に、f(i)を出力する(ステップS7)。

[0041] 従って、本実施例によれば、少ない回数の逆数計算とべき乗剰余とで擬似ランダム関数の関数値を計算できるため、関数値の計算アルゴリズムが単純になり、これにより、y＝f(x)を満たすxの知識のゼロ知識証明を効率的に行うことができる。

実施例 2

[0042] 本実施例は、擬似ランダム関数計算装置に適用したものである。

[0043] 図4を参照して、本実施例の装置構成を説明する。図4に示す本実施例の擬似ランダム関数計算装置1は、機能上、擬似ランダム関数用鍵生成手段2、秘密鍵記憶部3、公開鍵記憶部4、入力手段5、擬似ランダム関数計算手段6、及び出力手段7を有している。

[0044] 擬似ランダム関数計算装置1は、一例としてコンピュータ機のCPU、記憶装置、その他の各種入出力装置によって実現される。この例では、擬似ランダム関数用鍵生成手段2及び擬似ランダム関数計算手段6は、コンピュータ機のCPUが記憶装置上のコンピュータプログラムの命令を実行することにより実現される。ここで用いられるコンピュータプログラムは、各手段2、6の処理アルゴリズム(後述参照)に応じて予め設定される。また、秘密鍵記憶部3及び公開鍵記憶部4は、コンピュータ機の記憶装置により実装される。さらに、入力手段5及び出力手段7は、コンピュータ機の入出力装置に対応する。

[0045] 以下の説明では、ωをセキュリティ・パラメータとし、G__1を有限巡回群でG__1の位数がωビットの整数であるものとし、qをG__1の位数とする。本実施例の擬似ランダム関数計算装置は、なんらかの方法でG__1のパラメータ、ω、およびqを事前に入手しており、G__1のパラメータ、ω、およびqが公開鍵記憶部4に書き込まれているも

のとする。G__1のパラメータ、ω、およびqを入手する方法はどのようなものでもよい。例えば、外部からの入力として得る、装置の回路にハードウェア的に書き込んでおく、といった方法がある。

[0046] 次に、擬似ランダム関数用鍵生成手段2の処理手順を図5に従って説明する。図5に示す処理手順は、コンピュータ機の記憶装置上に格納されるコンピュータプログラムとして実現され、コンピュータ機のCPUにより実行される。

[0047] 図5において、擬似ランダム関数用鍵生成手段2は、Z__qの元yをランダムに選び(ステップS11)、yを秘密鍵記憶部3に書き込む(ステップS12)。

[0048] 図5において、擬似ランダム関数計算手段6は、次の処理を実行する。

[0049] まず、ビット列XとG__1の元iを入力として受け取る(ステップS13)。

[0050] 次に、(g__{X}, h__{X})=H__{G__1^2}(X)を計算する(ステップS14)。ここで、H__{G__1^2}は、G__1^2に値を取るハッシュ関数を示す。

[0051] 次に、f[y]__{ω}(i, X)=g__{X}^{i}h__{X}^{1/(y+i)}を計算する(ステップS15)。

[0052] 最後に、f[y]__{ω}(i, X)を出力する(ステップS16)。

[0053] 従って、本実施例でも、前述の実施例1と同様に、少ない回数の逆数計算とべき乗剰余とで擬似ランダム関数の関数値を計算できるため、関数値の計算アルゴリズムが単純になり、これにより、y=f(x)を満たすxの知識のゼロ知識証明を効率的に行うことができる。

実施例 3

[0054] 本実施例は、前述した擬似ランダム関数計算装置を用いた回数制限匿名認証システムに適用したものである。

[0055] 図6を参照して、本実施例の装置構成を説明する。図6に示すシステムは、機能上、メンバー特定情報生成装置10、乱数生成装置20、タグ生成装置30、メンバー特定情報抽出装置40、及び鍵生成装置50を有している。前述した擬似ランダム関数計算装置は、タグ生成装置30により用いられている。

[0056] メンバー特定情報生成装置10は、機能上、秘密鍵生成手段11、公開情報記憶部12、メンバー特定情報生成手段13、書き込み手段14、及び通信手段15を備える。

[0057] 乱数生成装置20は、機能上、公開情報記憶部21、乱数選択手段22、及び通信手段23を備える。

[0058] タグ生成装置30は、前述した擬似ランダム関数計算装置を用いたもので、機能上、公開情報記憶部31、タグ計算手段32、入力手段33、及び通信装置34を備える。

[0059] メンバー特定情報抽出装置40は、機能上、公開情報記憶部41、メンバー特定情報抽出手段42、一致性判定手段43、出力手段44、及び通信装置45を備える。

[0060] 鍵生成装置50は、機能上、入力手段51、公開情報記憶部52、タグ用鍵生成手段53、及び通信手段54を備える。

[0061] 上記の各装置10～50は、一例としてコンピュータ機のCPU、記憶装置、ネットワークインターフェース部、その他の各種入出力装置によって実現される。この例では、各手段11、13、14、22、32、42、43、53は、各コンピュータ機のCPUが記憶装置上のコンピュータプログラムの命令を実行することにより実現される。ここで用いられるコンピュータプログラムは、各手段の処理アルゴリズム(後述参照)に応じて予め設定される。また、公開情報記憶部12、21、31、41、52は、各コンピュータ機の記憶装置により実装される。通信手段15、23、54、及び通信装置34, 45は、各コンピュータ機のネットワークインターフェース部に、また入力手段23、51、及び出力手段44は、各コンピュータ機の入出力装置にそれぞれ対応する。

[0062] 図6では、もっとも単純な場合として、5種類の装置10～50がそれぞれ一台ずつしかない場合を描いているが、各種類の装置が複数台あってもかまわない。また、図6では、各装置10～50が別々の機械である場合を描いているが、一つの機械が二種類の装置の機能を兼ねていてもかまわない。

[0063] メンバー特定情報生成装置10、乱数生成装置20、タグ生成装置30、メンバー特定情報抽出装置40、及び鍵生成装置50は、それぞれ通信手段14、23、34、45、54を使って相互に通信できる。通信媒体としては、どのような通信媒体を用いてもかまわない。例えばインターネット、電波、電話回線といったものを用いることができる。

[0064] 各装置10～50は、他の装置が公開している公開情報をなんらかの方法で入手できるものとする。公開情報の入手手段は、どのような方法を用いてもかまわない。例えば、公開情報を公開している装置から前述の通信手段を用いて直接受け取る方法、

WAVELYNX EX. 1003, Pg. 641
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

公開情報のリストを持っているサーバから前述の通信手段を用いて受け取る方法、を用いることができる。

[0065] 次に、本実施例の動作を図7及び図8に従って説明する。図7及び図8に示す処理手順は、各装置に対応するコンピュータ機の記憶装置上に格納されるコンピュータプログラムとして実現され、各コンピュータ機のCPUにより実行される。

[0066] 以下の説明では、ωをセキュリティ・パラメータとし、G__1を有限巡回群でG__1の位数がωビットの整数であるものとし、qをG__1の位数とする。本実施例で適用される擬似ランダム関数計算装置は、なんらかの方法でG__1のパラメータ、ω、およびqを事前に入手しており、G__1のパラメータ、ω、およびqが各装置10～50の公開情報記憶部12、21、31、41、52に書き込まれているものとする。G__1のパラメータ、ω、およびqを入手する方法はどのようなものでもよい。例えば外部からの入力として得る、装置の回路にハードウェア的に書き込んでおく、といった方法がある。各装置10～50は、必要に応じてこれらのデータを読み込む。

[0067] 図7において、まず、メンバー特定情報生成装置10は、秘密鍵生成手段11の処理を行って、Z__qの元yをランダムに選ぶ(ステップS21)。g__1をG__1の元とし、ν__{ω}(y)＝g__1^{y}とする。g__1は事前に何らかの方法で選ばれ、全ての装置10～50の公開情報記憶部12、21、31、41、52に事前に記憶されているものとする。g__1をどのような方法で選び、どのような方法で各装置10～50に配布してもかまわないが、安全性の観点から言えば、g__1を何らかの値のハッシュ値にセットすることが望ましい。

[0068] 次にメンバー特定情報生成装置10は、メンバー特定情報生成手段42の処理を行って、t__1＝ν__{ω}(y)とする(ステップS22)。

[0069] ここで、G__2、H__2、G__3を、位数がqである有限巡回群とし、＜・，・＞はG__2×H__2の元(g, h')にG__3の元＜g, h'＞を対応させる写像で、＜g^x, h'^y＞＝＜g, h'＞^{xy}が任意のg, h', x, yに対して成立するものとする。このような組(G__2、H__2、G__3、＜・，・＞)の生成方法、および＜・，・＞の計算方法は様々なものが知られているが、そのどれを用いてもよい(例えば、非特許文献2参照)。

[0070] 次に、鍵生成装置50がタグ用鍵生成手段53の処理を行う。

[0071] タグ用鍵生成手段53は、ステップS23、S24にて、それぞれ電子署名用鍵生成手段、電子署名手段による処理を行うので、タグ用鍵生成手段53の処理を説明する前に、電子署名用鍵生成手段と、電子署名手段との各処理を図8に従って説明する。

[0072] 図8において、電子署名用鍵生成手段は、まず、G__2の元g__2をランダムに選ぶ(ステップS41)。次に、H__2の元g'__2をランダムに選ぶ(ステップS42)。次に、Z__qの元sskをランダムに選ぶ(ステップS43)。

[0073] 次に、h'__2=g'__2^{ssk}を計算する(ステップS44)。

[0074] 最後に、(g__2, g'__2, h'__2)を電子署名用公開鍵spkとしてセットする(ステップS45)。

[0075] 電子署名手段は、まず、spkを(g__2, g'__2, h'__2)とパースする(ステップS46)。次に、平文βに対する署名文S=g__2^{1/(ssk+β)}を計算する(ステップS47)。

[0076] 鍵生成装置50は、非負整数kを入力として受け取ったら、図7に示すタグ用鍵生成手段53の処理を行う。

[0077] 図7において、まず、鍵生成装置50は、電子署名用鍵生成手段の処理を行い、電子署名用公開鍵spkと、電子署名用秘密鍵sskとを生成する(ステップS23)。

[0078] 次に、鍵生成装置50は、平文β__1, …, β__kを選び、(spk, ssk)を使って電子署名手段の処理を行い、各β__iに対する署名文S__iを作成する(ステップS24)。

[0079] 最後に鍵生成装置50は、apk=(spk, (β__1, S__1), …, (β__k, S__k))とする(ステップS25)。

[0080] 次に乱数生成装置20は、乱数選択手段21の処理を行い、Z__qの元lをランダムに選んで出力する(ステップS26)。

[0081] 次にタグ生成装置30は、通信手段23、34を使って乱数生成装置20と通信して、lを受信し、通信装置54、34を使って鍵生成装置50と通信して、apkを受信し、タグ生成手段32の処理を開始する。このタグ生成手段32の処理を説明する。

[0082] まず、タグ生成装置30は、タグ生成手段32の処理により、タグ生成装置30のID、タグ生成装置32がアクセスを許す回数の上限値k, および現時点でのアクセス回数i(i≦k)を受け取る(ステップS27)。

WAVELYNX EX. 1003, Pg. 643
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0083] 次にタグ生成装置30は、apkを(spk, (β__1, S__1), …, (β__k, S__k))とパースする(ステップS28)。

[0084] 次にタグ生成装置30は、f[y]__{ω}を実施例2の擬似ランダム関数とし、F[y]__{ω}(X, i)=f[y]__{ω}(X, −i)とする。

[0085] 最後に、タグ生成装置30は、(τ, μ)=(f[y]__{ω}(ID||k, β__i), F[y]__{ω}(ID||k, β__i))を計算する(ステップS29)。f[y]__{ω}とF[y]__{ω}は、前述した実施例2の擬似ランダム関数計算手段6の処理を行うことで計算する。

[0086] 次に、タグ生成手段32の処理が同じ入力(ID, k, i)を使って二度行われたとする。各タグ生成手段32の出力を(τ, μ, l)、(τ, μ', l')とする。

[0087] この場合、メンバー特定情報抽出装置40は、通信手段34、45を使って、タグ生成装置30から(μ, l)、(μ', l')を受け取り、メンバー特定情報抽出手段42と一致性判定手段43の各処理を順に行う。

[0088] メンバー特定情報抽出装置40は、メンバー特定情報抽出手段42の処理を行い、(μ/μ')^{1/(l−l')}を計算する(ステップS30、S31)。

[0089] メンバー特定情報抽出装置40は、一致性判定手段43の処理を行い、t__1と(μ/μ')^{1/(l−l')}とが入力されたら、t__1=(τ/τ')^{1/(l−l')}であるかどうかを出力する(ステップS32)。

[0090] 従って、本実施例によれば、回数制限匿名認証は既存の方式とは異なり、ユーザが計算すべきデータ個数がO(log k)個であるため、認証時におけるユーザの計算量は制限回数kに比例しないから、効率的な回数制限匿名認証方式を実現することができる。

実施例 4

[0091] 本実施例は、前述した擬似ランダム関数計算装置を用いた回数制限匿名認証システムに適用したものである。

[0092] 図9を参照して、本実施例の装置構成を説明する。図9に示すシステムは、機能上、メンバー特定情報生成装置10、乱数生成装置20、タグ生成装置30、及びメンバー特定情報抽出装置40を有している。前述した擬似ランダム関数計算装置は、タグ生成装置30により用いられている。

WAVELYNX EX. 1003, Pg. 644
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0093] メンバー特定情報生成装置10は、機能上、秘密鍵生成手段11、公開情報記憶部12、メンバー特定情報生成手段13、通信手段14、及び書き込み手段15を備える。

[0094] 乱数生成装置20は、機能上、公開情報記憶部21、乱数選択手段22、及び通信手段23を備える。

[0095] タグ生成装置30は、前述した擬似ランダム関数計算装置を用いたもので、機能上、公開情報記憶部31、入力手段33、通信装置34、及びタグ計算手段35を備える。

[0096] メンバー特定情報抽出装置40は、機能上、公開情報記憶部41、メンバー特定情報抽出手段42、一致性判定手段43、出力手段44、及び通信装置45を備える。

[0097] 上記の各装置10～50は、一例としてコンピュータ機のCPU、記憶装置、ネットワークインターフェース部、その他の各種入出力装置によって実現される。この例では、各手段11、13、14、22、42、43、53は、各コンピュータ機のCPUが記憶装置上のコンピュータプログラムの命令を実行することにより実現される。ここで用いられるコンピュータプログラムは、これら各手段の処理アルゴリズム(後述参照)に応じて予め設定される。また、公開情報記憶部12、21、31、41、52は、コンピュータ機の記憶装置により実装される。通信手段15、23、54、及び通信装置34, 45は、各コンピュータ機のネットワークインターフェース部に、また入力手段23、51、及び出力手段44は、各コンピュータ機の入出力装置にそれぞれ対応する。

[0098] 図9では、もっとも単純な場合として、5種類の装置がそれぞれ一台ずつしかない場合を描いているが、各種類の装置が複数台あってもかまわない。また、図では、各装置が別々の機械である場合を描いているが、一つの機械が二種類の装置の機能を兼ねていてもかまわない。

[0099] メンバー特定情報生成装置10、乱数生成装置20、タグ生成装置30、メンバー特定情報抽出装置40は、それぞれ通信手段14、23、34、45を使って相互に通信できる。通信媒体としてどのような通信媒体を用いてもかまわない。例えばインターネット、電波、電話回線といったものを用いることができる。

[0100] 各装置10～40は、他の装置が公開している公開情報をなんらかの方法で入手できるものとする。公開情報の入手手段は、どのような方法を用いてもかまわない。例えば、公開情報を公開している装置から前述の通信手段を用いて直接受け取る方法、

公開情報のリストを持っているサーバから前述の通信手段を用いて受け取る方法、を用いることができる。

[0101] 実施例4のメンバー特定情報生成装置10、乱数生成装置20、メンバー特定情報抽出装置40は、前述した実施例3のメンバー特定情報生成装置10、乱数生成装置20、及びメンバー特定情報抽出装置40と同じ動作をする。

[0102] また、実施例4のタグ生成装置30の通信装置34、公開情報記憶部31、入力手段33は、前述した実施例3のタグ生成装置30の通信装置34、公開情報記憶部31、及び入力手段33と同じ機能を持っている。

[0103] 次に、図10を参照して、実施例4のタグ計算手段35を説明する。図10に示す処理手順は、コンピュータ機の記憶装置上に格納されるコンピュータプログラムとして実現され、コンピュータ機のCPUにより実行される。

[0104] 以下の説明では、ωをセキュリティ・パラメータとし、G__1を有限巡回群でG__1の位数がωビットの整数であるものとし、qをG__1の位数とする。擬似ランダム関数計算装置は、なんらかの方法でG__1のパラメータ、ω、およびqを事前に入手しており、G__1のパラメータ、ω、およびqが各装置の公開情報記憶部に書き込まれているものとする。G__1のパラメータ、ω、およびqを入手する方法どのようなものでもよい。例えば外部からの入力として得る、装置の回路にハードウェア的に書き込んでおく、といった方法がある。各装置10～50は、必要に応じてこれらのデータを読み込む。

[0105] 図10において、タグ生成装置3は、タグ計算手段35の処理を行い、タグ生成装置のID、タグ生成装置がアクセスを許す回数の上限値k、および現時点でのアクセス回数i≦kを受け取る(ステップS51)。

[0106] f[y]__{ω}を前述した実施例2の擬似ランダム関数とし、F[y]__{ω}(X, i)＝f[y]__{ω}(X, －i)とする。

[0107] 最後に、タグ生成装置3は、(τ, μ)＝(f[y]__{ω}(ID｜｜k, i), F[y]__{ω}(ID｜｜k, i))を計算する(ステップS52)。

[0108] f[y]__{ω}とF[y]__{ω}は、実施例2の擬似ランダム関数計算手順BPRF5を行うことで計算する。

[0109] 従って、本実施例でも、前述の実施例3と同様に、回数制限匿名認証は既存の方

式とは異なり、ユーザが計算すべきデータ個数がO(log k)個であるため、認証時におけるユーザの計算量は制限回数kに比例しないから、効率的な回数制限匿名認証方式を実現することができる。

実施例 5

[0110] 本実施例は、前述した擬似ランダム関数計算装置を用いた回数制限匿名認証システムに適用したものである。

[0111] 図1を参照して、本実施例の装置構成を説明する。

[0112] 図1に示す本実施例の回数制限匿名認証システムは、前述した実施例3、4のシステムに様々な計算手順を追加して構成されており、機能上、GM(Group Manager)装置(グループ管理装置)100、リスト記憶装置200、ユーザ装置300、AP(Application Provider)装置(アプリケーション提供装置)400、及びトレース装置500の5種類の装置を備える。前述した擬似ランダム関数計算装置は、後述するグループ署名手段(グループ証明手段、グループ検証手段)に適用されている。

[0113] GM装置100は、機能上、GMセットアップ手段101、発行手段102、秘密情報記憶部103、公開情報記憶部104、及び通信手段105を備える。

[0114] リスト記憶装置200は、機能上、リスト記憶部201、及び通信手段202を備える。

[0115] ユーザ装置300は、機能上、参加手段301、グループ証明手段302、秘密情報記憶部303、及び公開情報記憶部304、及び通信手段305を備える。

[0116] AP装置400は、機能上、APセットアップ手段401、グループ検証手段402、公開情報記憶部403、履歴記憶部404、及び通信手段405を備える。

[0117] トレース装置500は、機能上、トレース手段501、公開情報記憶部502、及び通信手段503を備える。

[0118] 上記の各装置100～500は、一例としてコンピュータ機(サーバ機、クライアント機等)のCPU、記憶装置、ネットワークインターフェース部、その他の各種入出力装置によって実現される。この例では、各手段101、102、301、302、401、401、501は、各コンピュータ機のCPUが記憶装置上のコンピュータプログラムの命令を実行することにより実現される。ここで用いられるコンピュータプログラムは、これら各手段の処理アルゴリズム(後述参照)に応じて予め設定される。また、公開情報記憶部104、3

WAVELYNX EX. 1003, Pg. 647
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

04、403、502、秘密情報記憶部103、303、リスト記憶部201、履歴記憶部404は、各コンピュータ機の記憶装置により実装される。通信手段105、202、305、405、503は、各コンピュータ機のネットワークインターフェース部に対応する。

[0119] 図1では、もっとも単純な場合として5種類の装置がそれぞれ一台ずつしかない場合を描いているが、各種類の装置が複数台あってもかまわない。また、図1では、各装置100～500が別々の機械である場合を描いているが、一つの機械が二種類の装置の機能を兼ねていてもかまわない。例えばGM装置100の機能とリスト装置200の機能を両方持つ機械を使ってもかまわない。

[0120] GM装置100、リスト記憶装置200、ユーザ装置300、AP装置400、トレース装置500は、それぞれ通信手段105、202、305、405、503を使って相互に通信できる。通信媒体としては、例えばインターネット、電波、電話回線といったものを用いることができるが、本発明ではどのような通信媒体を用いてもかまわない。

[0121] GM装置100、ユーザ装置300、AP装置400、トレース装置500は、それぞれの公開情報記憶部104、304、403、502に、自分自身や他の装置が公開している公開情報を記憶する。また、リスト記憶装置200は、リスト記憶部201という、公開情報を記憶するための部分を持っている。リスト記憶装置200は、自分自身や他の装置が公開している公開情報をリスト記憶部201に記憶する。

[0122] 各装置100～500は、他の装置が公開している公開情報をなんらかの方法で入手できるものとする。公開情報の入手手段としては例えば、公開情報を公開している装置から前述の通信手段を用いて直接受け取る方法、公開情報のリストを持っているサーバから前述の通信手段を用いて受け取る方法、といったものが考えられるが、どのような方法を用いてもかまわない。

[0123] GM装置100、ユーザ装置300は、それぞれ秘密情報記憶部103、303にそれぞれの秘密情報を記憶する。

[0124] 各装置100～500には、事前にセキュリティ・パラメータωが配布されている。セキュリティ・パラメータをどのような方法で配布するのか、セキュリティ・パラメータをどのような方法で決定するのかは問わない。

[0125] ユーザ装置300、AP装置400には、事前に固有のIDが割り振られており、各装置

100～500は、全てのユーザ装置300、AP装置400のIDを事前に知っている。IDとしてどのようなデータを用いるのか、IDをどのように配布するのかは問わない。例えば、装置保有者の名前、装置に割り振られたIPアドレス、装置に割り振られたMACアドレス、および乱数といったものを装置のIDとして用いることができる。

[0126] GM装置100は、GMセットアップ手段101により、発行者用鍵生成手段の処理を行う。以下、発行者用鍵生成手段の詳細を略して、GMセットアップ手段101の処理を説明し、後で発行者用鍵生成手段の詳細を説明する。

[0127] 次に、本実施例の動作を図11～図21を参照して説明する。図11～図21に示す処理手順は、各装置に対応するコンピュータ機の記憶装置上に格納されるコンピュータプログラムとして実現され、各コンピュータ機のCPUにより実行される。

[0128] まず、GMセットアップ手段101の処理を図11に従って説明する。

[0129] 図11において、GM装置100は、まずセキュリティ・パラメータωを公開情報記憶部104から読み込む(ステップS61)。GM装置100は、次に発行者用鍵生成手段の処理を行い、GM公開鍵gpkとGM秘密鍵gskを作り(ステップS62)、gpkを公開情報記憶部104に、gskを秘密情報記憶部103に記憶する(ステップS63)。

[0130] 次に、発行者用鍵生成手段の処理を図16に従って説明する。

[0131] 以下の説明では、G__2、H__2、G__3を有限巡回群とし、<・, ・>はG__2×H__2の元(g, h')にG__3の元<g, h'>を対応させる写像で、<g^x, h'^y>=<g, h'>^{xy}が任意のg, h', x, yに対して成立するものとする。このような組(G__2、H__2、G__3、<・, ・>)の生成方法、および<・, ・>の計算方法は様々なものが知られているが、そのどれを用いてもよい。例えば非特許文献2を参照されたい。qをG__2の位数とし、整数環Zをイデアルの生成元qで割った商環をZ__qと書く。

[0132] 図16において、GM装置100は、G__2の元g__3、h__3、a__3とH__2の元g'__3をランダムに選ぶ(ステップS131、S132)。次に、GM装置1は、Z__qの元gskをランダムに選ぶ(ステップS133)。次に、GM装置1は、u'__3=g__3^{ssk}を計算する(ステップS134)。最後にGM装置1は、(g__3, h__3, a__3, g'__3, u'__3)を発行者用公開鍵gpkとしてセットする(ステップS135)。

[0133] 次に、GM装置100の発行手段102とユーザ装置300の参加手段302の各処理を

説明する。

[0134] GM装置100とユーザ装置300は、相互通信をしながら、それぞれ発行手段102、参加手段302の処理を行う。

[0135] GM装置100は、発行手段102により、第一発行手段、第二発行手段の各処理を行う。ユーザ装置300は、参加手段302により、第一参加手段、ファイ計算手段、メンバー特定情報生成手段、第二参加手段、メンバー鍵検証手段の各処理を行う。

[0136] まず、第一発行手段、第一参加手段、ファイ計算手段、メンバー特定情報生成手段、第二発行手段、第二参加手段、およびメンバー鍵検証手段の詳細を略して、発行手段102と参加手段302の説明を行い、その後で、第一発行手段、第一参加手段、ファイ計算手段、メンバー特定情報生成手段、第二発行手段、第二参加手段、及びメンバー鍵検証手段の詳細を説明する。

[0137] 最初に、GM装置100の発行手段102とユーザ装置300の参加手段302の各処理を図12に従って説明する。

[0138] 図12において、GM装置100は、まず(ω, gpk)を公開情報記憶部104から、gskを秘密情報記憶部103から読み込む(ステップS71)。

[0139] ユーザ装置300は、(ω, gpk)を公開情報記憶部304から読み込む(ステップS81)。

[0140] 次にGM装置100とユーザ装置300は、互いに通信しながら、それぞれ第一発行手段、第一参加手段を行い、GM装置100は、St__{GM}を、ユーザ装置300は、St__{U}とメンバー秘密鍵mskをそれぞれ得る(ステップS72、S82)。ただし、GM装置100とユーザ装置300のいずれかが、それぞれ第一発行手段、第一参加手段の各処理の途中で異常終了した場合は、それぞれ発行手段102、参加手段301の各処理を終了する(ステップS72、S82)。

[0141] 次にGM装置100は、gpkを公開情報記憶部104に、gskを秘密情報記憶部103に書き込む(ステップS73)。

[0142] 次にユーザ装置300は、ファイ計算手段の処理を行い、y＝φ__{ω}(msk)を計算する(ステップS83)。

[0143] 次にユーザ装置300は、メンバー特定情報生成手段の処理を行い、メンバー特定

情報t__1を得る(ステップS84)。次に、ユーザ装置300は、t__1とリスト記憶装置200に送信する(ステップS85)。

[0144] リスト記憶装置200は、受信したt__1をユーザ装置300のIDと組にしてリスト記憶部201に記憶する(ステップS80)。

[0145] 次に、GM装置100は、リスト装置200からt__1を受け取る(ステップS744)。もしリスト装置200がt__1を保管していなかったら、GM装置100は、発行手段の処理を終了する(ステップS75)。リスト装置200からt__1を受け取るのに成功したら、次に、ユーザ装置300は、t__1の正当性をGM装置100に証明し、GM装置100は、その証明を検証する(ステップS86、S76)。正当性証明は、どのような方法で行ってもよい。例えば非特許文献3の方法で行うことができる。

[0146] ユーザ装置300の証明が正当でなければ、GM装置100は、発行手段の処理を終了する(ステップS77)。ユーザ装置300の証明が正当であれば、GM装置100とユーザ装置300は、それぞれ第二発行手段、第二参加手段の各処理を行い、ともにメンバー公開鍵mpkを得る(ステップS78、S87)。

[0147] 次にGM装置100は、mpkを公開情報記憶部104に書き込む(ステップS79)。

[0148] ユーザ装置300は、メンバー鍵検証手段の処理を行うことで、verkey(mpk, msk)を計算し、verkey(mpk, msk)＝acceptなら、(mpk, msk)を公開情報記憶部304に書き込む(ステップS88)。

[0149] 次に、第一発行手段と第一参加手段の各処理を図17に従って説明する。

[0150] 図17において、GM装置100は、まずGM公開鍵gpkを(g__3, h__3, a__3, g'__3, u'__3)とパースする(ステップS141)。ユーザ装置300も、GM公開鍵gpkを(g__3, h__3, a__3, g'__3, u'__3)とパースする(ステップS145)。

[0151] 次にユーザ装置300は、Z__qの元x, r'をランダムに選ぶ(ステップS146)。次に、ユーザ装置300は、w＝a__3g__3^{x}・h__3^{r'}を計算する(ステップS147)。次に、ユーザ装置300は、wをGM装置1に送信する(ステップS148)。次に、GM装置100は、wを受信する(ステップS142)。

[0152] 次にユーザ装置300は、wの正当性をGM装置1に証明し、GM装置100は、その証明を検証する(ステップS149、S143)。証明は、どのような方法で行ってもよいが

WAVELYNX EX. 1003, Pg. 651
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

、例えば非特許文献3の方法で証明することができる。

[0153] 次にGM装置100は、wが正当なら、St__{GM}＝wとして、第一発行手段の処理を正常終了し、そうでなければ、異常終了する(ステップS144)。最後に、ユーザ装置300は、St__{U}＝wとして、第一参加手段の処理を正常終了する(ステップS150)。

[0154] 次に、ファイ計算手段の処理を図18に従って説明する。

[0155] 図18において、ユーザ装置300は、mskを(x, r')とパースする(ステップS161)。ユーザ装置300は、y＝xとする(ステップS162)。

[0156] 次に、メンバー特定情報生成手段の処理を図7に従って説明する。

[0157] ユーザ装置300は、前述した実施例3のメンバー特定情報生成手段の処理(図7)を行い、t__1＝ν__{ω}(y)＝g__1^yを計算する(ステップS22)。ここで、g__1は事前に決められた公開情報である。g__1は、どの装置がどのような方法で公開してもよいが、安全性の観点から言えば、g__1はなんらかのデータのハッシュ値であることが望ましい。

[0158] 次に、第二発行手段と第二参加手段の各処理を図18に従って説明する。

[0159] 図18において、GM装置100は、Z__qの元e, r"をランダムに選び(ステップS151、S152)、v＝(wh__3^{r"})^{1／(gsk＋e)}を計算する(ステップS153)。次にGM装置100は、mpk＝(v, e)とする(ステップS154)。次にGM装置100は、(mpk, r")をユーザ装置300に送信する(ステップS155)。

[0160] 次にユーザ装置300は、(mpk, r")を受信したら(ステップS156)、r＝r'＋r"mod qとし、msk＝(x, r)とする(ステップS157、S158)。

[0161] 次に、メンバー鍵検証手段の処理を図18に従って説明する。

[0162] 図18において、ユーザ装置300は、mpkを(v, e)とパースする(ステップS159)。

[0163] ユーザ装置300は、＜w, u'__3g'__3^{e}＞＝＜v, g'__3＞が成立するかどうかを調べ、＜w, u'__3g'__3^{e}＞＝＜v, g'__3＞が正当ならmpk＝(v, e)とし、そうでなければ異常終了する(ステップS160)。

[0164] 次に、AP装置400のAPセットアップ手段401の処理を図11に従って説明する。

[0165] 図11において、AP装置400は、APセットアップ手段401の処理を行う前に、各ユ

ーザ装置300にアクセスを許す上限回数kを決めておく必要がある。どのような方法でkを決めてもよい。

[0166] AP装置400は、まずセキュリティ・パラメータω、自分の識別子ID、および上限値kを公開情報記憶部403から読み込む(ステップS64)。

[0167] 次にAP装置400は、実施例3のタグ用鍵生成手段の処理(図7)を行い、AP公開鍵apkを得る(ステップS65)。最後に、AP装置400は、apkを公開情報記憶部AP3に書き込む(ステップS66)。

[0168] ユーザ装置300とAP装置400は、互いに通信しながら、それぞれグループ証明手段302、グループ検証手段403の各処理を行う。

[0169] 次に、ユーザ装置300のグループ証明手段302とAP装置400のグループ検証手段303を図13に従って説明する。

[0170] 図13において、まずユーザ装置300は、(ω, gpk, ID, k, apk, mpk, msk)を公開情報記憶部304から読み込む(ステップS91)。

[0171] AP装置400は、(ω, gpk, ID, k, apk)を公開情報記憶部404から読み込む(ステップS101)。

[0172] 次にAP装置400は、ランダムにlを選び(ステップS102)、lをユーザ装置300に送信する(ステップS103)。

[0173] 次にユーザ装置300は、lを受信したら(ステップS92)、実施例3のタグ生成手段の処理(図7)を行い、知識(τ, μ)を生成する(ステップS93)。

[0174] ver__{spk}(β, S)を、<S, h'__2g'__2^β>=<g__2, g'__2>が成立すれば、acceptを、そうでなければ、rejectを出力する関数とする。

[0175] 次にユーザ装置300は、知識(τ, μ)の正当性証明文pf__{τ, μ}を作成し(ステップS94)、(τ, μ, pf__{τ, μ})をAP装置400に送信する(ステップS95)。

[0176] ここで、証明文pf__{τ, μ}の作成方法を図19、及び図20に従って説明する。

[0177] 図19において、まずユーザ装置300は、Z__qの元βを選び、v__{4}=v・h__3^{−β}を計算する(ステップS171)。

[0178] 次にユーザ装置300は、Z__qの元x__{4}、e__{4}、γ__{4}、β__{4}をランダムに選び、X__{4}=<g__3^{x__{4}}v__{4}^{e__{4}}h__3^{γ}, g'__3><

h__3^{β__{4}}, u'__3>を計算する(ステップS172)。

[0179] 次にユーザ装置300は、Z__qの元sをランダムに選び、s'=(x+i)s、b=τ・g^{−i}a__3^sを計算する(ステップS173)。

[0180] 次にユーザ装置300は、Z__qの元i__{4}、s__{4}、s'__{4}を選び、Z__qの元s'__{4}=(x__{4}+i__{4})s__{4} mod q、b__{4}=g^{−i__{4}}a__3^{s__{4}}、h__{4}=b^{x__{4}+i__{4}}a__3^{−s'__{4}}を計算する(ステップS174)。

[0181] 次にユーザ装置300は、Z__qの元tをランダムに選び、t'=(x+i)t mod q、B=μg__1^{−lx}g^{−i}a__3^{t}を計算する(ステップS175)。

[0182] 次にユーザ装置300は、Z__qの元t__{4}、t'__{4}を選び、B__{4}=g__1^{−l・x__{4}}g^{−i__{4}}a__3^{t__{4}}、H__{4}=B^{−x__{4}−i__{4}}a__3^{−t'__{4}}を計算する(ステップS176)。

[0183] 次にユーザ装置300は、Z__qの元ρをランダムに選び、θ=ρx mod q、T=Sh^{ρ}を計算する(ステップS177)。

[0184] 図20において、次にユーザ装置300は、Z__qの元θ__{4}、ρ__{4}をランダムに選び、Y__{4}=<T, h'__2><T, g'__2>^{x__{4}}<h, g'__2>^{−θ__{4}}<h, h'__2>^{−ρ__{4}}を計算する(ステップS178)。

[0185] 次にユーザ装置300は、c=Hash__{Z__q}(gpk, apk, v__{4}, X__{4}, b, b__{4}, h__{4}, B, B__{4}, H__{4}, Y__{4})を計算する(ステップS179)。Hash__{Z__q}は、Z__qに値をとるハッシュ関数を表す。

[0186] 次にユーザ装置300は、x__{5}=cx+x__{4} mod q、e__{5}=ce+e__{4} mod q、r__{5}=c(r+βe)+γ mod q、i__{5}=ci+i__{4} mod q、s__{5}=cs+s__{4} mod q、s'__{5}=cs'+s'__{4} mod q、t__{5}=ct+t__{4} mod q、t'__{5}=ct'+t'__{4} mod q、ρ__{5}=cρ+ρ__{4} mod q、θ__{5}=cθ+θ__{4} mod qを計算する(ステップS180)。

[0187] 最後にユーザ装置300は、pf__{τ, μ}=(b, B, c, x__{5}, e__{5}, r__{5}, i__{5}, s__{5}, s'__{5}, t__{5}, t'__{5}, ρ__{5}, θ__{5})とする。

[0188] ここで、図13に戻り、説明を続ける。

[0189] 図13において、AP装置400は、(τ, μ, pf__{τ, μ})を受信したら(ステップS104)、τがすでに履歴記憶部404に記載されているかどうかを調べ、記載されていたら、rejectを出力してグループ検証手段403の処理を終了する(ステップS105)。

[0190] 次にAP装置400は、pf__{τ, μ}が正当性を検証し、pf__{τ, μ}が正当でなければ、rejectを出力してグループ検証手段を終了し(ステップS106)、pf__{τ, μ}が正当であれば、(τ, μ, l, pf__{τ, μ})を履歴記憶部404に記載し、acceptを出力して、グループ検証手段403の処理を終了する(ステップS107)。

[0191] ここで、pf__{τ, μ}の正当性を検証する方法を図21に従って説明する。

[0192] 図21において、まずAP装置400は、X__{4}=<g__3^{x__{5}}v__{4}^{e__{5}}h__3^{r__{5}}, g'__3><h__3^{r__{5}}, u'__3>(<a__3, g__3>/<v__{4}, u'__3>)^cを計算する(ステップS181)。

[0193] 次にAP装置400は、b__{4}=(τb^{-1})^{-c}g^{-i__{5}}a__3^{s__{5}}を計算する(ステップS182)。

[0194] 次にAP装置400は、h__{4}=h^{-c}b^{x__{4}+i__{4}}a__3^{-s'__{5}}を計算する(ステップS183)。

[0195] 次にAP装置400は、B__{4}=(B^{-1}μ)^{c}g__1^{-l･x__{5}}g^{-i__{5}}a__3^{t__{5}}を計算する(ステップS184)。

[0196] 次にAP装置400は、H__{4}=B^{-x__{5}-i__{rej}}a__3^{-t'__{5}}を計算する(ステップS185)。

[0197] 次にAP装置400は、C__{4}=C^{-c}g^{x__{5}}h^{ρ__{5}}を計算する(ステップS186)。

[0198] 次にAP装置400は、Y__{4}=<g__2, g'__2>^{-c}<T, h'__2><T, g'__2>^{x__{4}}<h, g'__2>^{-θ__{4}}<h, h'__2>^{-ρ__{4}}を計算する(ステップS187)。

[0199] 最後にAP装置400は、c=Hash__{Z__q}(gpk, apk, v__{4}, X__{4}, b, b__{4}, h__{4}, B, B__{4}, H__{4})が成立するかどうかを確認し、c=Hash__{Z__q}(gpk, apk, v__{4}, X__{4}, b, b__{4}, h__{4}, B, B__{4}, H__{4}, Y__{4})が成立していれば、pf__{τ, μ}をacceptし、そうでなければrejectする(ス

テップS188)。

[0200] 次に、トレース装置500のトレース手段501の処理を図14、図15に従って説明する。

[0201] 図14において、まずトレース装置500は、(ω, gpk, ID, k, apk)を公開情報記憶部502から読み込む(ステップS111)。

[0202] 次にトレース装置500は、AP装置400の履歴記憶部404に記憶されているデータ(τ, μ, l, pf__{τ, μ})、(τ', μ', l', pf'__{τ', μ'})を受信する(ステップS112)。このとき、AP装置400がいつトレース装置500に(τ, μ, pf__{τ, μ})、(τ', μ', l', pf'__{τ', μ'})を送信するのかは問わない。

[0203] 次にトレース装置500は、τ = τ'であるかどうかを調べ、τ = τ'でないなら、「AP装置400が不正なデータ(τ, μ, l, pf__{τ, μ})、(τ', μ', l', pf'__{τ', μ'})をトレース装置500に送った」ことを意味する文字列を出力してトレース手段501の処理を終了する(ステップS113)。

[0204] 次にトレース装置500は、l=l'であるかどうかを調べ、l=l'でないなら、「AP装置4が不正なデータ(τ, μ, l, pf__{τ, μ})、(τ', μ', l', pf'__{τ', μ'})をトレース装置500に送った」ことを意味する文字列を出力してトレース手段501の処理を終了する(ステップS114)。

[0205] 次にトレース装置500は、pf__{τ, μ}が正当であるかどうかを調べ、正当でなければ、「AP装置400が不正なデータ(τ, μ, l, pf__{τ, μ})、(τ', μ', l', pf'__{τ', μ'})をトレース装置5に送った」ことを意味する文字列を出力してトレース手段501の処理を終了する(ステップS115)。

[0206] 次にトレース装置500は、pf'__{τ', μ'}が正当であるかどうかを調べ、正当でなければ、「AP装置400が不正なデータ(τ, μ, l, pf__{τ, μ})、(τ', μ', l', pf'__{τ', μ'})をトレース装置500に送った」ことを意味する文字列を出力してトレース手段501の処理を終了する(ステップS116)。

[0207] 図15において、次にトレース装置500は、実施例3のメンバー特定情報抽出手段の処理(図7)を行い、メンバー特定情報t__1を得る(ステップS117)。

[0208] 次にトレース装置500は、t__1をリスト記憶装置200に送信する(ステップS118)。

リスト記憶装置200は、t__1を受信すると(ステップS121)、t__1に対応するIDをトレース装置500に送信する(ステップS122)。そのようなIDがなければ、ID＝GMとし、ID＝GMを送信する。

[0209] 次にトレース装置500は、t__1に対応するIDを受け取る(ステップS122、S119)。最後にトレース装置500は、IDを出力する(ステップS120)。

[0210] 従って、本実施例でも、前述の実施例3、4と同様に、回数制限匿名認証は既存の方式とは異なり、ユーザが計算すべきデータ個数がO(log k)個であるため、認証時におけるユーザの計算量は制限回数kに比例しないから、効率的な回数制限匿名認証方式を実現することができる。

実施例 6

[0211] 本実施例は、前述した擬似ランダム関数計算装置を用いた回数制限匿名認証システムに適用したものである。

[0212] 前述の図1を参照して、本実施例の装置構成を説明する。

[0213] 図1に示す本実施例の回数制限匿名認証システムは、前述した実施例3、4のシステムに様々な計算手順を追加して構成されており、機能上、GM装置100、リスト記憶装置200、ユーザ装置300、AP装置400、及びトレース装置500の5種類の装置を備える。前述した擬似ランダム関数計算装置は、後述するグループ署名手段(グループ証明手段、グループ検証手段)に適用されている。

[0214] このうち、GM装置100、リスト記憶装置200、ユーザ装置300、およびトレース装置500の装置構成は、前述した実施例5の装置構成と同じである。AP装置400は、前述した実施例5と同様に、グループ検証手段402、公開情報記憶部403、履歴記憶部404、通信手段405を備えるが、本実施例では、実施例5とは異なり、APセットアップ手段401が含まれない。従って、本実施例の装置構成は、図1の装置構成からAPセットアップ手段401を削除したものに対応する。

[0215] 上記の各装置100～500は、一例としてコンピュータ機(サーバ機、クライアント機等)のCPU、記憶装置、ネットワークインターフェース部、その他の各種入出力装置によって実現される。この例では、各手段101、102、301、302、501は、各コンピュータ機のCPUが記憶装置上のコンピュータプログラムの命令を実行することにより実

WAVELYNX EX. 1003, Pg. 657
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

現される。ここで用いられるコンピュータプログラムは、これら各手段の処理アルゴリズム（後述参照）に応じて予め設定される。また、公開情報記憶部104、304、403、502、秘密情報記憶部103、303、リスト記憶部201、履歴記憶部404は、各コンピュータ機の記憶装置により実装される。通信手段105、202、305、405、503は、各コンピュータ機のネットワークインターフェース部に対応する。

[0216] 図1では、もっとも単純な場合として5種類の装置がそれぞれ一台ずつしかない場合を描いているが、各種類の装置が複数台あってもかまわない。また、図1では、各装置が別々の機械である場合を描いているが、一つの機械が二種類の装置の機能を兼ねていてもかまわない。例えば、GM装置100の機能とリスト装置200の機能を両方持つ機械を使ってもかまわない。

[0217] GM装置100、リスト記憶装置200、ユーザ装置300、AP装置400、トレース装置500は、それぞれ通信手段105、202、305、405、503を使って相互に通信できる。通信媒体として、例えばインターネット、電波、電話回線といったものを用いることができるが、どのような通信媒体を用いてもかまわない。

[0218] GM装置100、ユーザ装置300、AP装置400、トレース装置500は、それぞれの公開情報記憶部104、304、403、502に、自分自身や他の装置が公開している公開情報を記憶する。また、リスト記憶装置200は、リスト記憶部201という、公開情報を記憶するための部分を持っている。リスト記憶装置200は、自分自身や他の装置が公開している公開情報をリスト記憶部201に記憶する。

[0219] 各装置100～500は、他の装置が公開している公開情報をなんらかの方法で入手できるものとする。公開情報の入手手段としては例えば、公開情報を公開している装置から前述の通信手段を用いて直接受け取る方法、公開情報のリストを持っているサーバから前述の通信手段を用いて受け取る方法、といったものが考えられるが、どのような方法を用いてもかまわない。

[0220] GM装置100とユーザ装置300は、それぞれ秘密情報記憶部103、303にそれぞれの秘密情報を記憶する。

[0221] 各装置100～300には、事前にセキュリティ・パラメータωが配布されている。セキュリティ・パラメータωをどのような方法で配布するのか、セキュリティ・パラメータωを

どのような方法で決定するのかは問わない。

[0222] ユーザ装置300とAP装置400には、事前に固有のIDが割り振られており、各装置100～500は、全てのユーザ装置300、AP装置400のIDを事前に知っている。IDとしてどのようなデータを用いるのか、IDをどのように配布するのかは問わない。例えば、装置保有者の名前、装置に割り振られたIPアドレス、装置に割り振られたMACアドレス、および乱数といったものを装置のIDとして用いることができる。

[0223] グループ証明手段302、グループ検証手段402、トレース手段501以外の、実施例6の各手段は、実施例5のそれと同じである。

[0224] 次に、本実施例の動作を図13、図22～図24を参照して説明する。図13、図22～図24に示す処理手順は、各装置に対応するコンピュータ機の記憶装置上に格納されるコンピュータプログラムとして実現され、各コンピュータ機のCPUにより実行される。

[0225] まず、グループ証明手段302とグループ検証手段402の各処理を図13に従って説明する。

[0226] 図13のステップS93、S94以外の処理は、前述した実施例5のグループ証明手段302と同じである。図13のステップS104以外の処理は、前述した実施例5のグループ検証手段402と同じである。

[0227] グループ証明手段302は、ステップS93にて、実施例3ではなく実施例4のタグ生成手段の処理(図10)を行い、ステップS104にて、図19、図20の代わりに図22、図23に従って、証明文pf__{τ, μ}を作成する。

[0228] 図22及び図23を参照して、証明文pf__{τ, μ}の作成方法を説明する。

[0229] 図22において、まずユーザ装置300は、Z__qの元βを選び、v__{4}＝v･h__3^{－β}を計算する(ステップS191)。

[0230] 次にユーザ装置300は、Z__qの元x__{4}、e__{4}、γ__{4}、β__{4}をランダムに選び、X__{4}＝<g__3^{x__{4}}v__{4}^{e__{4}}h__3^{γ}, g'__3><h__3^{β__{4}}, u'__3>を計算する(ステップS192)。

[0231] 次にユーザ装置300は、Z__qの元sをランダムに選び、s'＝(x＋i)s、b＝τ･g^{－i}a__3^sを計算する(ステップS193)。

[0232] 次にユーザ装置300は、Z__qの元i__{4}、s__{4}、s'__{4}を選び、Z__qの元s'__{4}＝(x__{4}＋i__{4})s__{4} mod q、b__{4}＝g^{－i__{4}}a__3^{s__{4}}、h__{4}＝b^{x__{4}＋i__{4}}a__3^{－s'__{4}}を計算する(ステップS194)。

[0233] 次にユーザ装置300は、Z__qの元tをランダムに選び、t'＝(x＋i)t mod q 、B＝μg__1^{－lx}g^{－i}a__3^{t}を計算する(ステップS195)。

[0234] 次にユーザ装置300は、Z__qの元t__{4}、t'__{4}を選び、B__{4}＝g__1^{－l･x__{4}}g^{－i__{4}}a__3^{t__{4}}、H__{4}＝B^{－x__{4}－i__{4}}a__3^{－t'__{4}}を計算する(ステップS196)。

[0235] 次にユーザ装置300は、log__2 k以上の整数で最も小さい整数をNとする(ステップS197)。

[0236] 図23において、次にユーザ装置300は、整数i＝0, . . , Nに対し、xの第iビットをx__iとし、x'__i＝1－x__iとし、z＝k－xとし、zの第iビットをz__iとし、z'__i＝1－z__iとする(ステップS198)。

[0237] 次にユーザ装置300は、Z__qのρ__1, …, ρ__N、θ__1, …, θ__Nで、ρ__1＋…＋ρ__N＝θ__1＋…＋θ__Nが成立するものを選び、ρ＝ρ__1＋…＋ρ__N、C__1＝h^{ρ__1}, …, C__N＝h^{ρ__N}、D__1＝h^{θ__1}, …, D__N＝h^{θ__N}、C＝g^{x}h^{ρ}、D＝g^{k－x}h^{ρ}を計算する(ステップS199)。

[0238] 次にユーザ装置300は、i＝1, …, N, j＝0, 1に対し、Z__qの元ρ__{4, i}, θ__{4, i}, c'__{i, x'__i}, d'__{i, z'__i}, ρ__{5, i, x'__i}, θ__{5, i, z'__i}をランダムに選び、C__{4, i, x__i}＝g^{x__i}h^{ρ__{4, i}}、D__{4, i, x__i}＝g^{z__i}h^{θ__{4, i}}、C__{4, i, x'__i}＝C^{－c'__{i, x'__i}}g^{x'__i}h^{ρ__{5, i, x'__i}}、D__{4, i, x'__i}＝D^{－d'__{i, z'__i}}g^{z'__i}h^{θ__{5, i, z'__i}}を計算する(ステップS200)。

[0239] 次にユーザ装置300は、c＝Hash__{Z__q}(gpk, apk, v__{4}, X__{4}, b, b__{4}, h__{4}, B, B__{4}, H__{4}, C__{4}, {C__{4, i, j}}__{i＝1, …, N, j＝0, 1}, {D__{4, i, j}}__{i＝1, …, N, j＝0, 1})を計算する(ステップS201)。

Hash__{Z__q}は、Z__qに値をとるハッシュ関数を表す。

[0240] 次にユーザ装置300は、x__{5}＝cx＋x__{4} mod q、e__{5}＝ce＋e__{4} mod q、r__{5}＝c(r＋βe)＋γ mod q、i__{5}＝ci＋i__{4} mod q、s__{5}＝cs＋s__{4} mod q、s'__{5}＝cs'＋s'__{4} mod q、t__{5}＝ct＋t__{4} mod q、t'__{5}＝ct'＋t'__{4} mod q、c__i＝c－c'__i mod q、d__i＝d－d'__i mod q、ρ__{5, i, x__i}＝c__iρ__i＋ρ__{4, i} mod q、θ__{5, i, x__i}＝c__iθ__i＋θ__{4, i} mod qを計算する(ステップS202)。

[0241] 最後にユーザ装置3は、pf__{τ, μ}＝(b, B, C, c, x__{5}, e__{5}, r__{5}, i__{5}, s__{5}, s'__{5}, t__{5}, t'__{5}, {c__{ij}})とする(ステップS203)。

[0242] 図13において、AP装置400は、(τ, μ, pf__{τ, μ})を受信したら(ステップS104)、τがすでに履歴記憶部BAP4に記載されているかどうかを調べ、記載されていたらrejectを出力してグループ検証手段402の処理を終了する(ステップS105)。

[0243] 次にAP装置400は、pf__{τ, μ}が正当性を検証し、pf__{τ, μ}が不当なら、rejectを出力してグループ検証手段402の処理を終了し(ステップS106)、pf__{τ, μ}が正当なら、AP装置400は、(τ, μ, l, pf__{τ, μ})を履歴記憶部404に記載し、acceptを出力してグループ検証手段402の処理を終了する(ステップS107)。

[0244] 次に、pf__{τ, μ}の正当性を検証する方法を図24に従って説明する。

[0245] 図24において、まずAP装置400は、X__{4}＝＜g__3^{x__{5}}v__{4}^{e__{5}}h__3^{r__{5}}, g'__3＞＜h__3^{r__{5}}, u'__3＞(＜a__3, g__3＞／＜v__{4}, u'__3＞)^cを計算する(ステップS211)。

[0246] 次にAP装置400は、b__{4}＝(τb^{－1})^{－c}g^{－i__{5}}a__3^{s__{5}}を計算する(ステップS212)。

[0247] 次にAP装置400は、h__{4}＝h^{－c}b^{x__{4}＋i__{4}}a__3^{－s'__{5}}を計算する(ステップS213)。

[0248] 次にAP装置400は、B__{4}＝(B^{－1}μ)^{c}g__1^{－l･x__{5}}g^{－i__{5}}a__3^{t__{5}}を計算する(ステップS214)。

[0249] 次にAP装置400は、H__{4}＝B^{－x__{5}－i__{rej}}a__3^{－t'__{5}}を

計算する(ステップS215)。

[0250] 次にAP装置400は、C__{4}=C^{−c}g^{x__{5}}h^{ρ__{5}}を計算する(ステップS216)。

[0251] 次にAP装置400は、i=1, . . , N、j=0, 1に対し、C__{4, i, j}=C__{−c__{ij}}g^{j}h^{ρ__{5, ij}}、D__{4, i, j}=D__{−c__{ij}}g^{j}h^{θ__{5, ij}}を計算する(ステップS217)。

[0252] 最後にAP装置400は、c=Hash__{Z__q}(gpk, apk, v__{4}, X__{4}, b, b__{4}, h__{4}, B, B__{4}, H__{4}, {C__{ij}}__{i=1, …, N, j=0, 1}, {D__{ij}}__{i=1, …, N, j=0, 1})と、c=c__1+c'__1=…=c__N+c'__N mod qが成立するかどうかを確認し、ともに成立していれば、pf__{τ, μ}をacceptし、そうでなければrejectする(ステップS218)。

[0253] 従って、本実施例でも、前述の実施例3〜5と同様に、回数制限匿名認証は既存の方式とは異なり、ユーザが計算すべきデータ個数がO(log k)個であるため、認証時におけるユーザの計算量は制限回数kに比例しないから、効率的な回数制限匿名認証方式を実現することができる。

[0254] 以上、本発明の各実施例を詳細に説明したが、本発明は、代表的に例示した上述の各実施例に限定されるものではなく、当業者であれば、請求の範囲の記載内容に基づき、本発明の要旨を逸脱しない範囲内で種々の態様に変形、変更することができる。これらの変形例や変更例も、本発明の権利範囲に属する。

[0255] 例えば、前述した各実施例を構成する各部の少なくとも一部の機能は、プログラム制御で動作するプロセッサ(CPU)と、制御プログラムや制御データ等を格納する記憶領域を有するメモリ(ROM/RAM)と、各種入出力装置、例えば、ハードディスクドライブ等の外部記録装置、通信モデムやLANインタフェース等の通信装置、CRTや液晶ディスプレイ等の表示装置、キーボードやポインティングデバイス等の入力装置等の各種周辺装置とを用いて実現することが可能である。この場合、プロセッサ、メモリ、及び各種入出力装置の各構成要素は、本発明の範疇に含まれる。

[0256] また、前述した各実施例を構成する各部の少なくとも一部の機能をプログラムコードを用いて実現する場合、かかるプログラムコード及びこれを記録する記録媒体は、本

発明の範疇に含まれる。この場合のプログラムコードは、オペレーティングシステムや他のアプリケーションソフト等と共同して上記機能が実現される場合は、それらのプログラムコードも含まれる。また、記録媒体としては、前述したハードディスクやROMのほか、フレキシブルディスク、光ディスク、光磁気ディスク、CD－ROM、磁気テープ、不揮発性のメモリカード等を用いることができる。

WAVELYNX EX. 1003, Pg. 663
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

請求の範囲

[1] 有限群の元を構成する要素として少なくとも第一の要素及び第二の要素を持つ組からなる公開鍵と、整数からなる秘密鍵とを生成し、生成された前記秘密鍵を記憶装置に秘密裡に保管し、生成された前記公開鍵を公開する鍵生成手段と、

整数が入力されると、擬似ランダム関数の関数値として、前記有限群の元を出力する擬似ランダム関数計算手段とを有し、

前記擬似ランダム関数計算手段は、前記有限群の元として、

前記公開鍵の前記第一の要素を底とし、前記入力された整数を指数として、べき乗剰余を計算して得られる値からなる第一の元と、

前記公開鍵の前記第二の要素を底とし、前記秘密鍵と前記入力された整数との和の有限体での逆数を指数として、べき乗剰余を計算して得られる値からなる第二の元と、

の積を出力することを特徴とする擬似ランダム関数計算装置。

[2] 整数からなる秘密鍵を生成し、生成された前記秘密鍵を記憶装置に秘密裡に保管する鍵生成手段と、

ビット列と整数との組が入力されると、擬似ランダム関数の関数値として、有限群の元を出力する擬似ランダム関数計算手段とを有し、

前記擬似ランダム関数計算手段は、前記有限群の元として、

前記入力された値で決まる値を底とし、前記入力された整数を指数として、べき乗剰余を計算して得られる値からなる第一の元と、

前記入力された値で決まる値を底とし、前記秘密鍵と前記入力された整数の和の逆数を指数として、べき乗剰余を計算して得られる値を指数として、べき乗剰余演算により得られる値からなる第二の元と、

の積を出力することを特徴とする擬似ランダム関数計算装置。

[3] 前記底は、前記入力された値のハッシュ値であることを特徴とする請求項2記載の擬似ランダム関数計算装置。

[4] 請求項1に記載の擬似ランダム関数計算装置を用いた回数制限匿名認証システムであって、

WAVELYNX EX. 1003, Pg. 664
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

識別子と、整数k、i、y、及びlと、有限群の元tとを入力するための入力手段と、

前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、前記有限群に値をとる擬似ランダム関数の関数値を計算する第一タグ計算手段と、

前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、前記有限群に値をとる擬似ランダム関数の関数値を計算し、計算された擬似ランダム関数の関数値をl乗した値にtを乗じた値を計算する第二タグ計算手段と、

前記第一タグ計算手段の計算結果と前記第二タグ計算手段の計算結果との組を出力する出力手段とを有するタグ生成手段を備えたことを特徴とする回数制限匿名認証システム。

[5] 請求項2又は3に記載の擬似ランダム関数計算装置を用いた回数制限匿名認証システムであって、

識別子と、整数k、i、y、及びlと、有限群の元tとを入力するための入力手段と、

前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、前記有限群に値をとる擬似ランダム関数の関数値を計算する第一タグ計算手段と、

前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、前記有限群に値をとる擬似ランダム関数の関数値を計算し、計算された擬似ランダム関数の関数値をl乗した値にtを乗じた値を計算する第二タグ計算手段と、

前記第一タグ計算手段の計算結果と前記第二タグ計算手段の計算結果との組を出力する出力手段と、

を有するタグ生成手段を備えたことを特徴とする回数制限匿名認証システム。

[6] 整数kを入力するための入力手段と、

電子署名方式の公開鍵及び秘密鍵ペアを選ぶ電子署名用鍵生成手段と、

k個の整数を選ぶ平文選択手段と、

前記k個の整数各々に対する署名文を前記公開鍵及び秘密鍵ペアを用いて計算する電子署名計算手段と、

前記電子署名方式の公開鍵と前記k個の整数と前記k個の署名文とからなる組を、前記タグ生成手段の計算に使用されるタグ用公開鍵として出力する出力手段と、

を有するタグ用鍵生成手段を更に備えたことを特徴とする請求項4記載の回数制限

WAVELYNX EX. 1003, Pg. 665
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

匿名認証システム。

[7] 前記電子署名計算手段は、

平文として整数を入力するための手段と、

平文と整数の和の有限体での逆元を計算する手段と、

計算された前記逆元を指数としてべき乗剰余を計算し、前記べき乗剰余の計算結果を含む組を前記タグ用公開鍵として出力する手段とを有することを特徴とする請求項6記載の回数制限匿名認証システム。

[8] 前記電子署名用鍵生成手段は、

有限群から元を選ぶ手段と、

整数を選ぶ手段と、

前記元を底とし、前記整数を指数として、べき乗剰余を計算する手段と、

選ばれた前記有限群の元と前記べき乗剰余の計算結果とからなる組を出力する手段とを有することを特徴とする請求項7記載の回数制限匿名認証システム。

[9] 前記タグ生成手段に整数lを入力して計算される計算結果をτとし、前記タグ生成手段に整数l'を入力して計算される計算結果をτ'としたとき、前記τ、l、τ'、l'の四つのデータを入力するための入力手段と、

前記τを前記τ'で割った値を底とし、前記lから前記l'を減じた値の有限体での逆数を指数として、べき乗剰余を計算する計算手段と、

前記べき乗剰余の計算結果を出力する出力手段と、

を有するメンバー特定情報抽出手段を更に備えたことを特徴とする請求項4記載の回数制限匿名認証システム。

[10] 前記タグ生成手段に整数lを入力して計算される計算結果をτとし、前記タグ生成手段に整数l'を入力して計算される計算結果をτ'としたとき、前記τ、l、τ'、l'の四つのデータを入力するための入力手段と、

前記τを前記τ'で割った値を底とし、前記lから前記l'を減じた値の有限体での逆数を指数として、べき乗剰余を計算する計算手段と、

前記べき乗剰余の計算結果を出力する出力手段と、

を有するメンバー特定情報抽出手段を更に備えたことを特徴とする請求項5記載の

WAVELYNX EX. 1003, Pg. 666
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

回数制限匿名認証システム。

[11] グループメンバーとしての公開鍵及び秘密鍵ペアと、アプリケーション提供者（以下、AP）装置の公開鍵と、前記AP装置の識別子と、整数k、i、及びlとを入力するための入力手段と、

前記グループメンバーとしての秘密鍵から整数yを作り、前記AP装置の識別子と、前記k、i、l、及びyとを入力として、前記タグ生成手段によりタグを成すデータを計算する手段と、

前記タグの正当性証明文を計算する正当性証明手段と、

前記タグと前記正当性証明文とを出力する出力手段と、

を有するグループ証明手段を更に備えたことを特徴とする請求項4記載の回数制限匿名認証システム。

[12] グループメンバーとしての公開鍵及び秘密鍵ペアと、アプリケーション提供者（以下、AP）装置の公開鍵と、前記AP装置の識別子と、整数k、i、及びlとを入力するための入力手段と、

前記グループメンバーとしての秘密鍵から整数yを作り、前記AP装置の識別子と、前記k、i、l、及びyとを入力として、前記タグ生成手段によりタグを成すデータを計算する手段と、

前記タグの正当性証明文を計算する正当性証明手段と、

前記タグと前記正当性証明文とを出力する出力手段と、

を有するグループ証明手段を更に備えたことを特徴とする請求項5記載の回数制限匿名認証システム。

[13] 有限群の元τと有限群の元μと整数lと証明文pとを要素とする第一の組と、有限群の元τ'と有限群の元μ'と整数l'と証明文p'とを要素とする第二の組とを入力するための入力手段と、

前記τと前記τ'とが同一であるか否かを判定する第一判定手段と、

前記lと前記l'とが同一であるか否かを判定する第二判定手段と、

前記証明文pが正当であるか否かを判定する第三判定手段と、

前記証明文p'が正当であるか否かを判定する第四判定手段と、

WAVELYNX EX. 1003, Pg. 667
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

予め設定された対応表に基づいて、前記メンバー特定情報抽出手段の計算結果と対応する識別子を取得する識別子取得手段と、

を有するトレース手段を更に有することを特徴とする請求項9記載の回数制限匿名認証システム。

[14] 有限群の元τと有限群の元μと整数lと証明文pとを要素とする第一の組と、有限群の元τ'と有限群の元μ'と整数l'と証明文p'とを要素とする第二の組とを入力するための入力手段と、

前記τと前記τ'とが同一であるか否かを判定する第一判定手段と、

前記lと前記l'とが同一であるか否かを判定する第二判定手段と、

前記証明文pが正当であるか否かを判定する第三判定手段と、

前記証明文p'が正当であるか否かを判定する第四判定手段と、

予め設定された対応表に基づいて、前記メンバー特定情報抽出手段の計算結果と対応する識別子を取得する識別子取得手段と、

を有するトレース手段を更に有することを特徴とする請求項10記載の回数制限匿名認証システム。

[15] 有限群の元を構成する要素として少なくとも第一の要素及び第二の要素を持つ組からなる公開鍵と、整数からなる秘密鍵とを生成し、生成された前記秘密鍵を記憶装置に秘密裡に保管し、生成された前記公開鍵を公開する鍵生成ステップと、

整数が入力されると、擬似ランダム関数の関数値として、前記有限群の元を出力する擬似ランダム関数計算ステップとを有し、

前記擬似ランダム関数計算ステップは、前記有限群の元として、

前記公開鍵の前記第一の要素を底とし、前記入力された整数を指数として、べき乗剰余を計算して得られる値からなる第一の元と、

前記公開鍵の前記第二の要素を底とし、前記秘密鍵と前記入力された整数との和の有限体での逆数を指数として、べき乗剰余を計算して得られる値からなる第二の元と、

の積を出力することを特徴とする擬似ランダム関数計算方法。

[16] 整数からなる秘密鍵を生成し、生成された前記秘密鍵を記憶装置に秘密裡に保管

WAVELYNX EX. 1003, Pg. 668
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

する鍵生成ステップと、

ビット列と整数との組が入力されると、擬似ランダム関数の関数値として、有限群の元を出力する擬似ランダム関数計算ステップとを有し、

前記擬似ランダム関数計算ステップは、前記有限群の元として、

前記入力された値で決まる値を底とし、前記入力された整数を指数として、べき乗剰余を計算して得られる値からなる第一の元と、

前記入力された値で決まる値を底とし、前記秘密鍵と前記入力された整数の和の逆数を指数として、べき乗剰余を計算して得られる値を指数として、べき乗剰余演算により得られる値からなる第二の元と、

の積を出力することを特徴とする擬似ランダム関数計算方法。

[17] 前記底は、前記入力された値のハッシュ値であることを特徴とする請求項16記載の擬似ランダム関数計算方法。

[18] 請求項15に記載の擬似ランダム関数計算方法を用いた回数制限匿名認証方法であって、

識別子と、整数k、i、y、及びlと、有限群の元tとを入力するためのステップと、

前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、前記有限群に値をとる擬似ランダム関数の関数値を計算する第一タグ計算ステップと、

前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、前記有限群に値をとる擬似ランダム関数の関数値を計算し、計算された擬似ランダム関数の関数値をl乗した値にtを乗じた値を計算する第二タグ計算ステップと、

前記第一タグ計算ステップの計算結果と前記第二タグ計算ステップの計算結果との組を出力するステップとを有するタグ生成ステップを備えたことを特徴とする回数制限匿名認証方法。

[19] 請求項16又は17に記載の擬似ランダム関数計算方法を用いた回数制限匿名認証方法であって、

識別子と、整数k、i、y、及びlと、有限群の元tとを入力するためのステップと、

前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、

WAVELYNX EX. 1003, Pg. 669
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

前記有限群に値をとる擬似ランダム関数の関数値を計算する第一タグ計算ステップと、

前記yを秘密鍵として用い、前記識別子と前記kと前記iとから決まる値を入力して、前記有限群に値をとる擬似ランダム関数の関数値を計算し、計算された擬似ランダム関数の関数値をl乗した値にtを乗じた値を計算する第二タグ計算ステップと、

前記第一タグ計算手段の計算結果と前記第二タグ計算手段の計算結果との組を出力するステップと、

を有するタグ生成ステップを備えたことを特徴とする回数制限匿名認証方法。

[20] 整数kを入力するための入力ステップと、

電子署名方式の公開鍵及び秘密鍵ペアを選ぶ電子署名用鍵生成ステップと、

k個の整数を選ぶ平文選択ステップと、

前記k個の整数各々に対する署名文を前記公開鍵及び秘密鍵ペアを用いて計算する電子署名計算ステップと、

前記電子署名方式の公開鍵と前記k個の整数と前記k個の署名文とからなる組を、前記タグ生成手段の計算に使用されるタグ用公開鍵として出力する出力ステップと、を有するタグ用鍵生成ステップを更に備えたことを特徴とする請求項18記載の回数制限匿名認証方法。

[21] 前記電子署名計算ステップは、

平文として整数を入力するためのステップと、

平文と整数の和の有限体での逆元を計算するステップと、

計算された前記逆元を指数としてべき乗剰余を計算し、前記べき乗剰余の計算結果を含む組を前記タグ用公開鍵として出力するステップとを有することを特徴とする請求項20記載の回数制限匿名認証方法。

[22] 前記電子署名用鍵生成ステップは、

有限群から元を選ぶステップと、

整数を選ぶステップと、

前記元を底とし、前記整数を指数として、べき乗剰余を計算するステップと、

選ばれた前記有限群の元と前記べき乗剰余の計算結果とからなる組を出力するス

WAVELYNX EX. 1003, Pg. 670
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

テップ

とを有することを特徴とする請求項21記載の回数制限匿名認証方法。

[23] 前記タグ生成ステップにて整数lを入力して計算される計算結果をτとし、前記タグ生成ステップにて整数l'を入力して計算される計算結果をτ'としたとき、前記τ、l、τ'、l'の四つのデータを入力するためのステップと、

前記τを前記τ'で割った値を底とし、前記lから前記l'を減じた値の有限体での逆数を指数として、べき乗剰余を計算するステップと、

前記べき乗剰余の計算結果を出力するステップと、

を有するメンバー特定情報抽出ステップを更に備えたことを特徴とする請求項18記載の回数制限匿名認証方法。

[24] 前記タグ生成ステップにて整数lを入力して計算される計算結果をτとし、前記タグ生成ステップにて整数l'を入力して計算される計算結果をτ'としたとき、前記τ、l、τ'、l'の四つのデータを入力するためのステップと、

前記τを前記τ'で割った値を底とし、前記lから前記l'を減じた値の有限体での逆数を指数として、べき乗剰余を計算するステップと、

前記べき乗剰余の計算結果を出力するステップと、

を有するメンバー特定情報抽出ステップを更に備えたことを特徴とする請求項19記載の回数制限匿名認証方法。

[25] グループメンバーとしての公開鍵及び秘密鍵ペアと、アプリケーション提供者(以下、AP)装置の公開鍵と、前記AP装置の識別子と、整数k、i、及びlとを入力するためのステップと、

前記グループメンバーとしての秘密鍵から整数yを作り、前記AP装置の識別子と、前記k、i、l、及びyとを入力として、前記タグ生成ステップにてタグを成すデータを計算するステップと、

前記タグの正当性証明文を計算するステップと、

前記タグと前記正当性証明文とを出力するステップと、

を有するグループ証明ステップを更に備えたことを特徴とする請求項18記載の回数制限匿名認証方法。

WAVELYNX EX. 1003, Pg. 671
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[26] グループメンバーとしての公開鍵及び秘密鍵ペアと、アプリケーション提供者（以下、AP）装置の公開鍵と、前記AP装置の識別子と、整数k、i、及びlとを入力するためのステップと、

前記グループメンバーとしての秘密鍵から整数yを作り、前記AP装置の識別子と、前記k、i、l、及びyとを入力として、前記タグ生成ステップにてタグを成すデータを計算するステップと、

前記タグの正当性証明文を計算するステップと、

前記タグと前記正当性証明文とを出力するステップと、

を有するグループ証明ステップを更に備えたことを特徴とする請求項19記載の回数制限匿名認証方法。

[27] 有限群の元τと有限群の元μと整数lと証明文pとを要素とする第一の組と、有限群の元τ'と有限群の元μ'と整数l'と証明文p'とを要素とする第二の組とを入力するためのステップと、

前記τと前記τ'とが同一であるか否かを判定するステップと、

前記lと前記l'とが同一であるか否かを判定するステップと、

前記証明文pが正当であるか否かを判定するステップと、

前記証明文p'が正当であるか否かを判定するステップと、

予め設定された対応表に基づいて、前記メンバー特定情報抽出ステップの計算結果と対応する識別子を取得するステップと、

を有するトレースステップを更に備えたことを特徴とする請求項23記載の回数制限匿名認証方法。

[28] 有限群の元τと有限群の元μと整数lと証明文pとを要素とする第一の組と、有限群の元τ'と有限群の元μ'と整数l'と証明文p'とを要素とする第二の組とを入力するためのステップと、

前記τと前記τ'とが同一であるか否かを判定するステップと、

前記lと前記l'とが同一であるか否かを判定するステップと、

前記証明文pが正当であるか否かを判定するステップと、

前記証明文p'が正当であるか否かを判定するステップと、

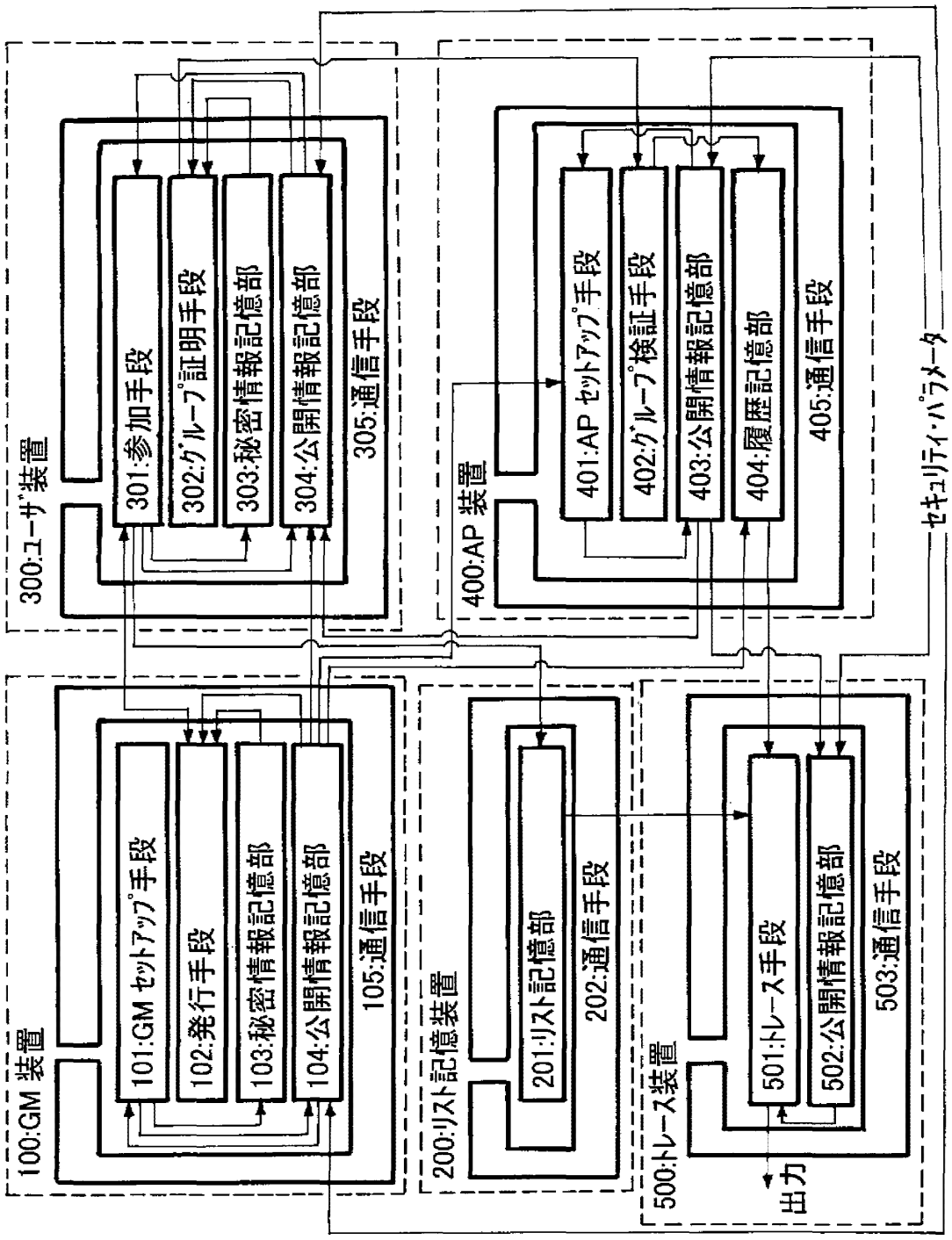
WAVELYNX EX. 1003, Pg. 672
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

予め設定された対応表に基づいて、前記メンバー特定情報抽出ステップの計算結果と対応する識別子を取得するステップと、
を有するトレースステップを更に備えたことを特徴とする請求項24記載の回数制限匿名認証方法。

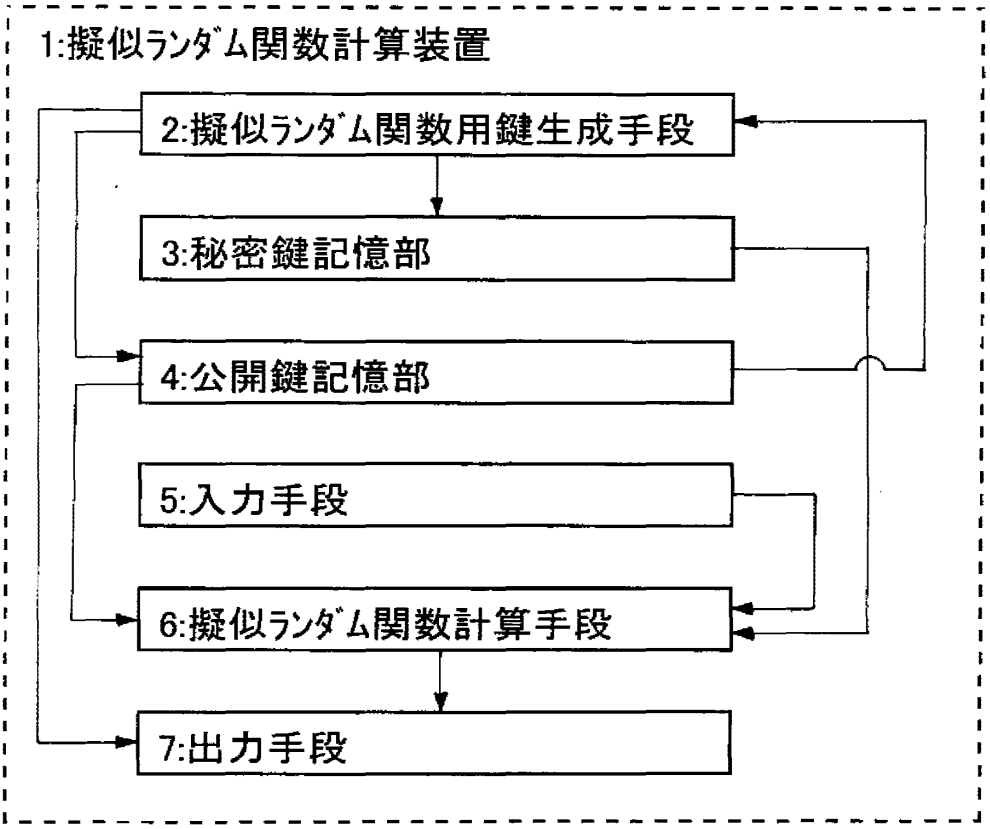
WAVELYNX EX. 1003, Pg. 673
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

WO 2006/126668 PCT/JP2006/310534

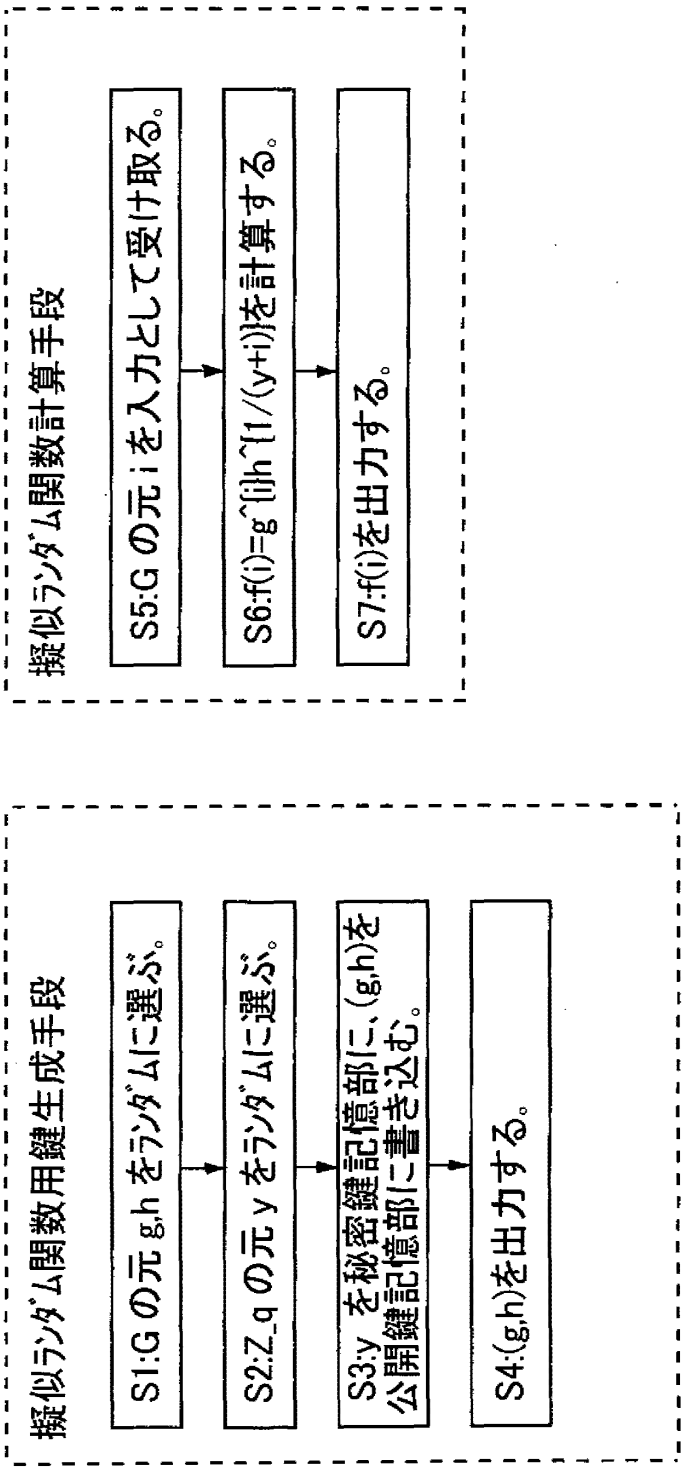
[図1]



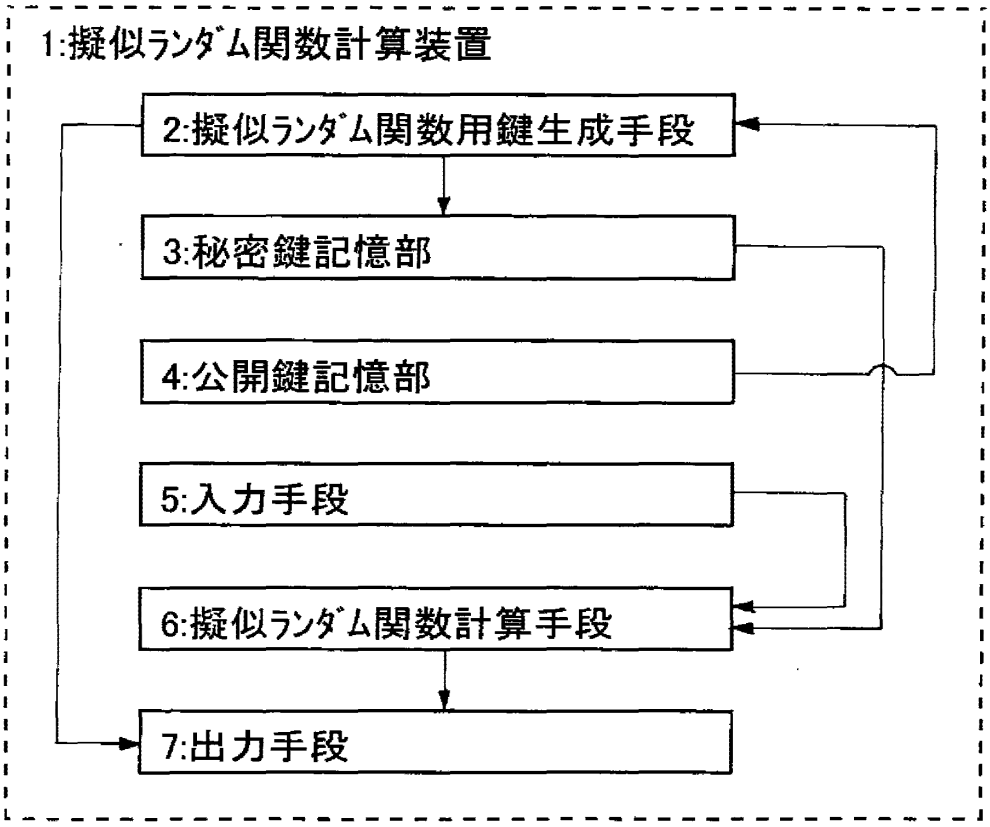
[図2]


1:擬似ランダム関数計算装置
2:擬似ランダム関数用鍵生成手段
3:秘密鍵記憶部
4:公開鍵記憶部
5:入力手段
6:擬似ランダム関数計算手段
7:出力手段

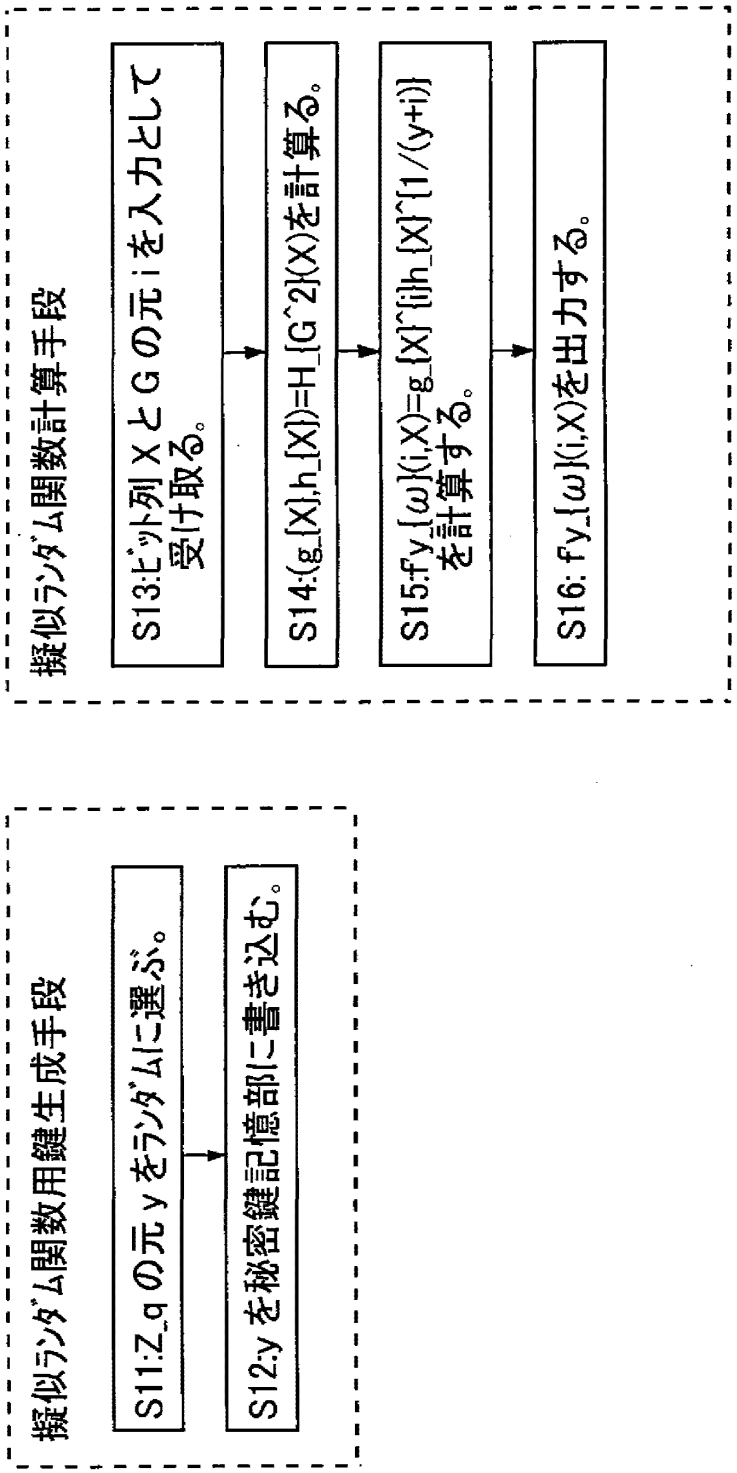
[図3]


擬似ランダム関数用鍵生成手段
S1:G の元 g,h をランダムに選ぶ。
S2:Z_q の元 y をランダムに選ぶ。
S3:y を秘密鍵記憶部に、(g,h)を公開鍵記憶部に書き込む。
S4:(g,h)を出力する。
擬似ランダム関数計算手段
S5:G の元 i を入力として受け取る。
S6:f(i)=g^{ih^{1/(y+i)}}を計算する。
S7:f(i)を出力する。

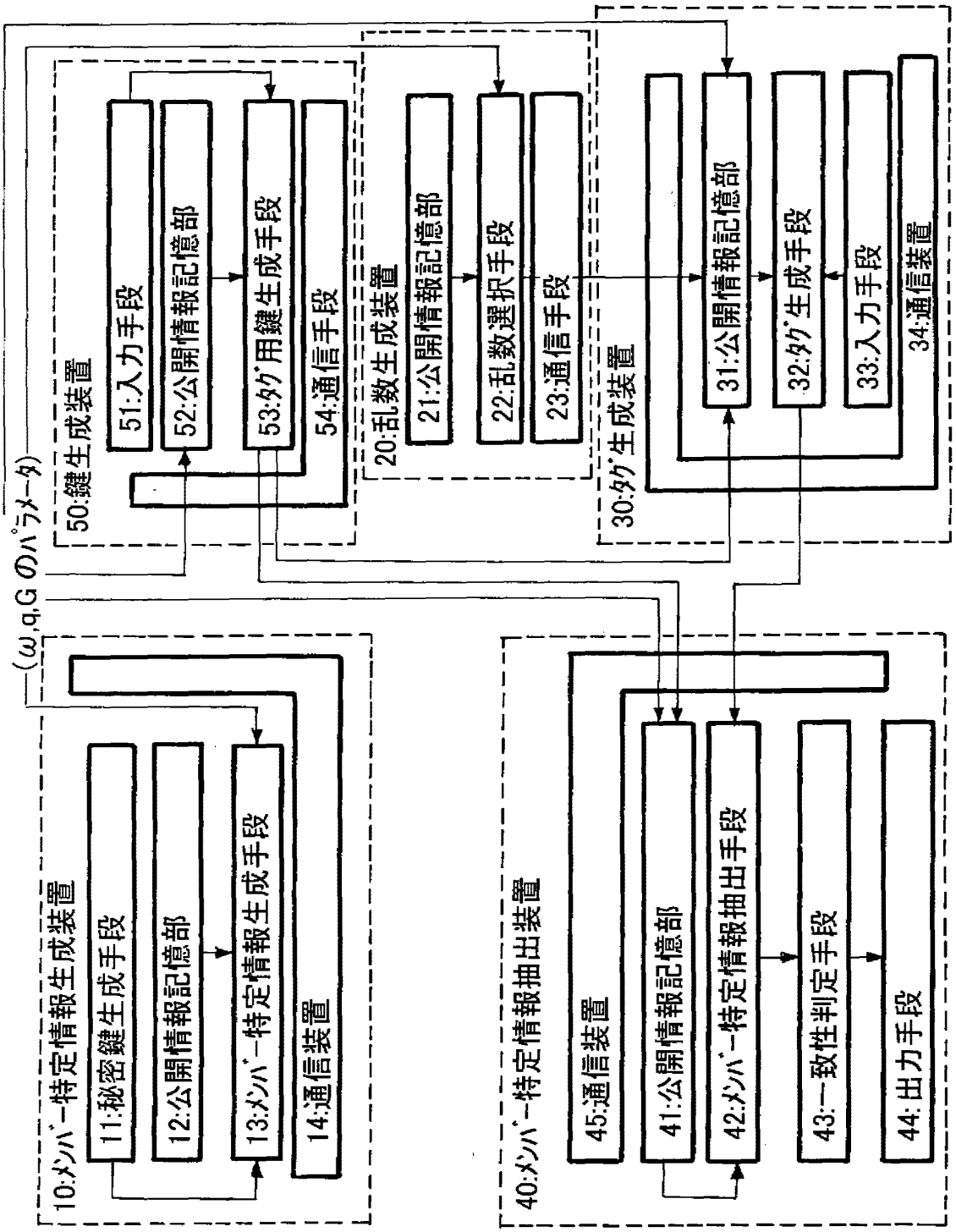
[図4]


1:擬似ランダム関数計算装置
2:擬似ランダム関数用鍵生成手段
3:秘密鍵記憶部
4:公開鍵記憶部
5:入力手段
6:擬似ランダム関数計算手段
7:出力手段

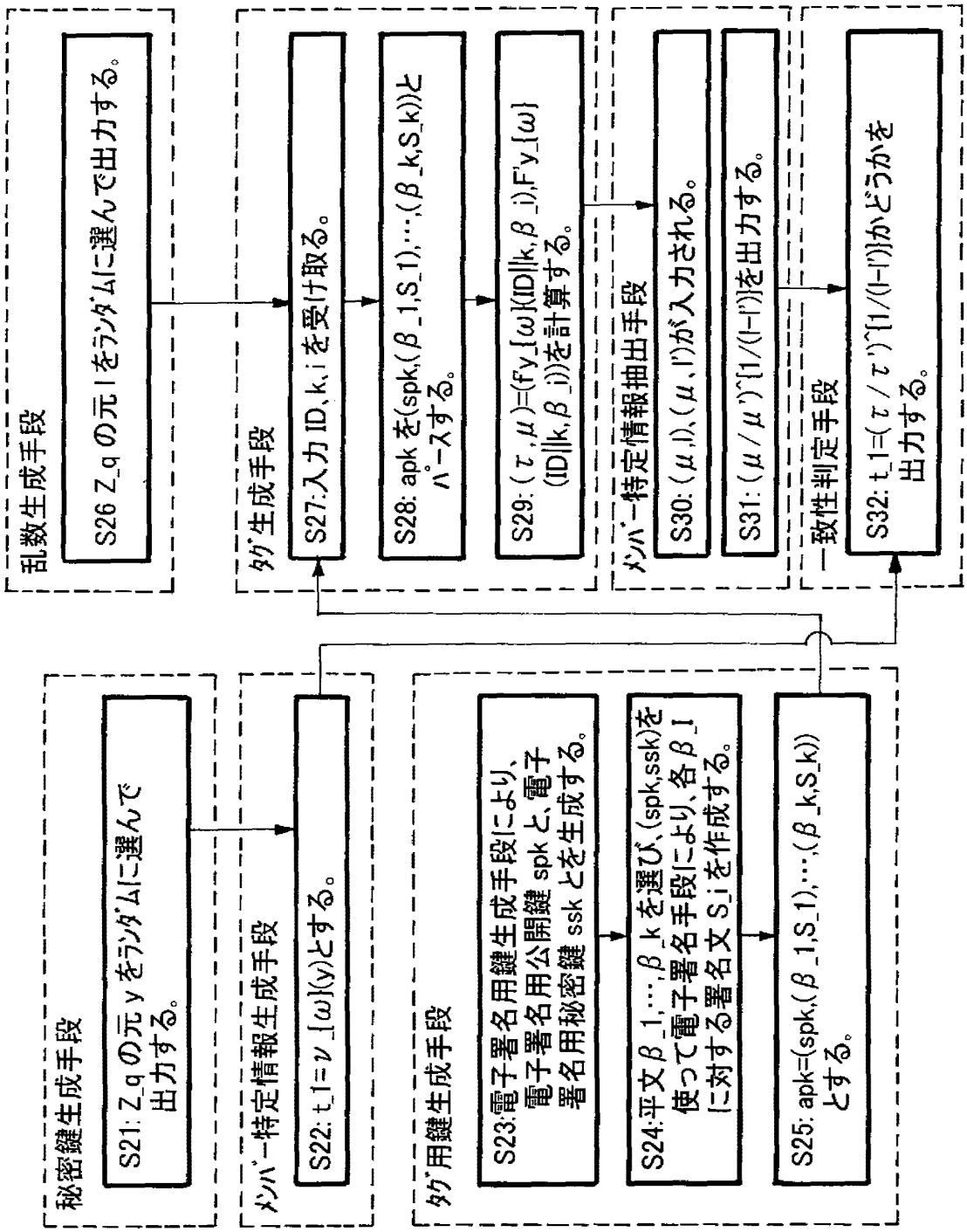
[図5]


擬似ランダム関数用鍵生成手段
S11:Z_q の元 y をランダムに選ぶ。
S12:y を秘密鍵記憶部に書き込む。
擬似ランダム関数計算手段
S13:ビット列 X と G の元 i を入力として受け取る。
S14:{g_{X},h_{X})=H_{G^2}(X)を計算る。
S15:f'y_{ω}(i,X)=g_{X}^{ih_{X}^{1/(y+i)}}を計算する。
S16: f'y_{ω}(i,X)を出力する。

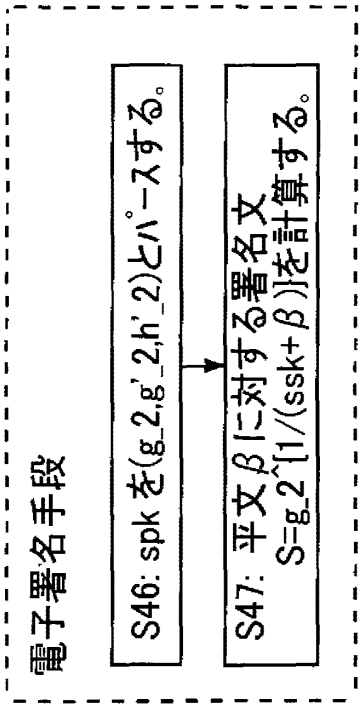
[図6]

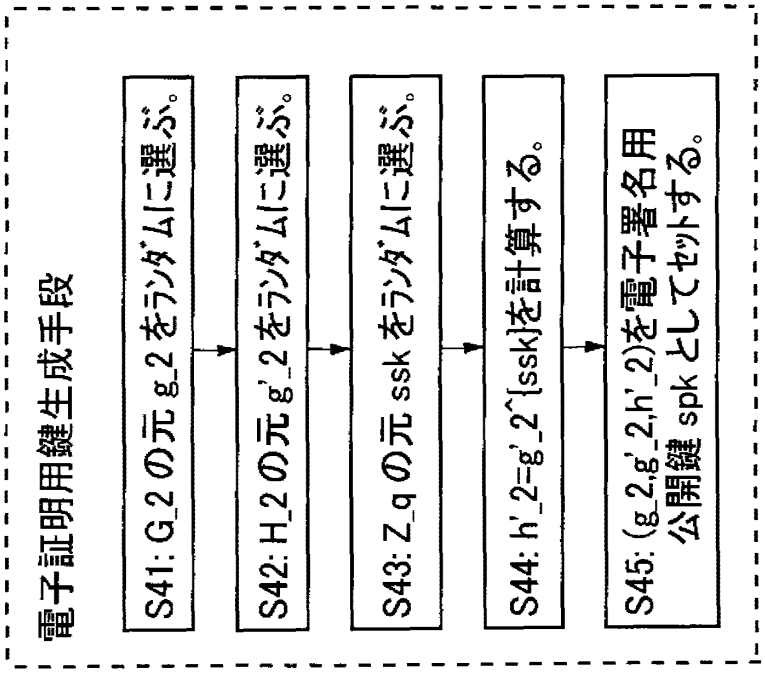

(ω,q,G のパラメータ)
50:鍵生成装置
51:入力手段
52:公開情報記憶部
53:ﾀﾞﾐｰ用鍵生成手段
54:通信手段
20:乱数生成装置
21:公開情報記憶部
22:乱数選択手段
23:通信手段
30:ﾀﾞﾐｰ生成装置
31:公開情報記憶部
32:ﾀﾞﾐｰ生成手段
33:入力手段
34:通信装置
10:メンバー特定情報生成装置
11:秘密鍵生成手段
12:公開情報記憶部
13:メンバー特定情報生成手段
14:通信装置
40:メンバー特定情報抽出装置
45:通信装置
41:公開情報記憶部
42:メンバー特定情報抽出手段
43:一致性判定手段
44:出力手段

[図7]

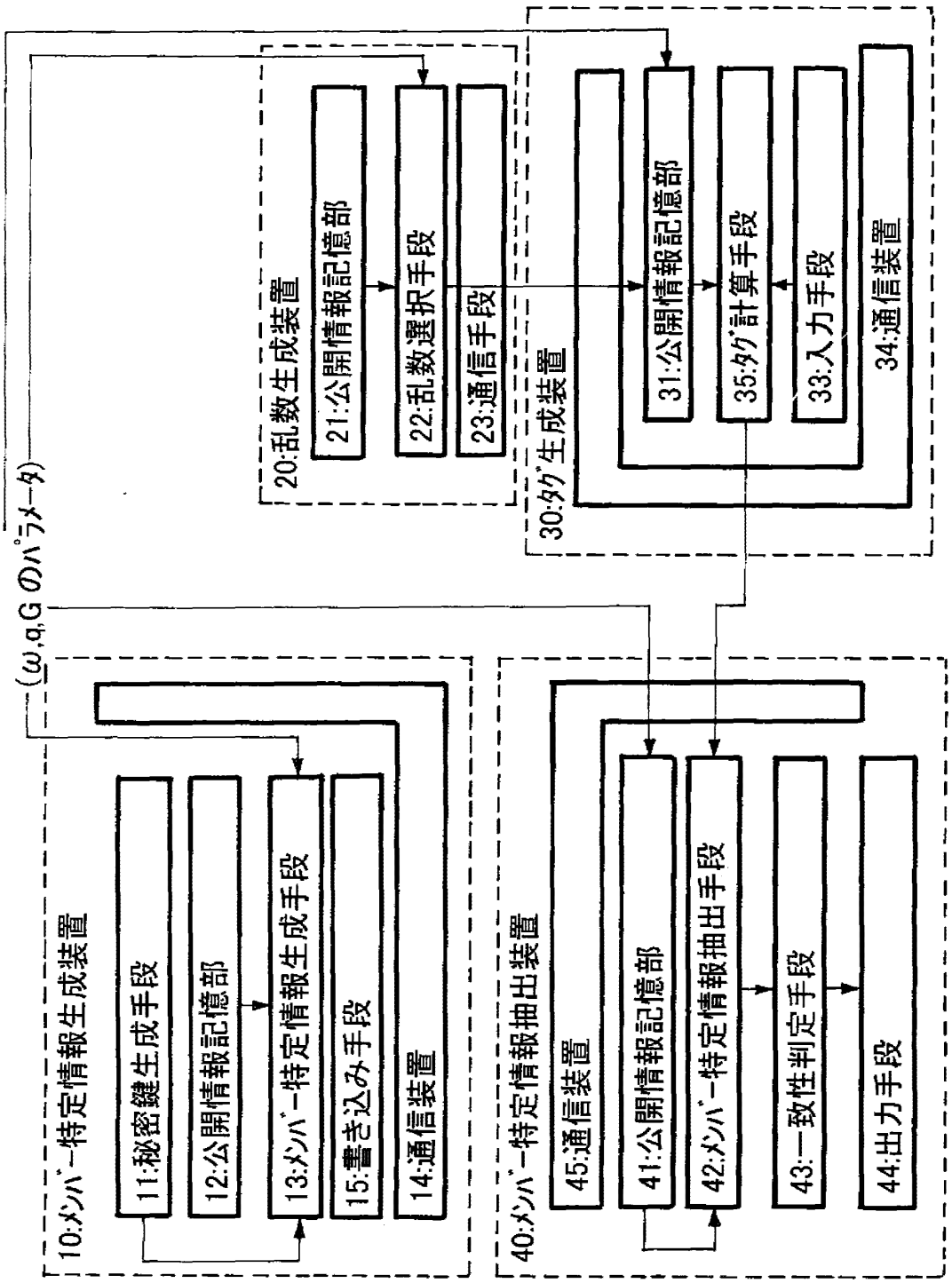

乱数生成手段
S26 Z_q の元 l をランダムに選んで出力する。
タグ生成手段
S27:入力 ID, k, i を受け取る。
S28: apk を(spk,(β_1,S_1),…,(β_k,S_k))とパースする。
S29: (τ,μ)=(f'y_{ω}(ID||k,β_i),F'y_{ω}(ID||k,β_i))を計算する。
メンバー特定情報抽出手段
S30: (μ,l),(μ,l')が入力される。
S31: (μ/μ')^{1/(l-l')}を出力する。
一致性判定手段
S32: t_1=(τ/τ')^{1/(l-l')}かどうかを出力する。
秘密鍵生成手段
S21: Z_q の元 y をランダムに選んで出力する。
メンバー特定情報生成手段
S22: t_1=ν_{ω}(y)とする。
タグ用鍵生成手段
S23:電子署名用鍵生成手段により、電子署名用公開鍵 spk と、電子署名用秘密鍵 ssk とを生成する。
S24:平文 β_1,…,β_k を選び、(spk,ssk)を使って電子署名手段により、各 β_i に対する署名文 S_i を作成する。
S25: apk=(spk,(β_1,S_1),…,(β_k,S_k))とする。

[図8]


電子署名手段
S46: spk を(g_2,g'_2,h'_2)とﾊﾟｰｽする。
S47: 平文βに対する署名文
S=g_2^{1/(ssk+β)}を計算する。

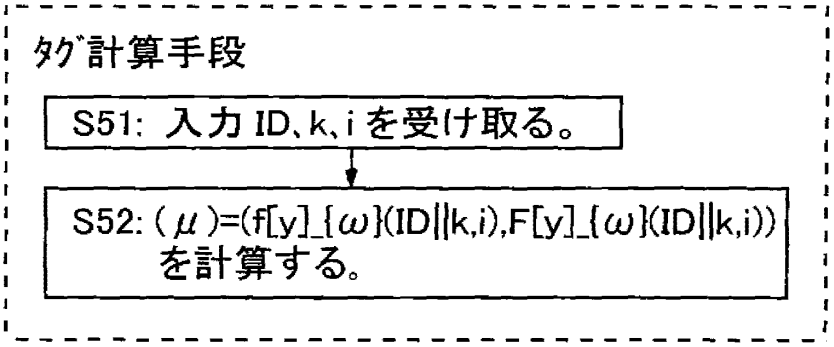

電子証明用鍵生成手段
S41: G_2 の元 g_2 をﾗﾝﾀﾞﾑに選ぶ。
S42: H_2 の元 g'_2 をﾗﾝﾀﾞﾑに選ぶ。
S43: Z_q の元 ssk をﾗﾝﾀﾞﾑに選ぶ。
S44: h'_2=g'_2^{ssk}を計算する。
S45: (g_2,g'_2,h'_2)を電子署名用
公開鍵 spk としてｾｯﾄする。

[図9]

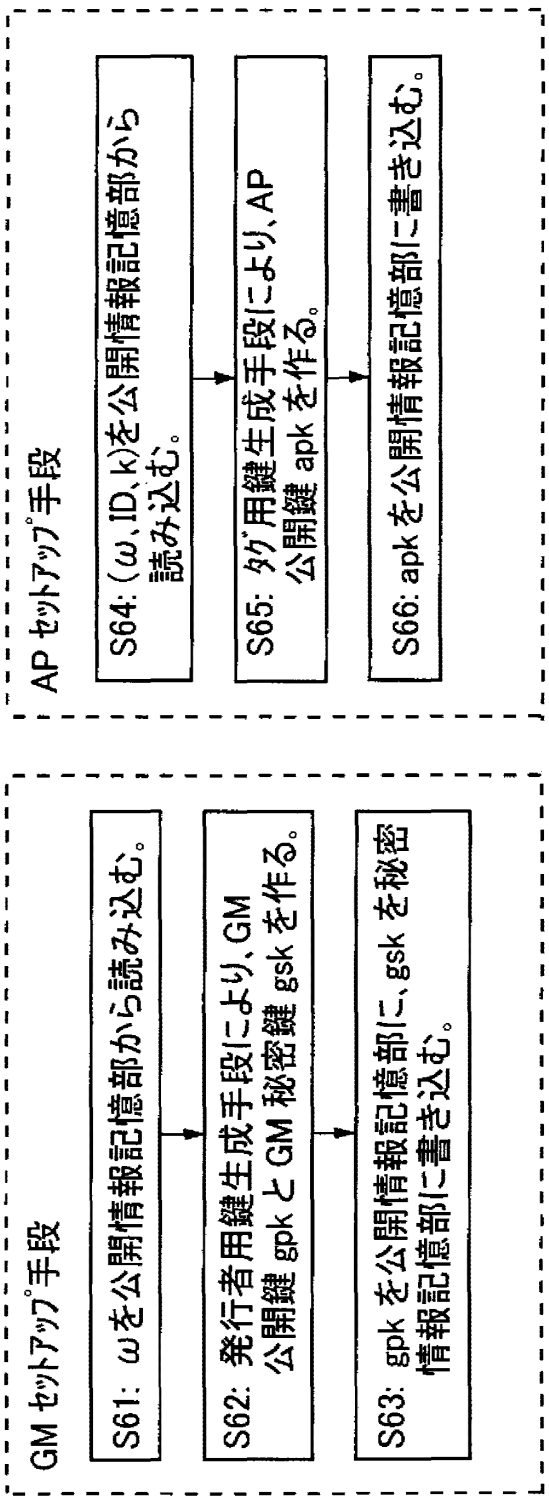


WAVELYNX EX. 1003, Pg. 682
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

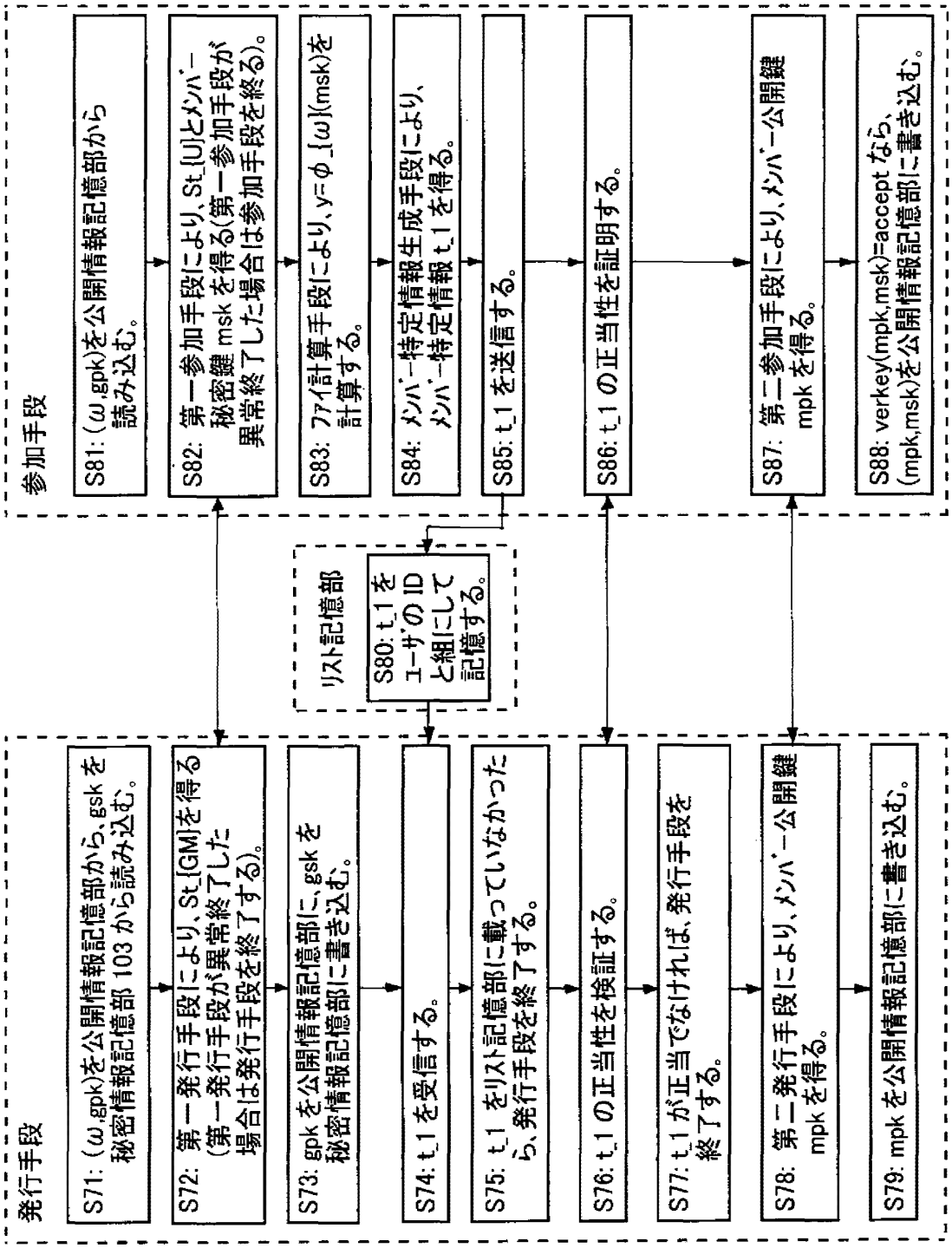
[図10]



[図11]

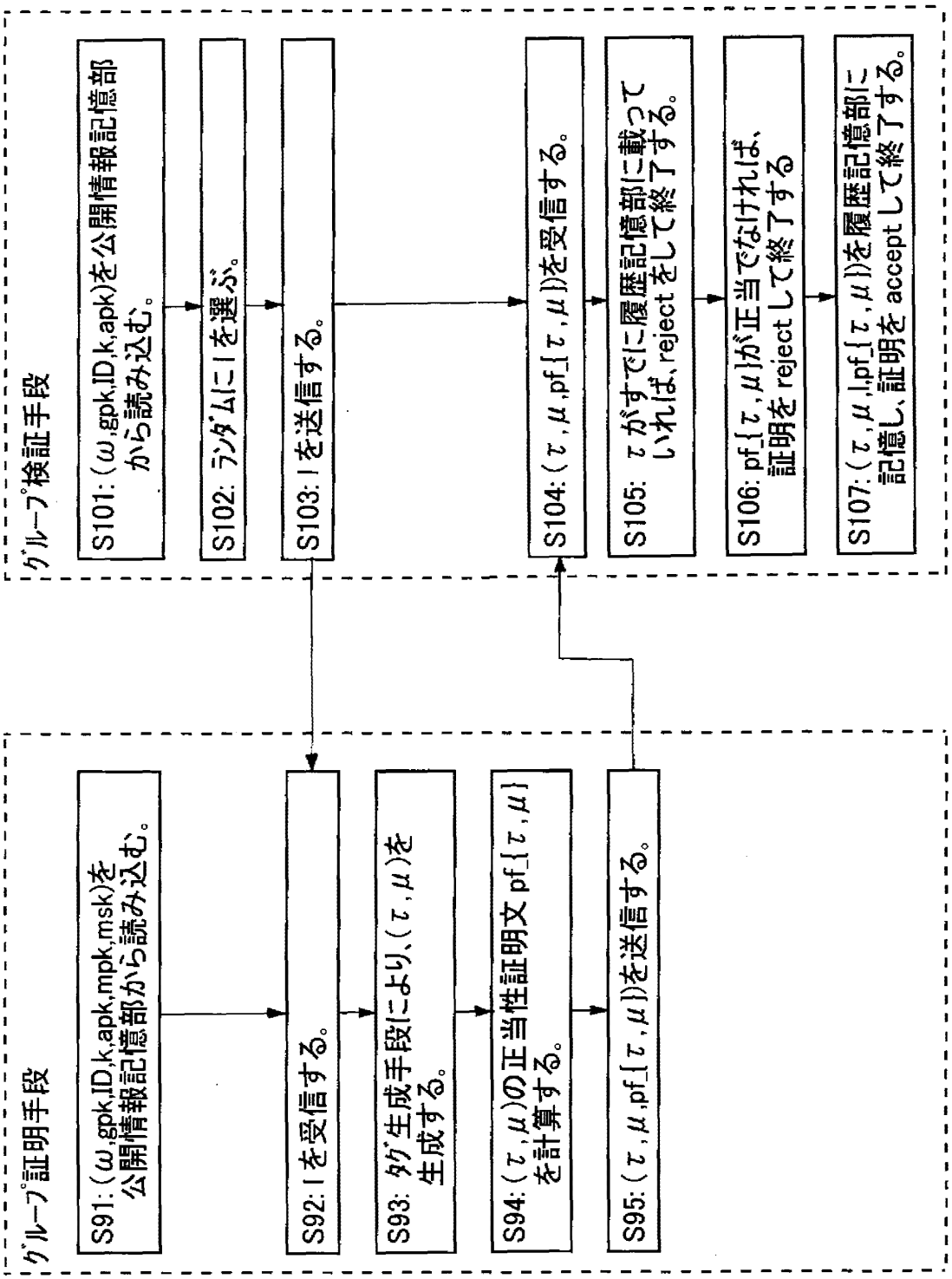

GM セットアップ手段
S61: ωを公開情報記憶部から読み込む。
S62: 発行者用鍵生成手段により、GM 公開鍵 gpk と GM 秘密鍵 gsk を作る。
S63: gpk を公開情報記憶部に、gsk を秘密情報記憶部に書き込む。
AP セットアップ手段
S64: (ω, ID, k)を公開情報記憶部から読み込む。
S65: ﾀｸﾞ用鍵生成手段により、AP 公開鍵 apk を作る。
S66: apk を公開情報記憶部に書き込む。

[図12]

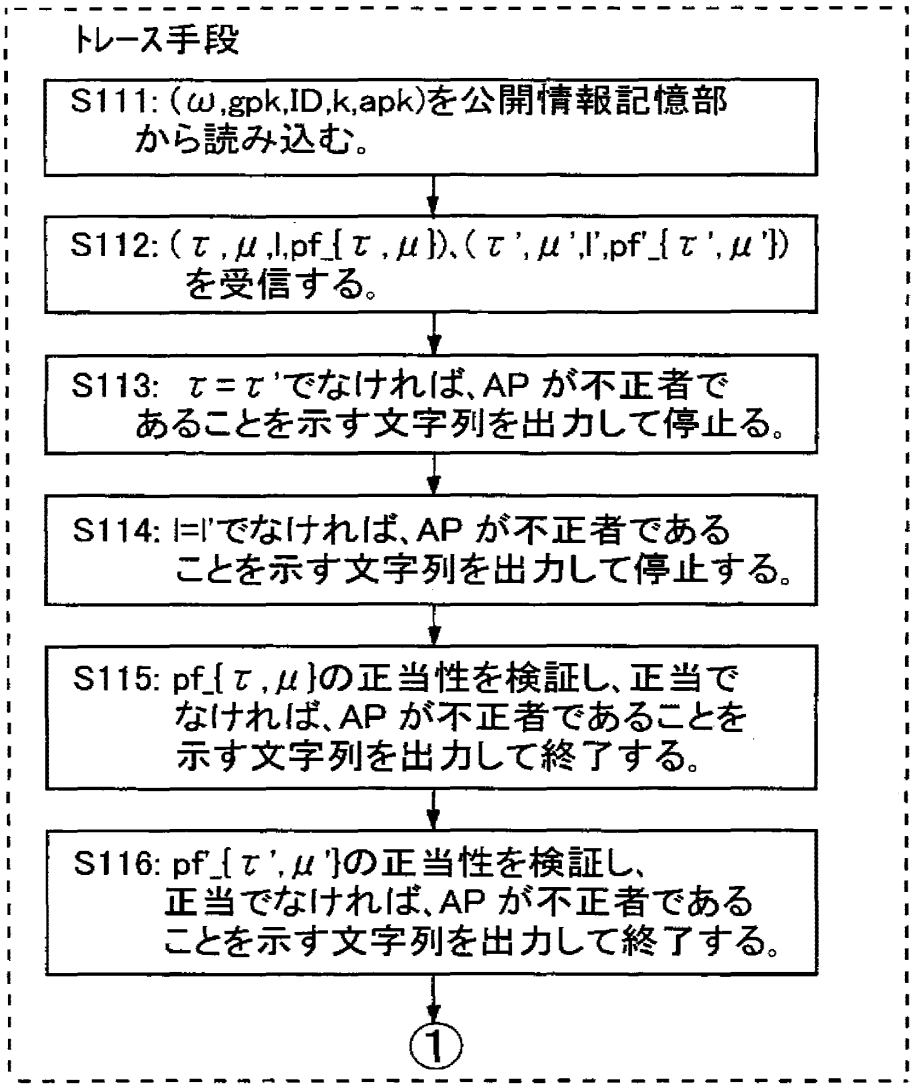


WAVELYNX EX. 1003, Pg. 685
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

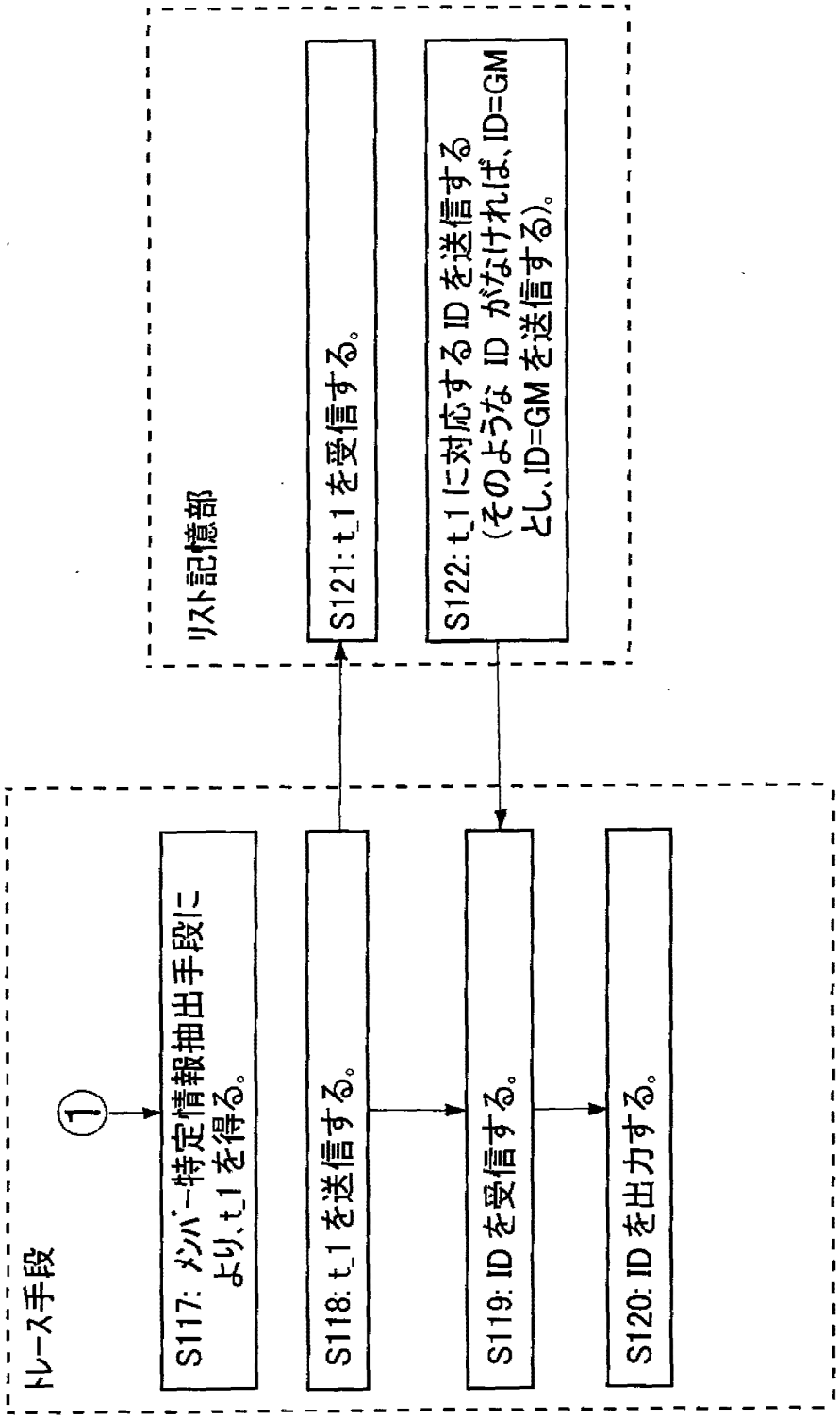
[図13]


グループ検証手段
S101: (ω,gpk,ID,k,apk)を公開情報記憶部から読み込む。
S102: ランダムにlを選ぶ。
S103: lを送信する。
S104: (τ,μ,pf_{τ,μ})を受信する。
S105: τがすでに履歴記憶部に載っていれば、rejectをして終了する。
S106: pf_{τ,μ}が正当でなければ、証明をrejectして終了する
S107: (τ,μ,pf_{τ,μ})を履歴記憶部に記憶し、証明をacceptして終了する。
グループ証明手段
S91: (ω,gpk,ID,k,apk,mpk,msk)を公開情報記憶部から読み込む。
S92: lを受信する。
S93: 乱τ生成手段により、(τ,μ)を生成する。
S94: (τ,μ)の正当性証明文pf_{τ,μ}を計算する。
S95: (τ,μ,pf_{τ,μ})を送信する。

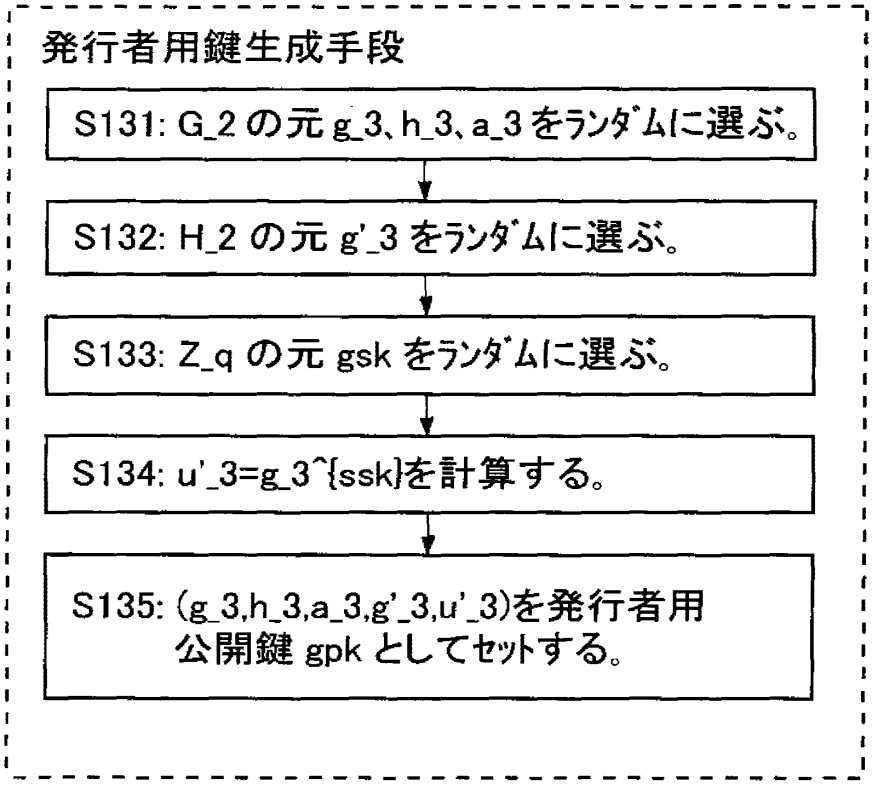
[図14]


トレース手段
S111: (ω,gpk,ID,k,apk)を公開情報記憶部から読み込む。
S112: (τ,μ,l,pf_{τ,μ}),(τ',μ',l',pf'_{τ',μ'})を受信する。
S113: τ=τ'でなければ、APが不正者であることを示す文字列を出力して停止る。
S114: l=l'でなければ、APが不正者であることを示す文字列を出力して停止する。
S115: pf_{τ,μ}の正当性を検証し、正当でなければ、APが不正者であることを示す文字列を出力して終了する。
S116: pf'_{τ',μ'}の正当性を検証し、正当でなければ、APが不正者であることを示す文字列を出力して終了する。
①

[図15]

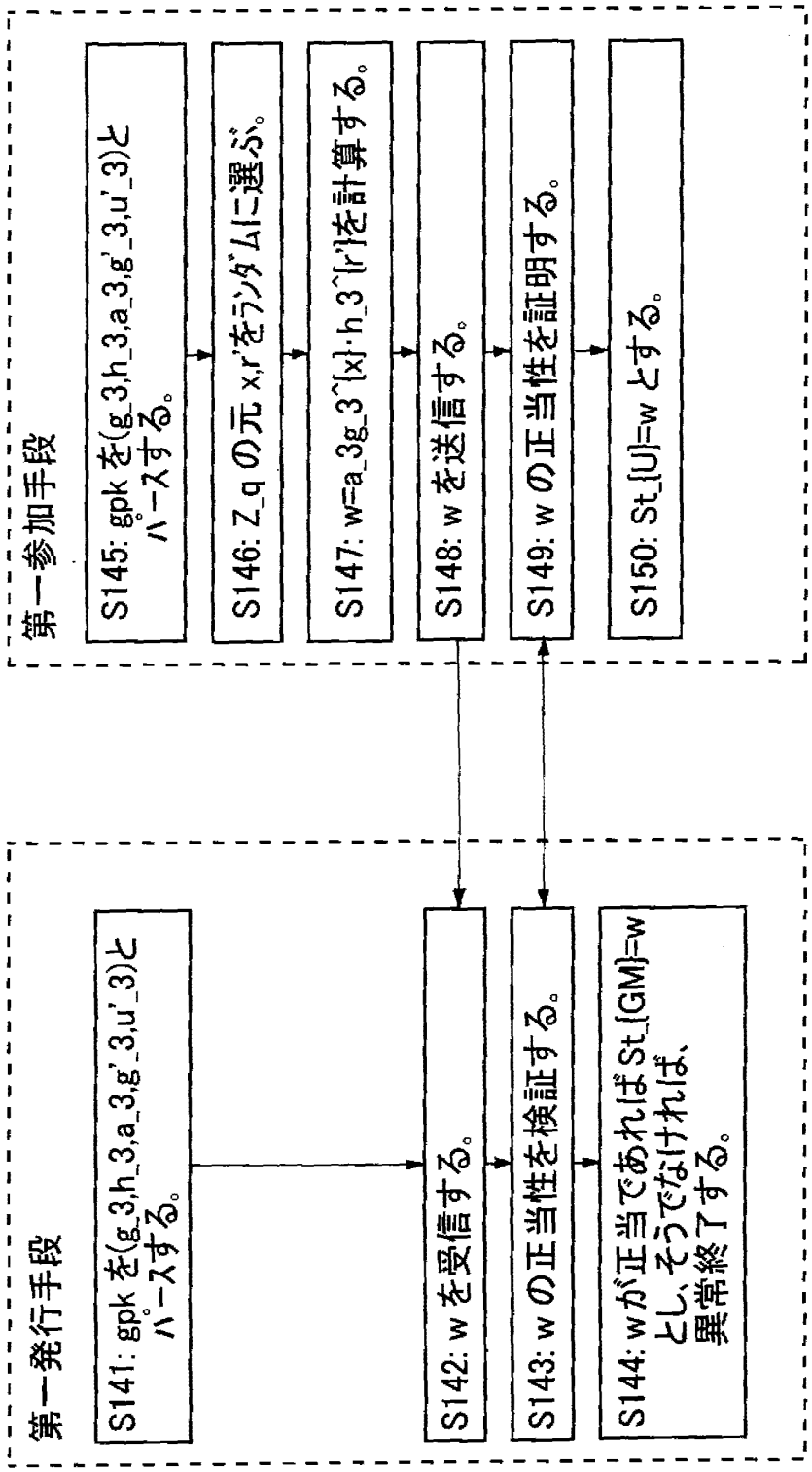

トレース手段
①
S117: メンバー特定情報抽出手段により、t_1を得る。
S118: t_1を送信する。
S119: IDを受信する。
S120: IDを出力する。
リスト記憶部
S121: t_1を受信する。
S122: t_1に対応するIDを送信する（そのようなIDがなければ、ID=GMとし、ID=GMを送信する）。

[図16]

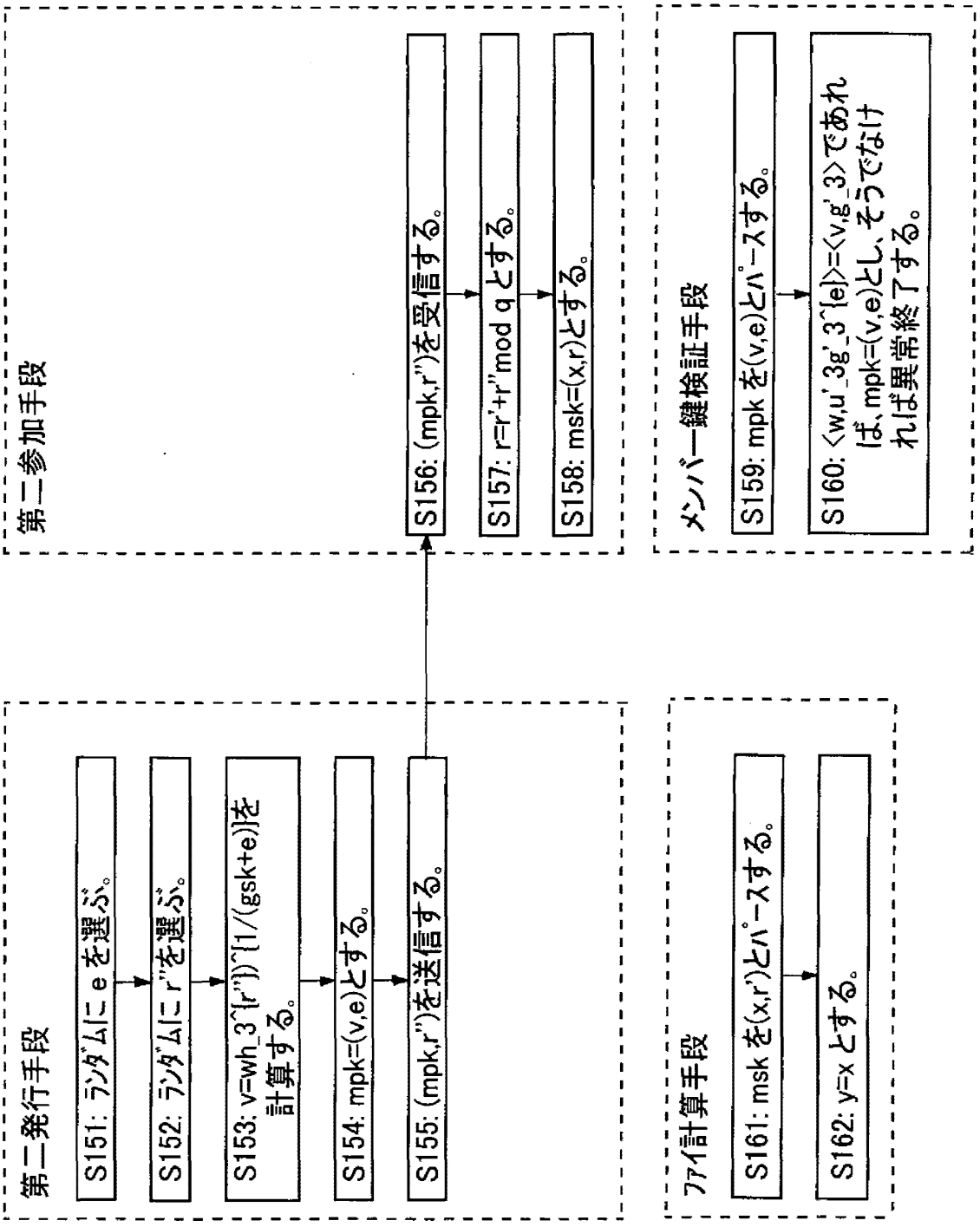

発行者用鍵生成手段
S131: G_2 の元 g_3、h_3、a_3 をランダムに選ぶ。
S132: H_2 の元 g'_3 をランダムに選ぶ。
S133: Z_q の元 gsk をランダムに選ぶ。
S134: u'_3=g_3^{ssk}を計算する。
S135: (g_3,h_3,a_3,g'_3,u'_3)を発行者用公開鍵 gpk としてセットする。

WO 2006/126668 PCT/JP2006/310534

[図17]

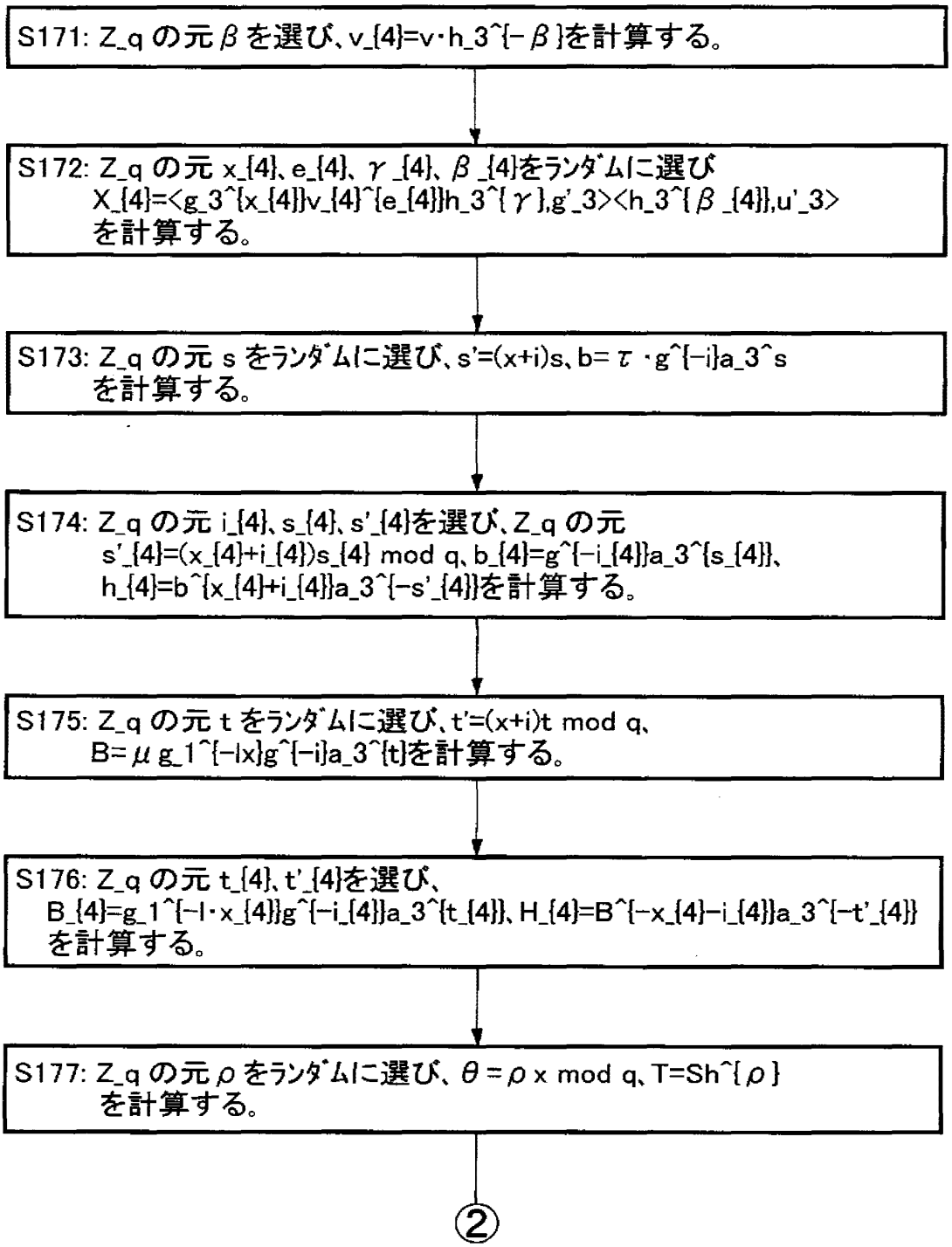


[図18]

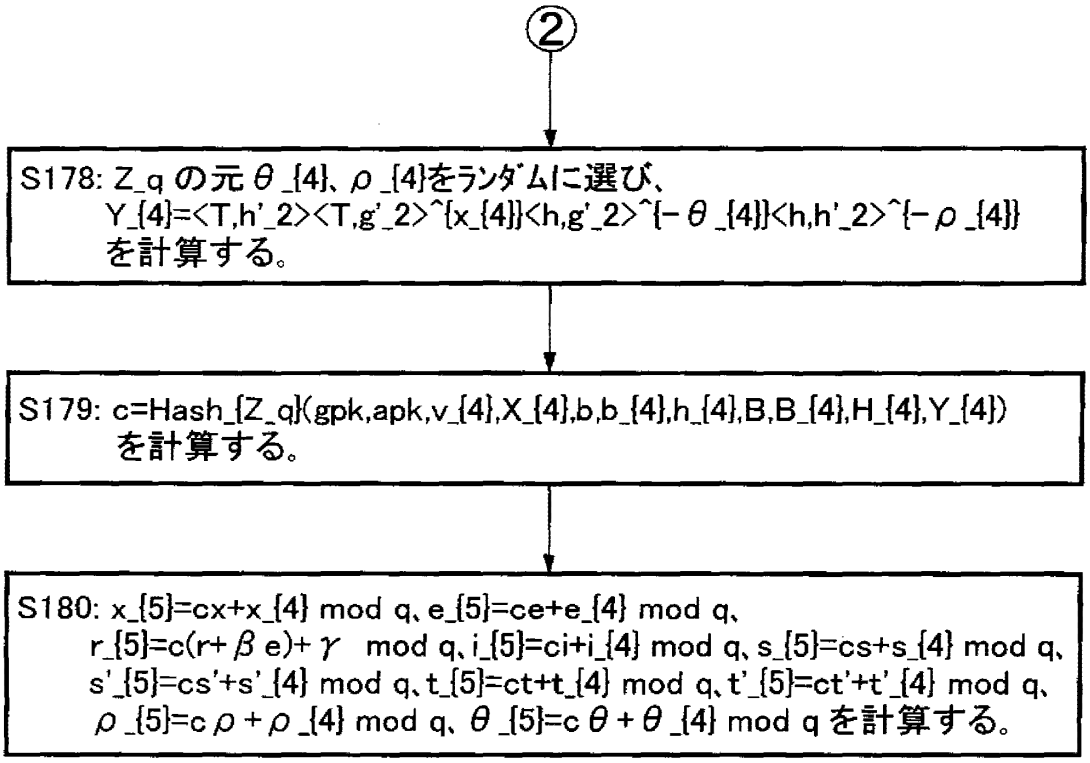


WAVELYNX EX. 1003, Pg. 691
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

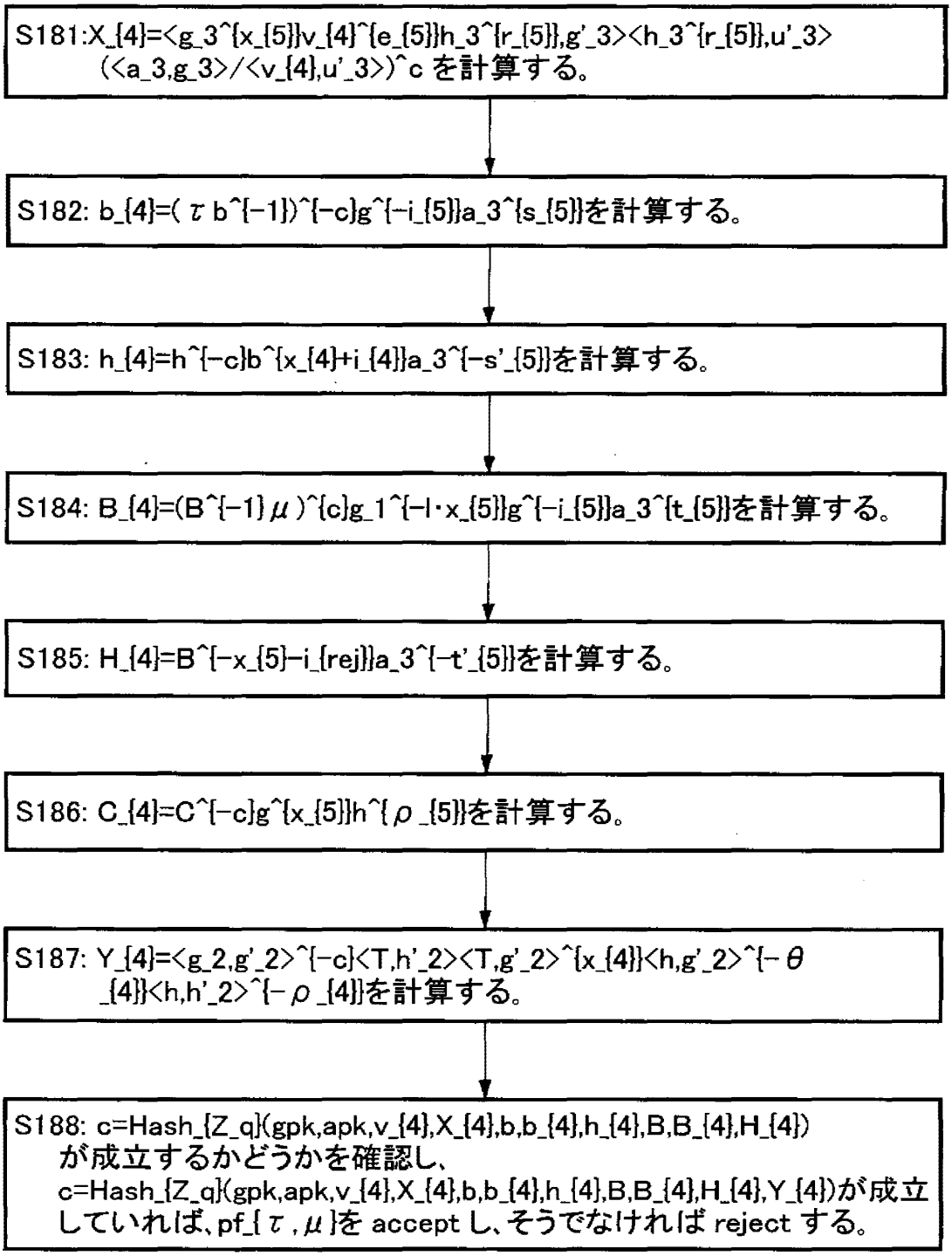
[図19]


S171: Z_q の元 β を選び、v_{4}=v・h_3^{−β}を計算する。
S172: Z_q の元 x_{4}、e_{4}、γ_{4}、β_{4}をランダムに選び
X_{4}=<g_3^{x_{4}}v_{4}^{e_{4}}h_3^{γ},g'_3><h_3^{β_{4}},u'_3>
を計算する。
S173: Z_q の元 s をランダムに選び、s'=(x+i)s、b=τ・g^{−i}a_3^s
を計算する。
S174: Z_q の元 i_{4}、s_{4}、s'_{4}を選び、Z_q の元
s'_{4}=(x_{4}+i_{4})s_{4} mod q、b_{4}=g^{−i_{4}}a_3^{s_{4}}、
h_{4}=b^{x_{4}+i_{4}}a_3^{−s'_{4}}を計算する。
S175: Z_q の元 t をランダムに選び、t'=(x+i)t mod q、
B=μg_1^{−lx}g^{−i}a_3^{t}を計算する。
S176: Z_q の元 t_{4}、t'_{4}を選び、
B_{4}=g_1^{−l・x_{4}}g^{−i_{4}}a_3^{t_{4}}、H_{4}=B^{−x_{4}−i_{4}}a_3^{−t'_{4}}
を計算する。
S177: Z_q の元 ρ をランダムに選び、θ=ρ x mod q、T=Sh^{ρ}
を計算する。
②

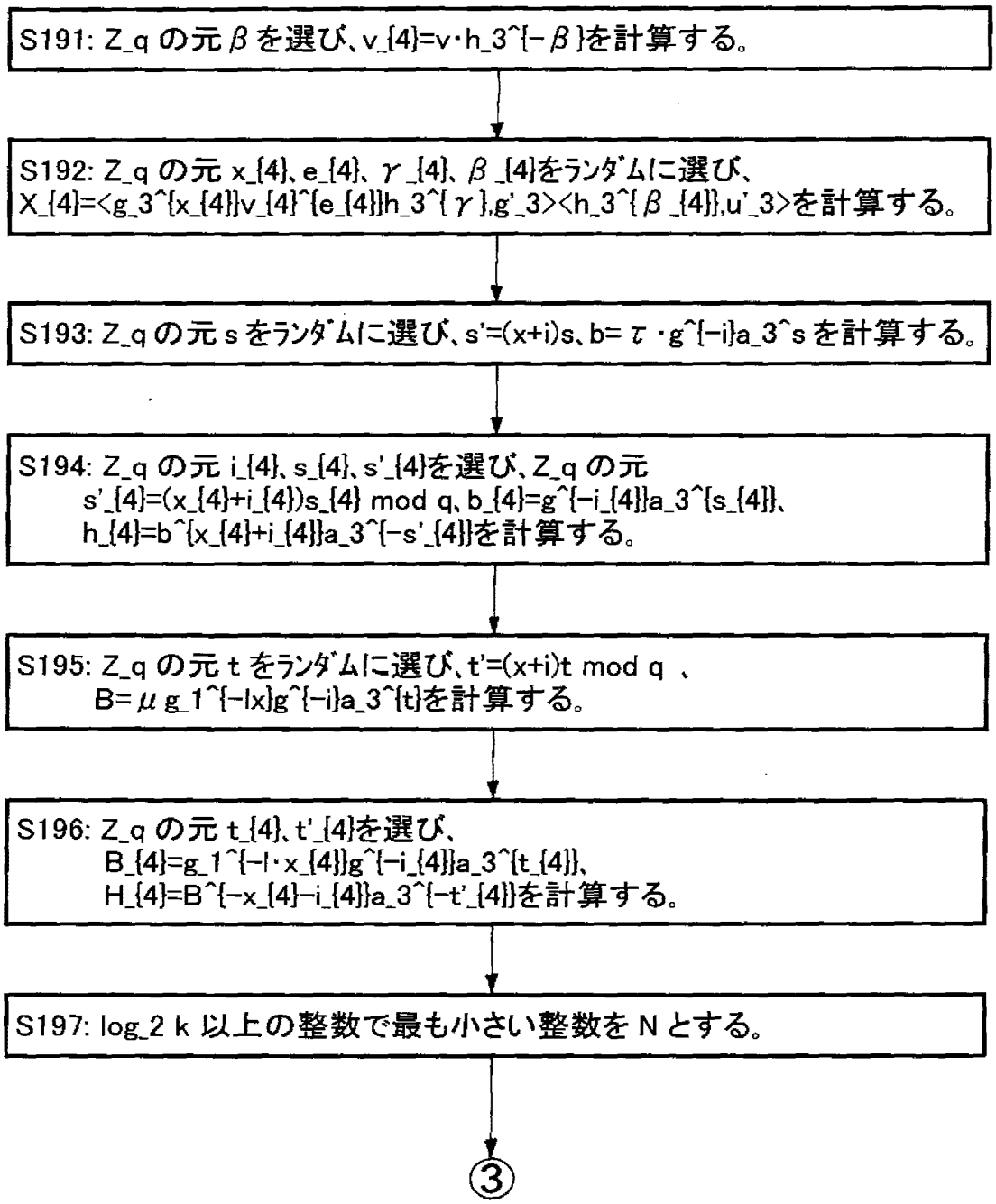
[図20]


②
S178: Z_q の元 θ _{4}、ρ _{4}をランダムに選び、
Y_{4}=<T,h'_2><T,g'_2>^{x_{4}}<h,g'_2>^{−θ _{4}}<h,h'_2>^{−ρ _{4}}
を計算する。
S179: c=Hash_{Z_q}(gpk,apk,v_{4},X_{4},b,b_{4},h_{4},B,B_{4},H_{4},Y_{4})
を計算する。
S180: x_{5}=cx+x_{4} mod q、e_{5}=ce+e_{4} mod q、
r_{5}=c(r+β e)+γ mod q、i_{5}=ci+i_{4} mod q、s_{5}=cs+s_{4} mod q、
s'_{5}=cs'+s'_{4} mod q、t_{5}=ct+t_{4} mod q、t'_{5}=ct'+t'_{4} mod q、
ρ _{5}=c ρ +ρ _{4} mod q、θ _{5}=c θ +θ _{4} mod q を計算する。

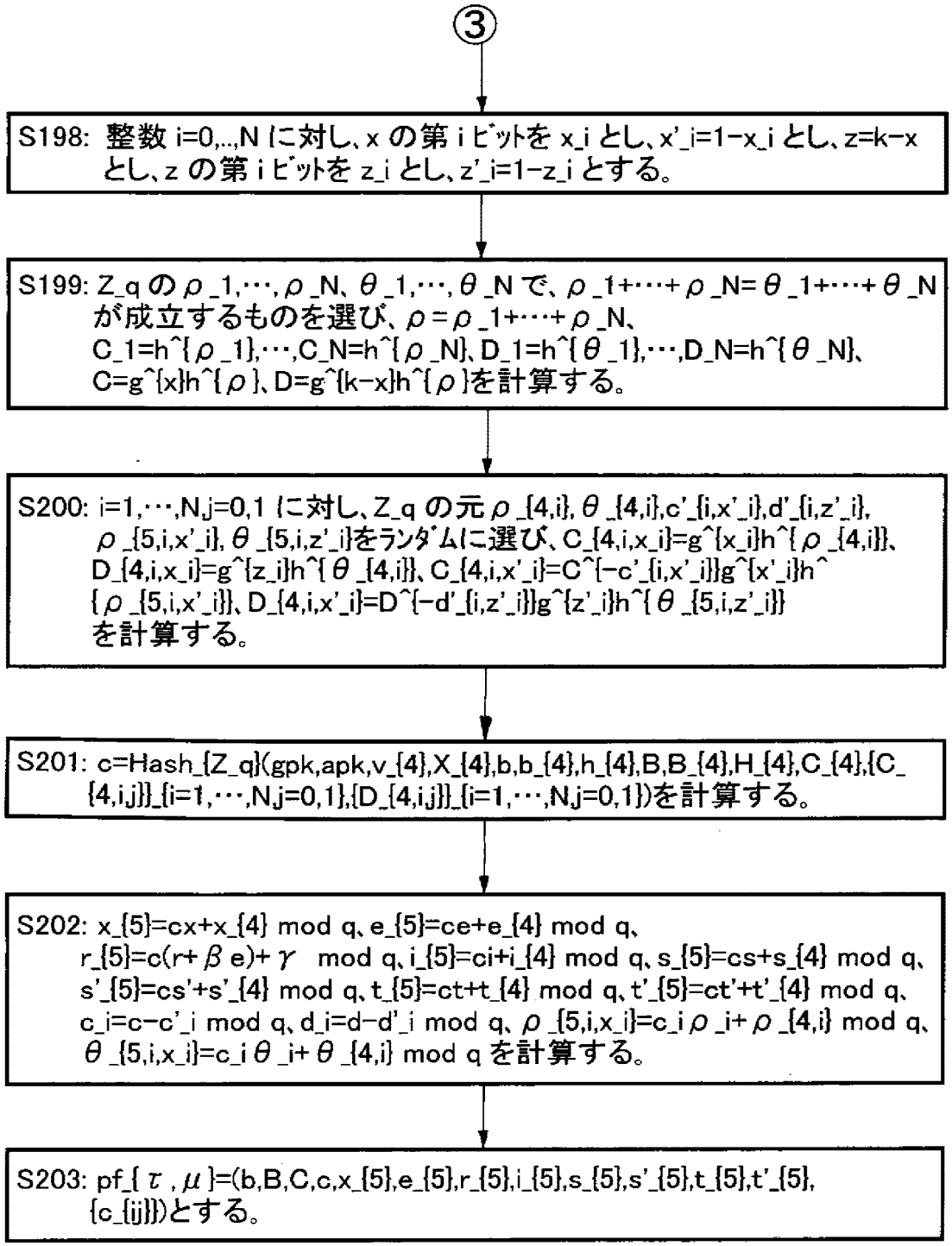
[図21]


S181:X_{4}=<g_3^{x_{5}}v_{4}^{e_{5}}h_3^{r_{5}},g'_3><h_3^{r_{5}},u'_3>
(<a_3,g_3>/<v_{4},u'_3>)^c を計算する。
S182: b_{4}=(τ b^{-1})^{-c}g^{-i_{5}}a_3^{s_{5}}を計算する。
S183: h_{4}=h^{-c}b^{x_{4}+i_{4}}a_3^{-s'_{5}}を計算する。
S184: B_{4}=(B^{-1} μ)^{c}g_1^{-l・x_{5}}g^{-i_{5}}a_3^{t_{5}}を計算する。
S185: H_{4}=B^{-x_{5}-i_{rej}}a_3^{-t'_{5}}を計算する。
S186: C_{4}=C^{-c}g^{x_{5}}h^{ ρ _{5}}を計算する。
S187: Y_{4}=<g_2,g'_2>^{-c}<T,h'_2><T,g'_2>^{x_{4}}<h,g'_2>^{- θ
_{4}}<h,h'_2>^{- ρ _{4}}を計算する。
S188: c=Hash_{Z_q}(gpk,apk,v_{4},X_{4},b,b_{4},h_{4},B,B_{4},H_{4})
が成立するかどうかを確認し、
c=Hash_{Z_q}(gpk,apk,v_{4},X_{4},b,b_{4},h_{4},B,B_{4},H_{4},Y_{4})が成立
していれば、pf_{ τ , μ }を accept し、そうでなければ reject する。

[図22]


S191: Z_q の元 β を選び、v_{4}=v・h_3^{−β}を計算する。
S192: Z_q の元 x_{4}、e_{4}、γ_{4}、β_{4}をランダムに選び、
X_{4}=<g_3^{x_{4}}v_{4}^{e_{4}}h_3^{γ},g'_3><h_3^{β_{4}},u'_3>を計算する。
S193: Z_q の元 s をランダムに選び、s'=(x+i)s、b= τ ・g^{−i}a_3^s を計算する。
S194: Z_q の元 i_{4}、s_{4}、s'_{4}を選び、Z_q の元
s'_{4}=(x_{4}+i_{4})s_{4} mod q、b_{4}=g^{−i_{4}}a_3^{s_{4}}、
h_{4}=b^{x_{4}+i_{4}}a_3^{−s'_{4}}を計算する。
S195: Z_q の元 t をランダムに選び、t'=(x+i)t mod q 、
B=μ g_1^{−lx}g^{−i}a_3^{t}を計算する。
S196: Z_q の元 t_{4}、t'_{4}を選び、
B_{4}=g_1^{−l・x_{4}}g^{−i_{4}}a_3^{t_{4}}、
H_{4}=B^{−x_{4}−i_{4}}a_3^{−t'_{4}}を計算する。
S197: log_2 k 以上の整数で最も小さい整数を N とする。
③

[図23]


③
S198: 整数 i=0,..,N に対し、x の第 i ビットを x_i とし、x'_i=1−x_i とし、z=k−x とし、z の第 i ビットを z_i とし、z'_i=1−z_i とする。
S199: Z_q の ρ_1,…,ρ_N、θ_1,…,θ_N で、ρ_1+…+ρ_N=θ_1+…+θ_N が成立するものを選び、ρ=ρ_1+…+ρ_N、C_1=h^{ρ_1},…,C_N=h^{ρ_N}、D_1=h^{θ_1},…,D_N=h^{θ_N}、C=g^{x}h^{ρ}、D=g^{k−x}h^{ρ}を計算する。
S200: i=1,…,N,j=0,1 に対し、Z_q の元 ρ_{4,i}, θ_{4,i},c'_{i,x'_i},d'_{i,z'_i}, ρ_{5,i,x'_i}, θ_{5,i,z'_i}をランダムに選び、C_{4,i,x_i}=g^{x_i}h^{ρ_{4,i}}、D_{4,i,x_i}=g^{z_i}h^{θ_{4,i}}、C_{4,i,x'_i}=C^{−c'_{i,x'_i}}g^{x'_i}h^{ρ_{5,i,x'_i}}、D_{4,i,x'_i}=D^{−d'_{i,z'_i}}g^{z'_i}h^{θ_{5,i,z'_i}} を計算する。
S201: c=Hash_{Z_q}(gpk,apk,v_{4},X_{4},b,b_{4},h_{4},B,B_{4},H_{4},C_{4},{C_{4,i,j}}_{i=1,…,N,j=0,1},{D_{4,i,j}}_{i=1,…,N,j=0,1})を計算する。
S202: x_{5}=cx+x_{4} mod q、e_{5}=ce+e_{4} mod q、r_{5}=c(r+βe)+γ mod q、i_{5}=ci+i_{4} mod q、s_{5}=cs+s_{4} mod q、s'_{5}=cs'+s'_{4} mod q、t_{5}=ct+t_{4} mod q、t'_{5}=ct'+t'_{4} mod q、c_i=c−c'_i mod q、d_i=d−d'_i mod q、ρ_{5,i,x_i}=c_iρ_i+ρ_{4,i} mod q、θ_{5,i,x_i}=c_iθ_i+θ_{4,i} mod q を計算する。
S203: pf_{τ,μ}=(b,B,C,c,x_{5},e_{5},r_{5},i_{5},s_{5},s'_{5},t_{5},t'_{5},{c_{ij}})とする。

[図24]


S211:X_{4}=<g_3^{x_{5}}v_{4}^{e_{5}}h_3^{r_{5}},g'_3><h_3^{r_{5}},u'_3>
(<a_3,g_3>/<v_{4},u'_3>)^c を計算する。
S212: b_{4}=(τ b^{-1})^{-c}g^{-i_{5}}a_3^{s_{5}}を計算する。
S213: h_{4}=h^{-c}b^{x_{4}+i_{4}}a_3^{-s'_{5}}を計算する。
S214: B_{4}=(B^{-1} μ)^{c}g_1^{-I・x_{5}}g^{-i_{5}}a_3^{t_{5}}を計算する。
S215: H_{4}=B^{-x_{5}-i_{rej}}a_3^{-t'_{5}}を計算する。
S216: C_{4}=C^{-c}g^{x_{5}} h^{ ρ _{5}}を計算する。
S217: i=1,..,N、j=0,1 に対し、C_{4,i,j}=C_{-c_{ij}}g^{j}h^{ ρ _{5,ij}}、
D_{4,i,j}=D_{-c_{ij}}g^{j}h^{ θ _{5,ij}}を計算する。
S218:c=Hash_{Z_q}(gpk,apk,v_{4},X_{4},b,b_{4},h_{4},B,B_{4},H_{4},{C_{ij}}_{i=1,
…,N,j=0,1},{D_{ij}}_{i=1,…,N,j=0,1})と、c=c_1+c'_1=…=c_N+c'_N mod
q が成立するかどうかを確認し、成立していれば、pf_{ τ , μ }を
accept し、そうでなければ reject する。

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2006/310534

A. CLASSIFICATION OF SUBJECT MATTER
G09C1/00(2006.01)i, *H04L9/32*(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G09C1/00, H04L9/32

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2006
Kokai Jitsuyo Shinan Koho	1971-2006	Toroku Jitsuyo Shinan Koho	1994-2006

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Yevgeniy Dodis and Aleksandr Yampolskiy, A Verifiable Random Function With Short Proofs and Keys, [online] Cryptology ePrint archive, 07 March, 2005 (07.03.05), [retrieved on 08 August, 2006 (08.08.06)]. Retrieved from the Internet: <URL:http://eprint.iacr.org/2004/310.pdf>.	1-28
A	JP 2004-228958 A (NEC Corp.), 12 August, 2004 (12.08.04), Full text (Family: none)	4-14,18-28

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
08 August, 2006 (08.08.06)	15 August, 2006 (15.08.06)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

Form PCT/ISA/210 (second sheet) (April 2005)

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００６／３１０５３４

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00(2006.01)i, H04L9/32(2006.01)i

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00, H04L9/32

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００６年
日本国実用新案登録公報	１９９６－２００６年
日本国登録実用新案公報	１９９４－２００６年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
A	Yevgeniy Dodis and Aleksandr Yampolskiy, A Verifiable Random Function With Short Proofs and Keys, [online] Cryptology ePrint archive, 2005.03.07, [retrieved on 2006.08.08]. Retrieved from the Internet: <URL: http://eprint.iacr.org/2004/310.pdf>.	1-28
A	JP 2004-228958 A（日本電気株式会社）2004.08.12, 全文（ファミリーなし）	4-14,18-28

☐　Ｃ欄の続きにも文献が列挙されている。

☐　パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー

「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの

「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「Ｏ」口頭による開示、使用、展示等に言及する文献

「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「＆」同一パテントファミリー文献

国際調査を完了した日 ０８．０８．２００６	国際調査報告の発送日 １５．０８．２００６
国際調査機関の名称及びあて先 日本国特許庁（ＩＳＡ／ＪＰ） 郵便番号１００－８９１５ 東京都千代田区霞が関三丁目４番３号	特許庁審査官（権限のある職員）　５Ｓ　３３６５ 石川　正二 電話番号　０３－３５８１－１１０１　内線　３５４６

WAVELYNX EX. 1003, Pg. 699
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 September 2007 (13.09.2007)

PCT

(10) International Publication Number
WO 2007/103906 A2

(51) International Patent Classification:
G06Q 99/00 (2006.01)

(21) International Application Number:
PCT/US2007/063361

(22) International Filing Date: 6 March 2007 (06.03.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/368,959 6 March 2006 (06.03.2006) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier applications:
US 11/286,890 (CON)
Filed on 23 November 2005 (23.11.2005)
US 10/854,604 (CIP)
Filed on 26 May 2004 (26.05.2004)
US 10/248,894 (CIP)
Filed on 27 February 2003 (27.02.2003)

(71) Applicant *(for all designated States except US)*: **IMAGINEER SOFTWARE, INC.** [US/US]; 3452 North Hackett Avenue, Milwaukee, WI 53211 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **MALINA, Richard** [US/US]; 7914 Eastfield Circle, Mequon, WI 53097 (US). **COCHRAN, William** [US/US]; 307 West Washington Street, Champaign, IL 61820 (US).

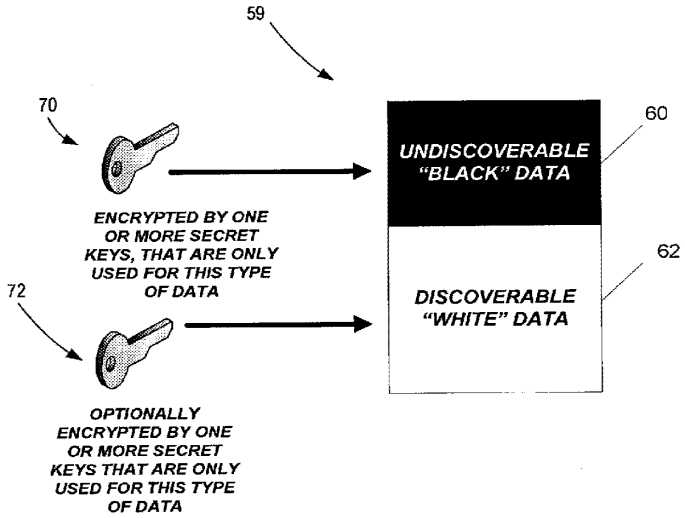
(74) Agent: **STETTNER, Derek, C.**; Michael Best & Friedrich LLP, 100 East Wisconsin Avenue, Suite 3300, Milwaukee, WI 53202-4108 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US (patent), UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) **Title:** SECURE DATA TRANSMISSION USING UNDISCOVERABLE OR BLACK DATA



(57) **Abstract:** Methods and systems of encrypting data. One method includes establishing data that includes discoverable or "white" data and undiscoverable or "black" data. Black data is generally unrecognizable. For example, it may be random data. White data generally has recognizable content or is transmitted in a recognizable format. A first key and a second key are created or established, such that the second key is substantially underivable from the first key. The discoverable or white data is encrypted with the first key and the undiscoverable or black data is encrypted with the second key. In subsequent communications or transactions, at least one of the first key and the second key is mutated.

WO 2007/103906 A2

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SECURE DATA TRANSMISSION USING UNDISCOVERABLE OR BLACK DATA

RELATED APPLICATIONS

[0001] This application claims priority from U.S. Patent Application No. 11/368,959, filed March 6, 2006, the entire contents of which are incorporated herein by reference. The present application is also a continuation-in-part of U.S. Patent Application Serial No. 11/286,890 filed on November 23, 2005, which is a continuation-in-part of U.S. Patent Application Serial No. 10/854,604 filed on May 26, 2004, which is a continuation-in-part of U.S. Patent Application Serial No. 10/248,894 filed on February 27, 2003, which claims priority to U.S. Provisional Patent Application Serial No. 60/360,023 filed on February 27, 2002, the entire contents of which are all hereby incorporated by reference.

BACKGROUND OF THE INVENTION

[0002] Encryption techniques provide a mechanism for disguising data (i.e., plaintext or unencrypted data) such that the data cannot be obtained without proper validation or authentication (e.g., possessing a key). Generally, encryption or encipherment processes apply an encryption key to plaintext data in order to generate ciphertext. To obtain the plaintext data out of the ciphertext, a decryption key is applied to the ciphertext to convert the ciphertext back to plaintext. Different types of keys can be used to generate and decrypt ciphertext. For example, symmetric key systems use a single key to perform both encryption and decryption (i.e., the encryption key equals the decryption key), and public/private key systems use a separate encryption key and decryption key. Public/private key systems obtain their name from the fact that the decryption key (i.e., the private key) is kept secret or private, and the encryption key (i.e., the public key) is generally publicly available. The security of public/private key systems lies in keeping the decryption key secret and ensuring that the decryption key is not easily calculated from the public encryption key.

[0003] Cryptanalysis is the science of recovering plaintext without the appropriate validation or authentication (e.g., without possessing a key). Attempted cryptanalysis is often called an attack. Various attacks can be performed to circumvent the security of encryption techniques. For example, a ciphertext-only attack can be performed when an eavesdropper obtains or steals ciphertext and attempts to determine the plaintext and/or the key used to generate the ciphertext. If an eavesdropper obtains the plaintext in addition to obtaining the

WAVELYNX EX. 1003, Pg. 702
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

corresponding ciphertext, he or she can perform a plaintext attack and can attempt to determine the key used to generate the ciphertext. Even if the eavesdropper can obtain only a portion of the plaintext or a known format of the plaintext in addition to obtaining the corresponding ciphertext, he or she might still be able to determine the key used to generate the plaintext. In most situations, the more information an eavesdropper can obtain the easier he or she can determine a key. Once an eavesdropper has determined a key, the eavesdropper can use the key to decrypt ciphertext.

SUMMARY OF THE INVENTION

[0004] One embodiment of the invention provides a method of encrypting data. The method includes establishing data that includes discoverable or "white" data and undiscoverable or "black" data. Black data is generally unrecognizable. For example, it may be random data. White data generally has recognizable content or is transmitted in a recognizable format. The method also includes establishing a first key and a second key, where the second key is substantially underivable from the first key. The discoverable or white data is encrypted with the first key and the undiscoverable or black data is encrypted with the second key. In subsequent communications or transactions, at least one of the first key and the second key is mutated.

[0005] In another embodiment, messages are transmitted between an entity and an authenticator by establishing an identifier associated with the entity; establishing a first key to encrypt black data, where the first key is known only to the entity and the authenticator. A second key to encrypt white data is established. The second key is known only to the entity and the authenticator. A request identifier is established and encrypted with the second key to create an encrypted request. A hash of the request identifier is generated and the hash is encrypted with the first key to create an encrypted request hash. The encrypted request and the encrypted request hash are sent to the authenticator, and at least one of the first key and the second key is mutated.

[0006] In another embodiment, a system for encrypting data that includes discoverable data and undiscoverable data is provided. The system includes a sender having a first key and a second key. The second key is substantially underivable from the first key. The sender is configured to encrypt the discoverable data with the first key, to encrypt the undiscoverable

WAVELYNX EX. 1003, Pg. 703
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

data with the second key, and to receive at least one of a mutated first key and a mutated second key from an authenticator.

[0007] Other embodiments and details are illustrated and described in the drawings and detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] In the drawings:

[0009] FIG. 1 schematically illustrates a system for transmitting data according to one embodiment of the invention.

[0010] FIG. 2 illustrates a bit stream (called a "mutating ID") according to one embodiment of the invention.

[0011] FIGS. 3A and 3B illustrate ways of distributing mutating IDs.

[0012] FIG. 4 illustrates an exemplary brute force attack.

[0013] FIG. 5 illustrates types of data included in data to be encrypted.

[0014] FIGS. 6 and 7 illustrate an encryption strategy for protecting the security of undiscoverable or "black" data according to one embodiment of the invention.

[0015] FIGS. 8, 9, and 10 illustrate key extraction and mutation protocols according to various embodiments of the invention.

[0016] FIG. 11 illustrates a separate encryption protocol for communicating with an authenticator device according to one embodiment of the invention.

DETAILED DESCRIPTION

[0017] Before embodiments of the invention are explained in detail, it is to be understood that the invention is not limited in its application to the details of the construction and the arrangements of the components set forth in the following description or illustrated in the drawings. The invention is capable of still other embodiments and of being practiced or being carried out in various ways. Also, it is to be understood that the phraseology and

WAVELYNX EX. 1003, Pg. 704
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

terminology used herein is for the purpose of description and should not be regarded as limiting.

[0018] In particular, it should be understood that some embodiments are implemented using various computer devices, such as personal or home computers, servers, and other devices that have processors or that are capable of executing programs or sets of instructions, including special purpose devices such as set top boxes (e.g., digital cable or satellite decoders). In general, some embodiments may be implemented using existing hardware or hardware that could be readily created by those of ordinary skill in the art. Thus, the architecture of exemplary devices will not be explained in detail, except to note that the devices will generally have a processor, memory (of some kind), and input and output devices. In some cases, the devices may also have operating systems and application programs that are managed by the operating systems. In some embodiments, the hardware devices or software executed by the hardware devices also provides some ability, depending on the role of the device in the particular embodiment of the invention implemented, to compress or decompress data or to encode data or decode encrypted data. In some instances, a decompression capability may be provided using available codecs, such as hardware-implemented Moving Picture Experts Group ("MPEG") codecs. A decryption capability may be provided using a decryption hardware or software module capable of decrypting data that is encrypted using a particular encryption algorithm. In some embodiments, the encryption algorithm includes the Rijndael algorithm, an example of which is available at http://www.esat.kuleuven.ac.be/~rijmen/rijndael/rijndaelref.zip.

[0019] FIG. 1 illustrates an exemplary system 20 configured to distribute content over a network. In reality, one or more networks or communication systems such as the Internet, the telephone system, wireless networks, satellite networks, cable TV networks, and various other private and public networks could be used in various combinations to provide the communication links desired or needed to create embodiments or implementations of the invention, as would be apparent to one of ordinary skill in the art. Thus, the invention is not limited to any specific network or combinations of networks. However, in some embodiments, the networks or communication systems used in the system 20 have the ability to support digital and/or secure communications, such as communications with data encrypted with a version of Rijndael encryption or others. Furthermore, data can be

WAVELYNX EX. 1003, Pg. 705
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

transferred from one entity to another with wired communications, wireless communications, or physical media being physically carried from one entity to another.

[0020] In the embodiment shown in FIG. 1, the system 20 includes three participants: a first device 22, a second device 24, and an authenticator device 28. In the exemplary embodiment illustrated in FIG. 1, it is assumed that the first device 22 possesses data to be transmitted to the second device. Although FIG. 1 only illustrates the first device 22 and the second device 24, in some embodiments, numerous devices are included in the system 20, where at least one of the devices possesses data to be transmitted to another device. Furthermore, in some embodiments, the system 20 includes multiple authenticator devices 28.

[0021] The first device 22, the second device 24, and the authenticator device 28 are connected to each other via two-way links 30, 32, and 38. The links 30, 32, and 38 can include all or part of one or more of the networks mentioned above. In some embodiments, the system 20 uses a key-based encryption algorithm and currently available algorithms, such as the Rijndael algorithm. Choices for the algorithms used in the system 20 can depend on a variety of factors including a trade off between the strength of the algorithm (in terms of being broken) and speed (in terms of a processor's capability to perform the mathematical operations required by the chosen algorithm).

[0022] In some embodiments, as shown in FIG. 1, the authenticator device 28 uses a random number generator 39 to generate numbers used by a protocol implemented or followed by the system 20. The random number generator 39 can produce numbers that are as truly random (i.e., numbers that are as random as is possible with the particular technology used to implement the invention). For example, communication traffic, such as requests from customers to obtain content, can be used to create random numbers. Such requests occur, in general, in an unpredictable manner. Thus, the random numbers generated based on such traffic are also truly or nearly truly random, as opposed to pseudo random numbers generated with algorithmic methods.

[0023] In some embodiments, the first device 22 and the second device 24 use mutating identifiers ("IDs") to transmit data. An exemplary mutating ID 38 is shown in FIG. 2. The mutating ID 38 is an identifier having two portions: a first portion 40 and a second portion 42. The first portion 40 includes an identifying number, which is a random number. As

WAVELYNX EX. 1003, Pg. 706
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

indicated in FIG. 2, in some embodiments, the two portions of a mutating ID each include a number of bits. For example, the first portion 40 and the second portion 42 can each include 256 bits. In other embodiments, the first portion 40 and/or the second portion 42 include a larger number of bits, such as 1 megabit or 1 megabyte.

[0024] The second portion 42 includes a secret key, which is also a random number and, in some embodiments, is a symmetric cipher key. A mutating ID can be used only once and then cannot be used again for a long time.

[0025] In addition, although FIG. 2 illustrates a mutating ID has having only two portions, a mutating ID can include additional sections or portions. For example, as described below, a mutating ID can include an identifying number, a secret key for a first type of data (e.g., discoverable data), and a secret key for a second type of data (e.g., undiscoverable data).

[0026] Mutating IDs are generated and tracked by the authenticator device 28. Because mutating IDs are one-time-use mechanisms, once the first device 22, the second device 24, or another device uses its supply of mutating IDs, (e.g., a single mutating ID or multiple single mutating IDs) the device can obtain another mutating ID (or multiple mutating IDs if applicable) from the authenticator device 28. The data included in a mutating ID can be chosen at random with an equal probability of all possible mutating IDs.

[0027] FIGS. 3a and 3b illustrate how mutating IDs can be distributed from the authenticator device 28 to the first device 22 or the second device 24. As shown in FIG. 3a, in some embodiments, a device 43 (such as the first device 22 or the second device 24) may request multiple mutating IDs from the authenticator device 28. The authenticator device 28 creates as many mutating IDs as the device 43 requests and sends a list of mutating IDs to the device 43. The device 43, knowing the quantity of mutating IDs requested and the size of each mutating ID, breaks the list into individual mutating IDs. In some embodiments, the authenticator device 28 provides information or instructions to the device 43 to assist the device 43 in separating the mutating IDs. For example, the authenticator device 28 can provide information or instructions to the device 43 to assist the device 43 in separating the mutating IDs using a data description language, such as extensible markup language ("XML").

WAVELYNX EX. 1003, Pg. 707
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0028] As shown in FIG. 3b, in other embodiments, a device 43 can receive a single mutating ID upon requesting a new mutating ID or upon using a previously provided mutating ID. The single, new mutating ID is sent to the device 43, which replaces the mutating ID previously provided to the device 43.

[0029] In the embodiment shown in FIG. 1, the authenticator device 28 randomly assigns or provides a mutating ID to the first device 22 (hereinafter referred to in this example as the "first mutating ID") and a mutating ID to the second device 24 (hereinafter referred to in this example as the "second mutating ID"). The first mutating ID is different from the second mutating ID and each of the first mutating ID and the second mutating ID do not provide information for determining the other mutating ID. As described above with respect to FIG. 2, each mutating ID includes a random number 40 and a random corresponding secret key 42. In some embodiments, a mutating ID takes the form of a modified hash. As described above, in addition to being random, the mutating ID (or the hash if applicable) is discarded after each use. In other words, the authenticator device 28 provides a new mutating ID with a new random number and a new random secret key 42 after a mutating ID is used. In some embodiments, the mutating ID is completely unrelated from the device using it. That is, the mutating ID or the hash does not contain any information concerning the identity of the device. In this way, the identities of devices are blind to all participants except for the authenticator device 28.

[0030] Some embodiments of the invention implement symmetric key systems. Symmetric key systems commonly encounter key management issues as the number of entities or parties of the system grows. For example, a network of n entities requires $n(n-1)/2$ keys to enable all entities to communicate with one another. Thus, for a system of 1000 entities, where every entity wishes to send identical content to every other entity, almost a half million keys are required.

[0031] Disclosed embodiments, however, do not require a separate key for every pair of entities of the system. As will be illustrated, each entity and each piece of content distributed by each entity each receives one key, which is mutated after each use. Therefore, for a system of 1000 entities, only 2000 keys are required compared to the almost half of a million keys with previous symmetric key systems. Also, the authenticator device 28 is not required to store the entire bit string of the mutating ID. The authenticator device 28 may use a hash

WAVELYNX EX. 1003, Pg. 708
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

function or simply a positional index to map each key partition of the mutating ID into a memory storage location based on the corresponding number.

[0032] Other differences between embodiments of the invention and prior security systems relate to speed and reduced vulnerability to certain attacks. The use of symmetric keys also allows fast computation (as compared to public key systems). The fundamental concept behind public key systems is the use one-way functions. One-way functions are easy to compute but hard to reverse. Public key systems use trapdoor one-way functions that provide a key to compute the one-way function in the opposite direction. Public key systems provide public keys for each participant that are freely accessed and used as a one-way function to apply to a message. Public key systems also provide private keys (which are believed to be computationally infeasible to derive from the public key) to each individual participant to compute the message given the calculation of the one-way function. The security of public key systems relies on the assumption that the private key cannot be derived from the public key. In order to maintain this requirement, the one-way functions used in public key systems are complex. The added complexity, however, comes at the cost of added computation time. Public key systems are often 1000 times slower than symmetric key systems.

[0033] The use of symmetric keys also reduces the effectiveness of chosen plaintext attacks. A chosen-plaintext attack occurs when an intruder has access to an encryption key or process, chooses specific plaintext to encrypt, and attempts to gain knowledge from the encrypted text. In public-key systems an individual's public key is known to all participants in a communication system. Any intruder can encrypt an endless number of messages using an individual's public key. If an attacker encrypts possible messages with an individual's public key and then intercepts an encrypted message sent to the individual, the intruder can compare the intercepted message with messages he or she has created. If an interception message matches an encrypted message created by the intruder, the message has been compromised and the intruder can now read a message that was not intended for him or her. This attack is relatively easy and effective if a small number of possible messages exist, but even if the number of possible messages is more than the intruder is able to encrypt or compare with intercepted encrypted messages, just knowing that an intercepted encrypted message does not correspond to a particular message can provide useful information to the intruder. In either situation, the intruder will not be able to deduce the private key of the

WAVELYNX EX. 1003, Pg. 709
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

individual but the intruder may be able to deduce the message, or information regarding the message, sent to the individual. Since embodiments of the invention utilize a symmetric key system, chosen-plaintext attacks are not applicable because encryption keys are not public knowledge.

[0034] There is another problem with prior symmetric key systems and public key systems. Once an unauthorized entity gains access to an authorized key, the unauthorized entity can decode all messages encrypted with the compromised key, and, perhaps more dangerous, can encrypt false messages to deceive other entities of the system. The mutating ID protocol reduces this weakness by mutating each secret key after it has been used. Even if a secret key is compromised, the compromised secret key cannot be used to generate future messages nor be used to decrypt future messages since it is marked by the authenticator as used and is not used again to encrypt future messages.

[0035] The system 20 uses a protocol to govern communications between entities. Each entity is randomly assigned a mutating ID, such as the identifier or ID 38 shown in FIG. 2, by the authenticator device 28. As noted, each mutating ID includes a random number 40 and a random corresponding secret key 42. In some embodiments, a mutating ID takes the form of a modified hash.

[0036] The authenticator device 28 also generates encryption keys for content or data distributed through the system 20. A device wishing to transmit data requests a key from the authenticator device 28. The device sending the data (i.e., the sending device) supplies the authenticator device 28 with a label or function (i.e., any identifying string) of the data to transmit, and the authenticator device 28 responds with an associated key. The key, like the mutating IDs, is unrelated to the data that it encrypts. The authenticator device 28 also has no knowledge of the data since only an identifier (e.g., a random identifier) of the data is provided. The authenticator device 28 records the key and the associated identifier of the data.

[0037] In some embodiments, after the authenticator device 28 generates and supplies a key associated with the identifier of the data to the sending device, the sending device uses the key to encrypt the data. A device receiving the encrypted data (i.e., the receiving device) can request the corresponding decryption key (e.g., the same key used to encrypt the data) from the authenticator device 28. In some embodiments, the authenticator device 28 supplies

WAVELYNX EX. 1003, Pg. 710
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

a decryption key to any authorized receiving device included in the system 20 that makes a legitimate request. A request for a decryption key includes a reference to the label or identifying string of the data. The authenticator device 28 determines the associated key based on the label indicated in the request and returns the appropriate key to the receiving device.

[0038] Exemplary embodiments of the invention will now be described using several examples.

[0039] As with many descriptions of communication protocols, names are assigned to the various devices (or computer systems associated with those devices) used in the protocol. In one embodiment, Alice (*A*) and Bob (*B*) represent the first device 22 and the second device 24 respectively, and Trent (*T*) represents the authenticator device 28, a trusted arbiter of communication. Carol (C) can also represent a third device included in the system 20. The following table, Table 1, is a list of other symbols used in this document to explain multiple embodiments of the protocol.

Table 1

Symbol	Meaning
A,B,C,T	Entities (e.g., devices) included in the system.
M	Data or content.
X_{id}	An identifier (e.g., public identifier) for an entity X.
X_{cred}	Secret information that identifies an entity X, which is known only to the entity X and the authenticator and is randomly assigned by the authenticator.
K_X	A key for a symmetric cipher associated with some entity X.
N_X	A one-use number associated with some key K_X.
$H(X)$	A function that produces a hash of X.
$E(K,X)$	A cipher that encrypts X with K.
$X \rightarrow Y : Z$	A message Z sent from X to Y.
$\{(N^i_X, K^i_X)\}$	A set of number/key pairs of arbitrary size associated with entity X.
$XOR(Y,Z)$	Bitwise exclusive or of Y and Z

WAVELYNX EX. 1003, Pg. 711
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

SESSION KEYS

[0040] In some embodiments, mutating IDs are used to exchange a communication or session key between two entities. For example, assume that Alice and Bob would like to communicate securely. Again assume that Alice and Bob trust Trent and that Trent assigns Alice a mutating ID that includes a number N_A and a secret key K_A for some symmetric cipher and assigns Bob a mutating ID that includes a number N_B and a secret key K_B for some symmetric cipher. Also assume that Alice and Bob each have credentials (e.g., A_{cred} and B_{cred} respectively) that are known only to Trent and the holder of the credentials.

[0041] To request a session key K_{AB} from Trent, Alice encrypts her credentials and an identifier of Bob (e.g., B_{ID}) with her secret key K_A and appends her number N_A to the result. Alice sends the message to Bob.

$$A \rightarrow B: N_A E(K_A, A_{cred} B_{id})$$

[0042] Bob concatenates his credentials and an identifier of Alice (e.g., A_{id}) with the message from Alice and encrypts the result with his secret key K_B. Bob appends his number K_B to the result of the encryption and sends the result to Trent.

$$B \rightarrow T: N_B E(K_B, B_{cred} A_{id} N_A E(K_A, A_{cred} B_{id}))$$

[0043] Trent identifies that the message has come from Bob because Trent knows that the number N_B is associated with Bob. Trent decrypts the message using K_B and checks the credentials B_{cred}. Trent also decrypts and verifies the part of the message constructed by Alice. If Bob's credentials B_{cred} match his number N_B and his identifier B_{id} provided by Alice and Alice's credentials A_{cred} match her number N_A and her identifier A_{id} provided by Bob, Trent verifies the request. After verifying the request, Trent generates a message for Alice and a message for Bob. The message for Alice includes a new number N_A', a new secret key K_A', Alice's credentials A_{cred}, and a session key K_{AB}. Trent encrypts the message for Alice with Alice's current secret key K_A. The message for Bob includes a new number N_B', a new secret key K_B', Bob's credentials B_{cred}, and a session key K_{AB}. Trent encrypts the message for Bob with Bob's current secret key K_B. Trent sends the messages to Alice and Bob.

$$T \rightarrow A: E(K_A, N_A' K_A' A_{cred} K_{AB})$$

WAVELYNX EX. 1003, Pg. 712
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

$T \rightarrow B$: $E(K_B, N_B' K_B' B_{cred} K_{AB})$

[0044] The above protocol can be extended to include more entities. For example, if Alice wants a session key associated with Bob and Carol, Alice can list known identifiers of Bob and Carol, such as Bob's identifier B_{id} and an identifier of Carol (e.g., C_{id}) in her message. Similarly, Bob can list identifiers of Alice and Carol, and Carol can list identifiers of Alice and Bob. Each entity can also include their credentials in their message. As shown above, each entity can forward their message to another entity associated with the requested session key and each entity can add their message to the received message. Once all the intended entities have added their message to the request, the last entity forwards the request to Trent. Trent verifies that the credentials of each entity match the mutating IDs (e.g., the numbers of the mutating IDs) assigned to each entity and that the list of identifiers specified by each entity match the provided credentials. After verifying the request, Trent sends a new mutating ID (e.g., a new number and a new secret key) and the session key associated with the listed entities to each entity along with their credentials.

CONTENT USE LICENSES

[0045] Mutating IDs can also be used to provide a license that an entity can use to obtain and decode a piece of content. For example, assume Alice has content or a message P that she wants to securely send to Bob. Again assume that Alice and Bob trust Trent and that Trent assigns Alice a mutating ID that includes a number N_A and a secret key K_A for some symmetric cipher and assigns Bob a mutating ID that includes a number N_B and a secret key K_B for some symmetric cipher. Also assume that Alice and Bob each have credentials (e.g., A_{cred} and B_{cred} respectively) that are known only to Trent and the holder of the credentials.

[0046] To obtain a license for the message P, Alice generates a hash of the message P, concatenates the hash $H(P)$ with her credentials A_{cred}, and encrypts the result with her secret key K_A. Alice also appends her number N_A to the encryption result. Alice sends a license request to Trent.

$A \rightarrow T$: $N_A E(K_A, A_{cred} H(P))$

[0047] Trent decrypts the request from Alice and generates a response to Alice that includes a mutating ID that includes a new number N_A' and a new secret key K_A' for Alice, a

WAVELYNX EX. 1003, Pg. 713
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

mutating ID to be associated with a license for the message P that includes a license number $N_{H(P)}$ and a license secret key $K_{H(P)}$, and a key K_P for the message P. In some embodiments, Trent also includes the hash $H(P)$ in the response to Alice so that Alice can ensure that the message has not been tampered with. Trent encrypts the response with Alice's current secret key N_A and sends the encrypted response to Alice.

$$T \rightarrow A: E(K_A, N_A' K_A' N_{H(P)} K_{H(P)} K_P H(P))$$

[0048] Once Alice obtains the response from Trent, Alice decrypts the response and obtains the key K_P and the mutating ID associated with a license for the message P. Alice encrypts the message P with the key K_P and generates a license for the encrypted message P. The license for the encrypted message P includes Alice's credentials A_{cred} and the hash of the message $H(P)$. In some embodiments, the license also includes an identifier of the recipient of the license. For example, if Alice is going to send the license to Bob, the license can include an identifier of Bob, such as B_{id}. In some embodiments, an identifier of the recipient is excluded from the license in order to reduce the complexity of the protocol. For example, digital media production companies may not know ahead of time or track potential recipients of the content. Alice encrypts the license with the license secret key $K_{H(P)}$ and appends the associated license number $N_{H(P)}$ to the encryption result. Alice sends the encrypted message and the associated license to Bob.

$$A \rightarrow B: E(K_P, P) \quad \text{(encrypted content)}$$

$$A \rightarrow B: N_{H(P)} E(K_{H(P)}, A_{cred} H(P) B_{id}) \quad \text{(license)}$$

[0049] At some point after receiving the encrypted message and the associated license, Bob can request the decryption key for the encrypted message P. To generate a request for the decryption key, Bob concatenates his credentials B_{cred} to the license Alice generated and encrypts the result with his secret key K_B. Bob also appends his number N_B to the encrypted concatenation and sends the request to Trent.

$$B \rightarrow T: N_B E(K_B, B_{cred} N_{H(P)} E(K_{H(P)}, A_{cred} H(P) B_{id}))$$

[0050] Trent unrolls the encryption, and, if an identifier of Bob is included in the license, Trent verifies that the credentials B_{cred} and the number N_B match the identifier in the license Alice generated. Trent can also verify that the hash $H(P)$ matches the license number $N_{H(P)}$

WAVELYNX EX. 1003, Pg. 714
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

and the license secret key $K_{H(P)}$. After verifying the message from Bob, Trent sends Bob the key K_P that can be used to decrypt the encrypted message P, a mutating ID that includes a new number N_B' and a new secret key K_B' for Bob, and Bob's credentials B_{cred} all encrypted with Bob's current secret key K_B.

$$T \rightarrow B: E(K_B, N_B' K_B' K_P B_{cred})$$

[0051] If desired by Alice, Trent can inform Alice that Bob requested the decryption key.

$$T \rightarrow A: E(K_A', \text{"Bob requested the key associated with the identifier } H(P)\text{"})$$

[0052] After providing the decryption key to Bob, the license Alice provided to Bob is no longer valid because Trent has already seen the license number $N_{H(P)}$ and the license secret key $K_{H(P)}$ that comprise, in this instance, the one-time use mutating ID associated with the license for the message P.

[0053] As in the previous example, this protocol can be extended to include multiple entities by having each entity add their credentials to the license, encrypt the result with their assigned mutating ID, and forward the modified license to the next entity. For example, if Alice generates and sends a license to Carol who forwards the license to David who then sends the license to Bob, the resulting license received by Trent would be as follows:

$$T \rightarrow A: N_B E(K_B, B_{cred}\ N_D E(K_D, D_{cred}\ N_C E(K_C, C_{cred}\ N_{H(P)} E(K_{H(P)}, A_{cred}\ H(P)\ B_{id}))))$$

DIGITAL SIGNATURES

[0054] So far we have discussed the use of mutating identifiers to establish a session and to deliver a license. In another embodiment, mutating IDs are used as digital signatures. Assume that Alice and Bob each have a copy of a document S that includes a piece of information P that requires an agreement between Alice and Bob. For example, the document S can include a bill of sale and the piece of information P can include the final price for the bill of sale. Also assume that Carol is an arbiter of agreements (e.g., a credit card company or a bank) such that she may need to know the piece of information P but not necessarily the document S. Again assume that Alice, Bob, and Carol trust Trent and that Trent assigns Alice a mutating ID that includes a number N_A and a secret key K_A for some symmetric cipher, assigns Bob a mutating ID that includes a number N_B and a secret key K_B

WAVELYNX EX. 1003, Pg. 715
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

for some symmetric cipher, and assigns Carol a mutating ID that includes a number N_C and a secret key K_C for some symmetric cipher. Also assume that Alice, Bob, and Carol each have credentials (e.g., A_{cred}, B_{cred}, and C_{cred} respectively) that are known only to Trent and the holder of the credentials.

[0055] To initiate the signing of the document S, Alice generates a message that includes the document S or the hash of the document S and a hash of her credentials A_{cred}. In some embodiments, Alice disguises or encodes the message. For example, Alice can generate an *XOR* of the hash of the document S and her credentials A_{cred}. The message can also include the piece of information P. Alice encrypts the message with her secret key K_A, appends her number N_A, and sends the result to Bob.

$$A \rightarrow B: N_A E(K_A, XOR(H(S), H(A_{cred}))P)$$

[0056] Bob generates a similar message that can include an *XOR* of the hash of the document S and a hash of his credentials B_{cred}. In some embodiments, Bob also adds the piece of information P to the message. Bob adds his message to the message from Alice and encrypts the result with his secret key K_B. Bob also appends his number N_B and sends the result to Trent.

$$B \rightarrow T: N_B E(K_B, XOR(H(S), H(B_{cred}))P\ N_A E(K_A, XOR(H(S), H(A_{cred}))P))$$

[0057] Trent decrypts the message from Bob and verifies that the hashes of the document S generated by Alice and Bob are equivalent. If Alice and Bob included the piece of information P in their messages, Trent also verifies that the pieces of information P provided from Alice and Bob are equivalent. After verifying the message, Trent generates receipts for Alice and Bob. Alice's receipt includes an identifier of Bob (e.g., B_{id}), the hash of the document S, and, optionally, the piece of information P. Trent encrypts Alice's receipt with a receipt secret key $K_{receipt}$ that is part of a mutating ID associated with the receipts for Alice and Bob. Trent also appends an associated receipt number $N_{receipt}$ included in the mutating ID associated with the receipts for Alice and Bob to Alice's receipt. Trent then encrypts Alice's receipt, a mutating ID that includes a new number N_A' and a new secret key K_A' for Alice, and Alice's credentials A_{cred} with Alice's current secret key K_A and sends the result to Alice.

$$T \rightarrow A: E(K_A, N_A'\ K_A'\ A_{cred}\ N_{receipt}\ E(K_{receipt}, B_{id}\ H(S)\ P))$$

WAVELYNX EX. 1003, Pg. 716
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0058] Trent generates a similar receipt for Bob that includes an identifier of Alice (e.g., A_{id}), the hash of the document S, and, optionally, the piece of information P. Trent encrypts Bob's receipt with the same receipt key $K_{receipt}$ as he encrypted Alice's receipt and appends the same receipt number $N_{receipt}$ as he appended to Alice's receipt. Trent encrypts Bob's receipt, a sixth mutating ID that includes a new number N_B' and a new secret key K_B', and Bob's credentials B_{cred} with Bob's current secret key K_B and sends the result to Bob.

$$T \rightarrow B: E(K_B, N_B' K_B' B_{cred} N_{receipt} E(K_{receipt}, A_{id} H(S) P))$$

[0059] Alice and Bob present their receipts to Carol, and Carol forwards one or both of the receipts to Trent for verification. For example, assume that Alice provides her receipt to Carol. Carol adds her credentials to Alice's receipt and encrypts the result with her secret key K_C. Carol appends her number N_C to the result and sends the message to Trent.

$$C \rightarrow T: N_C E(K_C, C_{cred} N_{receipt} E(K_{receipt}, B_{id} H(S) P))$$

[0060] Trent decrypts the message from Carol and verifies Alice's receipt by decrypting the receipt (since Trent and only Trent knows $K_{receipt}$) and providing Carol with the receipt details. For example, Trent can generate a message for Carol that includes a mutating ID that includes a new number N_C' and a new secret key K_C' for Carol, Carol's credentials C_{cred}, an identifier of Alice (e.g., A_{id}), an identifier of Bob (e.g., B_{id}), the hash of the document S, and, optionally, the piece of information P. Trent encrypts the message with Carol's current secret key K_C and sends the result to Carol.

$$T \rightarrow C: N_C E(K_C, N_C' K_C' C_{cred} A_{id} B_{id} H(S) P)$$

[0061] Carol can use the information from Trent to arbitrate the agreement between Alice and Bob.

[0062] It should be understood that the above protocol can be expanded to include other numbers of entities.

[0063] As illustrated in the above examples, mutating IDs offer several advantages over static values when used in the above protocols. A first advantage is that an eavesdropper or attacker must capture the entire mutation history of an identifier or key of a mutating ID in order to attempt a ciphertext-only brute force attack.

WAVELYNX EX. 1003, Pg. 717
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0064] A second advantage of mutating IDs is that they provide a unique form of security and intrusion detection. The history of an entity's mutating IDs is kept as an audit trail by the authenticator device 28 as the entity's "mutating ID lineage." The mutating ID lineage tracks who each mutating ID as assigned to, the time the mutating ID was assigned, and the time the mutating ID was used. If a mutating ID or a portion thereof is somehow copied, stolen, or activated by an entity other than the authenticator device 28, one of two events will take place. In a first event, if the rightful owner of a mutating ID uses the mutating ID before an imposter does, the stolen or copied mutating ID that the imposter possess becomes obsolete and a future use of the mutating ID by the imposter will be immediately detected by the authenticator device 28. In a second event, if an attacker or imposer uses a stolen mutating ID before the mutating ID is used by the rightful owner, the mutating ID will become obsolete before the rightful owner uses the mutating ID. At a later time when the rightful owner does use the mutating ID, the authenticator device 28 will immediately identify the mutating ID as obsolete and the rightful owner of the mutating ID can be notified of the intrusion so that proper actions can be taken.

[0065] A third advantage of mutating IDs is that an entity can obtain a new mutating ID using an existing mutating ID at any time during the protocol regardless of the whether the entity has used the existing mutating ID. This periodic mutation can limit the time that an attacker has to steal a mutating ID and use a stolen mutating ID.

BLACK PROTOCOL

[0066] The secret keys of mutating IDs (e.g., K_A, K_B, K_C, and K_D) should remain secret in order to protect the security of the transmitted data. For example, if Trent provides Alice with a new mutating ID encrypted with Alice's current secret key (e.g., K_A), an eavesdropper who has determined Alice's current secret key can obtain Alice's new mutating ID provided by Trent. The eavesdropper can then use the new mutating ID to send false data and/or obtain the plaintext of future data exchanged between Alice and Trent.

[0067] As described above, eavesdroppers can determine (or attempt to determine) a key used to encrypt particular data by performing an attack. FIG. 4 illustrates a brute force attack. A brute force attack includes decrypting ciphertext with every possible key until a key is found that produces coherent or recognizable data (e.g., human readable data). As shown in FIG. 4, an eavesdropper determines an initial or zero candidate key (step 50). The

WAVELYNX EX. 1003, Pg. 718
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

eavesdropper then uses the candidate key to decrypt ciphertext (step 52). After decrypting the ciphertext, the eavesdropper can inspect the result (i.e., the candidate plaintext) to determine if the ciphertext decrypted with the candidate key produces coherent plaintext or a coherent pattern (step 54). If the eavesdropper obtains or knows the plaintext (or a portion or pattern thereof) corresponding to the obtained ciphertext, the eavesdropper can more easily determine whether a correct candidate key has been found. For example, if the eavesdropper obtains ciphertext and knows that the ciphertext includes an individual's name followed by a 4-digit personal identification number ("PIN"), the eavesdropper can apply candidate keys until a candidate key produces the plaintext including the individual's name. The eavesdropper can then assume, with some certainty, that remaining information included in the generated plaintext corresponds to the PIN.

[0068] As shown in FIG. 4, if the eavesdropper finds a coherent pattern in candidate plaintext generated by decrypting ciphertext with a particular candidate key (step 56), the eavesdropper knows, with some certainty, that the current candidate key equals or is the key used to generate the ciphertext (step 57).

[0069] If the eavesdropper does not find a coherent pattern in the candidate plaintext generated by decrypting ciphertext with a particular candidate key (step 56), the eavesdropper can modify the candidate key (e.g., increment the candidate key) (step 58), and can use the modified candidate key to decrypt the ciphertext (step 52) and inspect the generated candidate plaintext for coherent plaintext or a coherent pattern (step 54). Given enough processing power and time, the eavesdropper can continue this process until a particular candidate key generates candidate plaintext with coherent plaintext or a coherent pattern and, therefore, determine the key used to generate the ciphertext.

[0070] However, if the eavesdropper has no knowledge of the plaintext or a pattern of the plaintext (i.e., has no content hint), the eavesdropper's ability to determine whether a correct candidate key has been found is greatly reduced and, perhaps, eliminated. For example, if plaintext includes a random number encrypted with a particular key, no matter how many keys the eavesdropper attempts in a brute force attack, the eavesdropper will have no way to determine whether candidate plaintext is the true plaintext corresponding to the ciphertext. Assuming that the key space (i.e., the pool of candidate keys) and the plaintext space of the encryption algorithm (i.e., the pool of candidate plaintext) are substantially the same,

WAVELYNX EX. 1003, Pg. 719
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

decrypting an encrypted random number with any candidate key will produce a random number that is equally likely to be the original random number as every other random number produced by every other candidate key.

[0071] Referring to the session key exchange example described above involving Alice, Bob, and Trent, if any portion of an encrypted message is recognizable, known, becomes known, or includes any content hints, an eavesdropper could possibly perform a plaintext or partial-plaintext attack on the encrypted message and uncover a secret key of Alice or Bob used to encrypt the message. For example, assume that Alice sends the following message to Bob that is intercepted by an eavesdropper.

$$A \rightarrow B: N_A E(K_A, A_{cred}\, B_{id})$$

[0072] The eavesdropper can perform a brute force attack on the intercepted message because Bob's identifier B_{id} and the format of the above message are known or public. Thus, the eavesdropper can obtain Alice's secret key K_A and her credentials A_{cred}. Furthermore, once the eavesdropper obtains Alice's current secret key K_A, the eavesdropper can use Alice's current secret key K_A to obtain all data encrypted with Alice's current secret key K_A, such as her next mutating ID (e.g., N_A' and K_A').

[0073] An eavesdropper can use other knowledge about an encrypted message or the communication protocol used to generate an encrypted message to perform brute force attacks. For example, an eavesdropper can use the mutating ID number (e.g., N_A), which is passed in the clear, to perform a brute force attack. An eavesdropper could also use knowledge of the algorithm used to generate the mutating ID numbers to perform a brute force attack.

[0074] Keys used to encrypt undiscoverable or "black" data (i.e., data that is random or has no content hints) cannot be determined or discovered using a brute force attack, since an eavesdropper will be unable to determine when a correct candidate key is found. Keys used to encrypt discoverable or "white" data (i.e., data that is known, may be later disclosed, is recognizable, or has a known or easily guessed format), however, can (theoretically) be determined using a brute force attack. When the white data and the black data are encrypted together or with the same encryption key, the key determined through a brute force attack

WAVELYNX EX. 1003, Pg. 720
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

using the white data is also the key used to encrypt the black data and, therefore, the black data can be discovered.

[0075] To increase the security of encrypted data and to reduce the effectiveness of brute force attacks, some embodiments of the invention provide an encryption strategy that protects the security of black data, such as the secret keys included in mutating IDs.

[0076] FIG. 5 illustrates types of data that can be included in data that is to be encrypted. As shown in FIG. 5, data 59 that is to be encrypted and transmitted to a particular receiver is separated into types or classes of data. A first class of data includes a black data class 60. The black data class 60 includes data that is kept secret and only known by authorized entities. For example, the black data class 60 can include the secret keys of the mutating IDs and/or the credentials of the entities included in the system 20, which are both random and known only by the authenticator device 28 and the holder of the credentials and the secret keys.

[0077] As shown in FIG. 5, a second class of data includes a white data class 62. The white data class 62 includes data that is publicly-known, recognizable (e.g., human-readable), or has a known or easily guessed format. For example, the white data class 62 includes data that has easily distinguished characteristics, such as standardized headers, known patterns, or other publicly available formats, such as content transmitted between entities (e.g., human-readable messages, movies, etc.) and the numbers (e.g., N_A, N_B, N_C, and N_D) of the mutating IDs provided by authenticator device 28.

[0078] The white data class 62 also includes indirect white data or keys used to encrypt messages that include white data, such as the secret keys (e.g., K_A, K_B, K_C, and K_D) of the mutating IDs. For example, as described above with respect to the session key exchange protocol, Alice sends Bob a message that includes Bob's publicly-known identifier B_{id} encrypted with her secret key K_A.

$$A \rightarrow B\text{: } N_A E(K_A, A_{cred}\, B_{id})$$

[0079] Since Bob's identifier B_{id} is publicly known and, therefore, is white data, Alice's secret key K_A can be considered indirect white data because it encrypts data that is publicly-known.

WAVELYNX EX. 1003, Pg. 721
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0080] FIGS. 6 and 7 illustrate an encryption strategy for protecting the security of black data according to one embodiment of the invention. To protect the security of the black data, separate keys can be used to encrypt the different types of data (hereinafter referred to as "separate encryption protocols"). For example, one or more keys (perhaps keys that are part of one or more mutating IDs) can be used to encrypt the black data and one or more different keys (perhaps keys that are part of one or more different mutating IDs) can be used to encrypt the white data. As described below, since the same keys are never used to encrypt black data and white data, the security of the black data is increased.

[0081] As shown in FIG. 6, data included in the black data class 60 can be encrypted with one or more keys 70 that are only used to encrypt black data (hereinafter referred to in this example as "black data keys 70"). Optionally, data included in the white data class 62 can be encrypted with one or more keys 72 that are only used to encrypt white data (hereinafter referred to in this example as "white data keys 72"). It should be understood that the black data keys 70 cannot be determined from (or are unrelated to) the white data keys 72. It should also be understood that the data 59 does not need to be separated and placed in contiguous blocks of data according to the data class the portions of the data 59 belong to. As shown in FIG. 7, data included in the black data class 60 and the white data class 62 can be divided into a number of portions that are mixed together.

[0082] By separating the black data from the white data and encrypting the black data with keys 70 that are different from the keys 72 used to encrypt the white data, even if a white data key 72 used to encrypt white data is determined using a brute force attack, the determined white data key 72 cannot be used to obtain black data.

EXCHANGING RANDOMLY-GENERATED SECRET CONTENT

[0083] In the following examples, the subscript i is used to denote an iteration count, where i is initially set to 0 or another predetermined value and is incremented by both parties involved in a protocol exchange after each exchange and/or on a different predetermined schedule. Therefore, the current value of a particular item is denoted with a subscript of i and a new or mutated value of the item is denoted with a subscript of $i+1$, rather than with a superscript of $'$ as was used in previous examples.

WAVELYNX EX. 1003, Pg. 722
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0084] The separate encryption protocol can be used to allow two entities to exchange randomly-generated secret data. For example, assume that a first entity *A* wants to exchange randomly-generated secret content S_i (i.e., black data) with a second entity *B*. Entities *A* and *B* share a previously-established randomly-generated secret encryption key K_i. Entities *A* and *B* can agree on the key K_i in person, one entity can send the secret key K_i to the other entity via secure communication means, the secret key K_i can be pre-stored in software or hardware operated by the entities, etc.

[0085] After establishing the secret encryption key K_i, entity *A* encrypts the secret data S_i and a new secret, random encryption key K_{i+1} with the encryption key K_i. Alice randomly generates the new secret key K_{i+1}. Entities A and B use the new secret key K_{i+1} to encrypt secret data in a future data exchange.

$$A \rightarrow B: E(K_i, K_{i+1} S_i)$$

[0086] In some variants, entity *A* can generate an *XOR* of the encryption key K_i and the new encryption key K_{i+1}.

$$XOR(K_i, K_{i+1})$$

[0087] After performing the *XOR* operation, entity *A* encrypts the bit string resulting from the *XOR* operation and the secret data S_i with the encryption key K_i and sends the result to entity *B*.

$$A \rightarrow B: E(K_i, XOR(K_i, K_{i+1}) S_i)$$

[0088] Using knowledge of the encryption key K_i, entity *B* can determine the new encryption key K_{i+1}.

[0089] For both cases, the message from entity *A* includes two parts: a key-mutation part (i.e., K_{i+1} or $XOR(K_i, K_{i+1})$) and a secret content part (i.e., S_i). Also in both cases, the key-mutation part is randomly-generated by entity *A*, and subsequent key-mutations are independent of prior key values.

[0090] Assuming the key space of the secret keys K_i and the plaintext space of the encryption function *E* are isomorphic, every candidate value of the secret key K_i will yield a completely possible candidate cleartext, and each candidate cleartext contains enough

WAVELYNX EX. 1003, Pg. 723
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

information to calculate a completely possible next key value. Therefore, an eavesdropper cannot rule out any possible initial key values and the protocol is substantially impervious to a ciphertext-only brute force attack.

COMMUNICATING WITH AN AUTHENTICATOR

[0091] Separate encryption protocols can also be used by the entities of the system 20, as shown in FIG. 1, to communicate with the authenticator device 28 in order to request session keys, request licenses, etc. For example, assume Alice, represented by the first device 22 wants to send a request to Trent, represented by the authenticator device 28. Trent has previously assigned Alice an initial randomly-generated secret encryption key, K_i, and a corresponding initial public identifier N_i (e.g., which may be part of a mutating ID assigned by Trent).

[0092] The parties also agree on a request and response protocol, which is shown in generalized form as a request consisting of a black request parameter set, B_{REQ}, and a white request parameter set, W_{REQ}, and a black response parameter set, B_{RSP}, and a white response parameter set, W_{RSP}. In some implementations, some of the parameter sets may be empty. Both the white request parameter set W_{REQ} and the white response parameter set W_{RSP} include publicly-known (i.e., white or discoverable data) request and response parameters, respectively. The black request parameter set B_{REQ} is underivable from the white request parameter set W_{REQ}.

[0093] In some embodiments of the invention, a white request parameter set W_{REQ} is paired with the black request parameter set B_{REQ} in order to form a valid request. Since the black request parameter set B_{REQ} cannot be determined or derived from the white request parameter set W_{REQ}, an eavesdropper cannot make a request to Trent posing as Alice without also providing the black request parameter set B_{REQ}. As noted, however, black content is undiscoverable. Therefore, knowledge of the white request parameter set W_{REQ} alone, does not allow an eavesdropper to forge a request. Similarly, the white response parameter set W_{RSP} is paired with the black response parameter set B_{RSP} and an eavesdropper who obtains the white response parameter set W_{RSP} could not send a response to Alice posing as Trent without also knowing the corresponding black response parameter set B_{RSP}.

WAVELYNX EX. 1003, Pg. 724
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0094] To send a request, Alice encrypts the black request parameter set B_{REQ} with the encryption key K_i. To identify herself as the sender of the request to Trent, Alice appends the public identifier N_i to the encryption result. In some variants, Alice also appends the corresponding white request parameter set W_{REQ} to the encryption result. Alice sends the resulting message to Trent.

$$A \rightarrow T: N_i\ W_{REQ}\ E(K_i, B_{REQ})$$

[0095] In some variants, Trent can assign Alice a separate encryption key that Alice can use to encrypt white data, such as the white request parameter set W_{REQ}. The encrypted white data is appended to the encrypted black response parameter set B_{REQ}. To prevent cipher-text-only brute force attacks from successfully identifying the randomly-generated secret black request parameter set B_{REQ} and the encryption key K_i, however, the white request parameter set W_{REQ} and any other white data cannot be encrypted in the same encryption package or envelope as the black request parameter set B_{REQ} or any other black data.

[0096] Trent receives the message from Alice and identifies Alice as the sender based on the identifier N_i appended to the message. After Trent determines that Alice sent the message, Trent uses the encryption key K_i to decrypt the message and obtain the black request parameter set B_{REQ}. Trent verifies that the black request parameter set B_{REQ} is valid. If Alice also provided the white request parameter set W_{REQ}, Trent also verifies that the white request parameter set W_{REQ} is valid (e.g., that there is a proper association or matching between it and the black request parameter set B_{REQ}). If the white request parameter set W_{REQ} is invalid, Trent rejects the request. Mismatched parameter sets can also signal that the request message was been tampered with (e.g., by a man-in-the middle attack) or may have been sent by an imposter.

[0097] To send a response to Alice, Trent encrypts the black response parameter set B_{RSP} and a new randomly-generated, secret encryption key K_{i+1} with the current encryption key K_i. In some variants, as described above, Trent replaces the new encryption key K_{i+1} in the response to Alice with the results of performing an *XOR* operation on the current encryption key K_i and the new encryption key K_{i+1}. Trent also generates a new identifier N_{i+1} for Alice and appends the new identifier N_{i+1} to the result of the encryption. In some variants, Trent also appends the corresponding white response parameter set W_{RSP} to the encryption result. As described above, rather than appending the white response parameter set W_{RSP} as

WAVELYNX EX. 1003, Pg. 725
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

plaintext, Trent can encrypt the white response parameter set W_{RSP} with a white data key known by Trent and Alice. In addition, Trent can use the white data key to provide Alice with a new or mutated white data key. Trent sends the resulting message to Alice.

$$T \rightarrow A: N_{i+1}\ W_{RSP}\ E(K_i, XOR(K_i, K_{i+1})\ B_{RSP})$$

[0098] Trent also marks the current encryption key K_i and the current identifier N_i as used and ensures that the key and the identifier are not reused for a long time. In some embodiments, Trent tracks the time that a key or identifier was issued and the time that it was later used in a request. Trent can use this information to make sure that a key or identifier is not reused until a predetermined amount of time has passed and/or to track when a key or identifier was used in improperly (e.g., by an entity posing as the rightful owner of the key or identifier).

[0099] Alice decrypts the message from Trent and obtains the new encryption key K_{i+1} and the black response parameter set B_{RSP}. Alice verifies that the black response parameter set B_{RSP} is valid. If the black response parameter set B_{RSP} is invalid, Alice can identify that the response was not sent by Trent and can disregard the response. If the message from Trent also includes the white response parameter set W_{RSP}, Alice also verifies that the white response parameter set W_{RSP} is valid. If the white response parameter set W_{RSP} is invalid, Alice can identify that the response was not sent by Trent and can disregard the response.

BLACK PROTOCOL WITH MESSAGE AUTHENTICATION

[00100] The protocols described above introduced separate encryption of black data using a mutating key or identifier (which can be parts of mutating IDs). Although the above protocols protect black data from ciphertext-only brute force attacks, the protocols provide limited message authentication or anti-tampering mechanisms. Consequently, an attacker may be able to inject bad data in to a message.

[00101] One way to provide message authentication and an anti-tampering mechanism involves Trent assigning Alice an initial large secret key K_i. Trent and Alice also independently keep an iteration count i. Alice and Trent can independently update or increment the iteration count i after each exchange and/or on another predetermined schedule.

WAVELYNX EX. 1003, Pg. 726
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00102] Alice uses the large secret key K_i and the iteration count i to exchange secret data (i.e., black data) S_i with Trent. For example, Trent can send Alice the secret data S_i using the following message protocol.

$$T \rightarrow A: E(K_{FX}(i, K_i), S_i M_i H(I_{FX}(i, K_i), S_i M_i F_{EX}(i, K_i)))$$

[00103] In the above message protocol, Trent encrypts a message to Alice with an encryption key generated by a key function K_{FX}. The key function K_{FX} generates a symmetric encryption key value using the iteration count i and the large secret key K_i.

[00104] The contents of the message from Trent to Alice include the secret data S_i, a mutation value M_i, and the result of a mutating secure-hash function H. The mutation value M_i is a randomly generated number based on the iteration count i and uniquely identifies a message.

[00105] The secure-hash function H generates a secure hash of contents of the message from Alice to Bob. In the above example, the secure-hash function H generates a secure hash of the secret data S_i, the mutation value M_i, and the large secret key K_i or a portion thereof. To use only a portion of the large secret key K_i in the hash, an extract function F_{EX} can be used. The extract function F_{EX} extracts and returns a portion of the large secret key K_i based on the iteration count i. In some variants, the extract function F_{EX} returns the entire large secret key K_i.

[00106] The secure-hash function H is initialized with an initialization value, such as an initialization vector I. The initialization vector I is generated with an initialization vector function I_{FX}. The initialization vector function I_{FX} generates an initialization vector I based on the iteration count i and the large secret key K_i.

[00107] Alice can decrypt the message from Trent since Alice knows the iteration count i, the large secret key K_i, and the functions used in the message. Decrypting the message allows Alice to obtain the secret data S_i, the mutation value M_i, and the hash $H(I_{FX}(i, K_i), S_i M_i F_{EX}(i,K_i))$. To make sure that the message from Trent was not tampered with during transmission, Alice authenticates the message from Trent by generating her own version of the hash that Trent included in the message. To generate her own version of the hash, Alice generates a hash of the secret data S_i and mutating value M_i obtained from the message from

WAVELYNX EX. 1003, Pg. 727
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Trent and the large secret key K_i. Alice uses the same secure hash function H as Trent. In some variants, Alice also generates an initialization vector I_{FX} for the hash function H as Trent did. In addition, Alice can use the same extraction function F_{EX} to extract a portion of the large secret key K_i to use in the hash as Trent did.

[00108] After generating her own version of the hash, Alice compares her version of the hash to the hash included in the message from Trent. If the hashes match or are the same, Alice can assume that the message from Trent was not tampered with. If the hashes do not match, it is likely that the message was tampered with and Alice can disregard it.

[00109] In some embodiments, Alice and Trent each individually mutate the large secret key K_i before using the large secret key K_i in a subsequent message. To accomplish this, Alice and Trent use a mutation function F_M to mutate the large secret key K_i. The mutation function F_M generates a new or mutated large secret key K_{i+1} based on the iteration count i, the mutation value M_i used in the most recent message, and/or the current large secret key K_i. For example, as illustrated in FIG. 8, the mutation function F_M can perform the function M_i *XOR* K_i, wherein M_i and K_i are substantially the same length.

$$K_{i+1} = F_M(i, M_i, K_i) = M_i \, XOR \, K_i$$

[00110] Using the above message protocol, an attacker desiring to obtain the secret data S_i or the large secret key K_i using brute force must first learn all of the algorithms used in the message protocol. The attacker must also obtain all of the messages exchanged between Alice and Trent. Next, the attacker must assume a candidate key C_i and use the candidate key C_i to decrypt a first message (i.e., iteration zero) exchanged between Alice and Trent. Decrypting the first message with the candidate key C_i reveals candidate text X_i. Since the attacker knows the algorithms of the message protocol, the attacker can divide the candidate text X_i into candidate secret data S_{iX}, a candidate mutation value M_{iX}, and a candidate secure-hash function result H_{iX}. Using the candidate secret data S_{iX}, the candidate mutation value M_{iX}, and the candidate key C_i, the attacker creates his or her own version of the secure-hash function result H_{iT} (e.g., a test result of the secure-hash function H for iteration i). If the test result of the secure-hash function H_{iT} matches the candidate secure-hash function result H_{iX}, the attacker can mark the candidate key C_i as an acceptable or potential candidate key. If the test result of the secure-hash function H_{iT} does not match the candidate secure-hash function result H_{iX}, the attacker can rule out the candidate key C_i as a potential correct key.

WAVELYNX EX. 1003, Pg. 728
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00111] For each acceptable or potential candidate key C_i, the attacker applies the mutation function F_M and generates a candidate key C_{i+1} for the second or next message exchanged between Alice and Trent. The attacker can then use the mutated acceptable candidate key C_{i+1} to perform the above encryption and test process as described above for the first message.

[00112] Given a large set of randomly generated input buffers, an ideal secure hash produces an unbiased and evenly distributed set of output values that fall between *0* and $(2^h - 1)$, where h is the number of bits returned by the hash function. Because of the periodicity of the secure-hash function H, each message exchanged between Alice and Trent reduces the potential key space (the list of possible keys) by a factor of 2^h, where h is the number of bits returned by the secure-hash function H. If the secret key is m bits in length, the attack may be capable of determining the initial secret key (e.g., secret key K_i) after (m/h) messages are exchanged.

[00113] To reduce the likelihood of success of a brute force attack, the large secret key K_i can be reset before the $(m/h)^{th}$ message is exchanged. In addition, if the large secret key K_i is reset j messages before the $(m/h)^{th}$ message is exchanged, a brute force attack would still leave 2^{jh} spurious large key values, and the correct or true large secret key value would be indistinguishable.

[00114] One possible embodiment of the invention can include a secure-hash function H that returns a 160-bit hash value and a large secret key K_i including 64 kilobytes. Using these parameters, the message protocol as described above would allow over 3,000 messages to be exchanged with limited risk of an effective brute force attack. In addition, the extraction function F_{EX} and the mutation function F_M can be selected in such a way that a moving window portion of the large secret key K_i is used in the message hash, and that the mutation of the large secret key K_i occurs in relatively narrow strips while still impacting the next iteration of the large secret key K_i. This can allow the protocol to have relatively low computational overhead, while maintaining approximately the same level of security.

[00115] As noted above and as shown in FIG. 8, in one embodiment the extraction function $F_{EX}(i,K_i)$ returns the entire large secret key K_i, and the mutation function $F_M(i, M_i, K_i)$ returns M_i *XOR* K_i, where M_i and K_i are equal length. Although these functions produce a working system, they can introduce substantial computational overhead for the secure hash

WAVELYNX EX. 1003, Pg. 729
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

function, since the entire large secret key K_i is appended to the end of message prior to calculating the hash. The functions can also introduce substantial protocol overhead, since the mutation value transmitted in response to each request is as long as the large secret key K_i.

[00116] As shown in FIG. 9, to address the above overhead issues, the extraction function F_{EX} can be modified to select p bits of the large secret key K_i at an offset of $(i \times h)$ bits from the start of the large secret key K_i, where h is the number of bits returned by the secure hash function H. Thus, the used portion of the large secret key is $((i \times h) + p)$ bits, and the ocean of spurious large keys contains 2^p values. Using this version of the extraction function F_{EX} allows for the use of larger secret key values without experiencing increased computational overhead in the calculation of the secure hash function.

[00117] Similarly, as also shown in FIG. 9, the mutation function F_M can receive a random mutation value of p bits and apply the mutation value via an *XOR* to the large secret key K_i at an offset of $((i+1) \times h)$ bits. In this way, the protocol overhead associated with the mutation is reduced to p bits and is independent of the size of the initial large secret key K_i. In addition, all of the bits that will be used in the next iteration of the protocol are mutated, which eliminates any mathematical correlation between the two key values.

[00118] As shown in FIG. 10, the extract function F_{EX} can also be modified to select p bits of the large secret key K_i at an offset of $(i \times p)$ bits from the start of the large secret key K_i. Thus, an entirely-new portion of the large secret key K_i is selected each time the extract function F_{EX} is used.

[00119] Similarly, as also shown in FIG. 10, the mutation function F_M can receive a random mutation value of p bits and apply the mutation value via an *XOR* to the large secret key K_i at an offset of $((i+1) \times p)$ bits.

[00120] Using this "updating in stripes" methodology, the initial large secret key K_i can be quite large and the system can support a very large number of message exchanges before a reset of public identifiers and associated keys is needed.

WAVELYNX EX. 1003, Pg. 730
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

BLACK PROTOCOL REQUESTS AND RESPONSES

[00121] FIG. 11 illustrates how the black protocol can be used to exchange requests and responses between entities (e.g., the first device 22 and the second device 24) and an authenticator (such as the device 28). Implementations of the protocol can be used by an entity to request a periodic mutation of a mutating ID, to request an encryption key, to request a decryption key, to get an authentication token (e.g., credentials), and to request authentication of an authentication token. Similarly, the authenticator device 28 can use the black protocol to provide responses to the entity for each type of request. As shown in FIG. 8, an entity can communicate with other entities using the responses from the authenticator. For example, an entity can request an encryption key from the authenticator, receive an encryption key from the authenticator, use the encryption key to encrypt a message for another entity, and send the encrypted message to the other entity. As also shown in FIG. 8, the entity receiving the encrypted message can request a decryption key from the authenticator, receive a decryption key from the authenticator, and use the decryption key to decrypt the encrypted message.

[00122] In one embodiment, Trent (the authenticator device 28) assigns Alice (the first device 22) an initial public identifier N_i, an initial secret key k_i, and an initial large secret key K_i. The public identifier N_i, the secret key k_i, and the large secret key K_i can be referred to as large mutating ID. Trent and Alice also independently keep an iteration count i, which is often initialized with a value of zero. Alice and Trent can independently update or increment the iteration count i after each request, after each response, or on another predetermined schedule.

[00123] Alice uses the public identifier N_i to identify herself as the originator of messages. Alice uses the secret key k_i (hereinafter referred to as Alice's white data key k_i) to encrypt white data, and Alice uses the large secret key K_i for a variety of purposes, including encryption of black data. Alice may also use the large secret key K_i as input to an initialization function I_{FX} in order to generate an initialization vector for a hash function H. In some embodiments, Trent assigns Alice a new or mutated public identifier N_i and white data key k_i after each use or periodically as requested by Alice or enforced by Trent, and Trent and Alice each individually mutate the large secret key K_i after each transaction or periodically as requested by Alice or enforced by Trent.

WAVELYNX EX. 1003, Pg. 731
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00124] Using the large mutating ID, Alice generates a request for Trent by encrypting request parameters T_{REQ} with her white data key k_i. The request parameters T_{REQ} include any information that is needed by Trent to respond to a particular request, such as a hash of a content for which Alice is requesting an encryption key for. The request parameters T_{REQ} are considered white data. Alice appends her public identifier N_i to the encryption result in order to identify herself as the sender of the request.

$$N_i\ E(k_i,\ T_{REQ})$$

[00125] To provide message authentication for the request, Alice generates a hash of her public identifier N_i, the request parameters T_{REQ}, and the large secret key K_i using a secure-hash function H. As described above, in some variants, Alice uses only a portion of the large secret key K_i in the hash and uses an extract function F_{EX} to extract a particular portion of the large secret key K_i based on the iteration count i and/or the number of bits returned by the secure-hash function H. In addition, Alice can use the initialization function I_{FX} to generate an initialization vector for the hash function H based on the iteration count i and the large secret key K_i.

$$H(I_{FX}(i,\ K_i),\ N_i\ T_{REQ}\ F_{EX}(i,\ K_i))$$

[00126] Alice encrypts the hash with the large secret key K_i or a portion thereof. For example, as described above, Alice can encrypt the hash with a portion of the large secret key K_i determined by the key function K_{FX} based on the iteration count i.

$$E(K_{FX}(i,\ K_i),\ H(I_{FX}(i,\ K_i),\ N_i\ T_{REQ}\ F_{EX}(i,\ K_i)))$$

[00127] Alice appends the encrypted hash to the encrypted request parameters with the appended public identifier N_i and sends the result to Trent.

$$T \rightarrow A\colon N_i\ E(k_i,\ T_{REQ})\ E(K_{FX}(i,\ K_i),\ H(I_{FX}(i,\ K_i),\ N_i\ T_{REQ}\ F_{EX}(i,\ K_i)))$$

[00128] Upon receiving the request from Alice, Trent determines that Alice sent the request based on the identifier N_i and decrypts the portions of the request using the white data key k_i and the large secret key K_i. After Trent obtains the request parameters T_{REQ} and the hash from the request, Trent generates his own version of the hash based on the request parameters obtained from the request. To authenticate the request, Trent compares his

WAVELYNX EX. 1003, Pg. 732
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

version of the hash to the hash obtained from the request. If the hashes match, Trent can verify that the white data was not tampered with during transmission of the request from Alice to Trent.

[00129] To generate a response for Alice, Trent generates a new or mutated public identifier N_{i+1} and a new white data key k_{i+1} for Alice. Trent encrypts the new white data key k_{i+1} and the response parameters T_{RSP} with Alice's current white data key k_i. Trent appends the new public identifier N_{i+1} to the encryption result.

$$N_{i+1}\, E(k_i,\, k_{i+1}\, T_{RSP})$$

[00130] To provide message authentication for the response, Trent generates a hash of Alice's new public identifier N_{i+1}, Alice's new white data key k_{i+1}, the response parameters T_{RSP}, a random mutation value M_i generated by Trent, and Alice's large secret key K_i using the secure-hash function H. As described above, in some variants, Trent uses only a portion of the large secret key K_i in the hash and uses the extract function F_{EX} to extract a particular portion of the large secret key K_i based on the iteration count i and/or the number of bits returned by the secure-hash function H. In addition, Trent can use the initialization function I_{FX} to generate an initialization vector for the hash function H based on the iteration count i and the large secret key K_i.

$$H(I_{FX}(i,\, K_i),\, N_{i+1}\, k_{i+1}\, T_{RSP}\, M_i\, F_{EX}(i,\, K_i))$$

[00131] Trent encrypts the hash and the mutation value M_i included in the hash with Alice's large secret key K_i or a portion thereof. For example, as described above, Trent can encrypt the hash with a portion of the large secret key K_i determined by the key function K_{FX} based on the iteration count i.

$$E(K_{FX}(i,\, K_i),\, M\, H(I_{FX}(i,\, K_i),\, N_{i+1}\, k_{i+1}\, T_{RSP}\, M_i\, F_{EX}(i,\, K_i)))$$

[00132] Trent appends the encrypted hash to the encrypted response parameters with the appended new public identifier N_{i+1} and sends the result to Alice.

$$A \rightarrow T\colon N_{i+1} E(k_i,\, k_{i+1}\, T_{RSP})\; E(K_{FX}(i,\, K_i),\, M_i\, H(I_{FX}(i,\, K_i),\, N_{i+1}\, k_{i+1}\, T_{RSP}\, M_i\, F_{EX}(i,\, K_i)))$$

[00133] Upon receiving the response from Trent, Alice obtains the new public identifier N_{i+1} and decrypts the remaining portion of the response in order to obtain the new white data

WAVELYNX EX. 1003, Pg. 733
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

key k_{i+1}, the response parameters T_{RSP}, the mutation value M_i, and the hash. Alice can verify the new public identifier N_{i+1}, the new white data key k_{i+1}, the response parameters T_{RSP}, and the mutation value M_i by generating her own version of the hash and comparing her version of the hash to the hash included in the response from Trent.

[00134] Alice and Trent use the mutation value M_i to independently generate a new or mutated large secret key K_{i+1}. As described above, Alice and Trent can use a mutation function F_M to generate a new large secret key K_{i+1} based on the iteration count i, the mutation value M_i randomly-generated by Trent, the current large secret key K_i, and/or the number of bits returned by the hash function H.

[00135] As illustrated in the above messages, the requests and responses include a similar layout and each include an identifier (e.g., N_i or N_{i+1}), encrypted white data (e.g., $E(k_i, T_{REQ})$ or $E(k_i, k_{i+1} T_{RSP})$), and encrypted black data including a message authentication code or hash (e.g., $E(K_{FX}(i, K_i), H(...))$ or $E(K_{FX}(i, K_i), M_i H(...))$). It should be understood that the white data (e.g., T_{REQ}, T_{RSP}, and k_{i+1}) could be also be transmitted as plaintext rather than being encrypted with the white data key k_i.

[00136] It should be understood that the above orderings and groupings of the request and response formats are for purpose of example only and can be varied.

[00137] Using the above request and response formats, Alice and Bob can exchange various types of requests and responses where only the request and response parameters are changed. Further specific examples of the message protocol therefore are expressed using the following abbreviated representation:

$$A \rightarrow T: BCP_{REQ}(T_{REQ})$$

$$T \rightarrow A: BCP_{RSP}(T_{RSP})$$

[00138] In some embodiments, the request and response parameters include command request codes that identify a message as a particular type of request or response. For example, assume that Trent assigns Alice the following randomly-generated initial command codes.

μ_i = "Mutate Mutating ID" command code

WAVELYNX EX. 1003, Pg. 734
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

α_i = "Get Encryption Key" command code

δ_i = "Get Decryption Key" command code

[00139] Each command code can be the same length as encryption keys provided by Trent. Each command code mutates or is changed after each use.

[00140] Since the request and response parameters can include different types of data having different lengths, the above message protocol can use randomly-generated padding values (a "*PAD*") to ensure that all requests are the same size and all responses are the same size. By making sure that each request or response is the same size, an eavesdropper cannot determine the type of a request or a response based strictly on the number of bits it includes.

[00141] Using the above command codes and padding values, to generate a "Mutate Mutating ID" request, Alice generates a request for Trent with the following exemplary form:

$A \rightarrow T$: $BCP_{REQ}(\mu_i\ PAD[X])$

where $PAD[X]$ is X bits of random padding.

[00142] In response, Trent generates a response for Alice with the following exemplary form:

$A \rightarrow T$: $BCP_{RSP}(XOR(\mu_i, \mu_{i+1})\ PAD[X])$

[00143] As a result of the above response from Trent, Alice's public identifier N_i, secret white data key k_i, and large secret key K_i have been mutated. In addition, the "Mutated Mutating ID" command code μ_i has been mutated.

[00144] Using the above command codes and padding values, Alice can generate a "Get Encryption Key" request for Trent using the following exemplary form:

$A \rightarrow T$: $BCP_{REQ}(\alpha_i\ PAD[X])$

[00145] In response, Trent generates a response for Alice with the following exemplary form:

$A \rightarrow T$: $BCP_{RSP}(XOR(\alpha_i, \alpha_{i+1})\ N_{RSP}\ E(K_{FX2}(i, K_i), K_{RSP})\ PAD[X])$

WAVELYNX EX. 1003, Pg. 735
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

where K_{RSP} is a randomly-generated encryption key and N_{RSP} is a randomly generated identifier associated with the encryption key K_{RSP}. The encryption key K_{RSP} is encrypted with Alice's large secret key K_i or a portion thereof. For example, in some variants, the encryption key K_{RSP} is encrypted with the result of a key function K_{FX2}. The key function K_{FX2} can be different than the key function K_{FX} used in the other portions of the response from Trent and can include an irreversible function that generates a key based on the iteration count i and the large secret key K_i. The key function K_{FX} is irreversible since, as described below, an eavesdropper could discover the encryption key K_{RSP} if Alice uses the encryption key to encrypt white or discoverable data. Using an irreversible function limits or prevents an eavesdropper from determining Alice's large secret key K_i upon obtaining knowledge of the encryption key K_{RSP}. For example, the key function K_{FX2} can include a secure hash function that generates a hash of a portion of Alice's large secret key K_i generated by an extraction function F_{EX2}, different from the extraction function F_{EX} described above, that extracts a portion of Alice's large secret key K_i based on the iteration count i. The secure hash function included in the key function K_{FX2} can also generate a hash based on an initialization vector generated by an initialization function I_{FX2}, different from the initialization function described above. The initialization function I_{FX2} can generate an initialization vector based on the iteration count i and Alice's large secret key K_i.

[00146] Since the requested encryption key K_{RSP} is encrypted with Alice's large secret key K_i or a version thereof, the encrypted encryption key K_{RSP} does leak some information about Alice's large secret key K_i. Similar to the information leaked by the message authentication code or hash described above, an attacker could use information contained in the encrypted encryption key K_{RSP} to rule out candidate keys when performing a brute-force attack. The amount of information leaked, however, in the encrypted encryption key K_{RSP}, can be accounted for in the calculation of when secret keys should be reset in order to substantially reduce or eliminate the possibility of a successful brute force attack. In some embodiments, the iteration count i is also incremented an additional count before being used in the key function K_{FX2} in order to further limit information leaked in the response.

[00147] As shown above in the response to the "Get Encryption Key" request, the encryption key K_{RSP} is included as part of the response parameters T_{RSP}, which are considered white data and encrypted with Alice's white data key k_i. The encryption key K_{RSP} is considered white data since it is likely that Alice will use the encryption key K_{RSP} to encrypt

WAVELYNX EX. 1003, Pg. 736
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

white data. If the encryption key K_{RSP} is used to encrypt white data, the encryption key K_{RSP} could potentially be discovered by an eavesdropper using a brute force attack. If the encryption key K_{RSP} were encrypted as black data with Alice's large secret key K_i, an eavesdropper could use a discovered value of the encryption key to perform a brute force attack and determine information about Alice's large secret key K_i and other black data exchanged between Alice and Trent. As described above, encrypting the encryption key K_{RSP} with a key generated by an irreversible function limits the ability of an eavesdropper to discover information about Alice's large secret key K_i from obtained knowledge of the encryption key K_{SRP}. Therefore, by encrypting the encryption key K_{RSP} with Alice's white data key k_i, Alice's large secret key K_A and other black data exchanged between Alice and Trent are kept secure even if an eavesdropper is able to obtain the encryption key K_{RSP}.

[00148] Alice can use the above command codes and padding values to generate a "Get Decryption Key" request for Trent using the following exemplary form:

$$A \rightarrow T\text{: } BCP_{REQ}(N_{REQ}\ \delta_i\ PAD[X])$$

where N_{REQ} is an identifier associated with the requested decryption key.

[00149] In response, Trent generates a response for Alice with the following exemplary form:

$$A \rightarrow T\text{: } BCP_{RSP}(XOR(\delta_i,\ \delta_i')\ E(K_{FX2}(i,\ K_i),\ K_{RSP})\ PAD[X])$$

where K_{RSP} is the requested decryption key associated with the identifier N_{REQ} provided in the request. As described above with respect to the "Get Encryption Key" request, the encryption key K_{RSP} can be encrypted with the result of a key function K_{FX2}, different than the key function K_{FX} used in the other portions of the response from Trent, that includes an irreversible function that generates a key based on the iteration count i and large secret key K_i. As noted above, Trent can increment the iteration count i an additional count before using the count in the key function K_{FX2}.

[00150] As previously noted, entities can use the above request formats to request encryption and decryption keys from the authenticator. The entities use the encryption keys to encrypt data (e.g., digital content) before transmitting the data to other entities, and the entities receiving the encrypted data request the needed decryption keys from the

WAVELYNX EX. 1003, Pg. 737
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

authenticator. Using the authenticator as the key distributor allows the entities to communicate with each other without having to establish a separate encryption or session key between entities. Since neither entity is required to initiate a secure session with the other entity by sending plaintext or non-encrypted data to the other data (i.e., no handshaking is required), data transmitted between the entities can always be encrypted. Using this encryption strategy, the above request and response formats can be used to establish mutating transport layer security that can be applied to various applications, such as e-mail exchange, digital content management, secure file transfers, client-server sessions, etc.

MUTATING LOCK BOX

[00151] In some embodiments, the mutating IDs held by an entity, such as an identification mutating ID, a digital signature mutating ID, a key pair mutating ID, a fingerprint mutating ID, and/or a large mutating ID are stored in a secure memory device, referred to as a mutating lock box. The data stored in the mutating lock box can be encrypted to reduce the effectiveness of offline brute force attacks. In some embodiments, the data stored in the mutating lock box is separately encrypted, as described above, such that black data and white data are not encrypted with the same keys. Activating or opening the mutating lock box can require multiple steps or factors. For example, an entity attempting to activate the mutating lock box may be required to provide a user password and/or a PIN. The authenticator device 28 may also perform multiple checks and analyses, such as performing a box identifier currency check and analyzing available system data. In some embodiments, the mutating lock box is stored in a hard drive or permanent memory device of an entity. In other embodiments, the mutating lock box is stored in a portable memory device, such as a universal serial bus ("USB") Flash drive.

[00152] Various features of the invention are set forth in the following claims.

WAVELYNX EX. 1003, Pg. 738
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

CLAIMS

1. A method of encrypting data, the method comprising:

establishing first data including first discoverable data and first undiscoverable data;

establishing a first key and a second key, the second key being substantially underivable from the first key;

encrypting the first discoverable data with the first key; and

encrypting the first undiscoverable data with the second key

2. The method of claim 1 further comprising mutating at least one of the first key and the second key, establishing second data including second discoverable data and second undiscoverable data, encrypting the second discoverable data with the first key, and encrypting the second undiscoverable data with the second key.

3. The method of claim 1 further comprising establishing a third key.

4. The method of claim 3 wherein establishing a third key includes establishing a third key based on at least one of the first key and the second key.

5. The method of claim 3 wherein establishing a third key includes establishing a third key by performing an exclusive OR of a fourth key and at least one of the first key and the second key.

6. The method of claim 3 further comprising encrypting at least a portion of the discoverable data with the third key.

7. The method of claim 6 further comprising mutating the third key.

8. The method of claim 1 wherein encrypting the discoverable data with a first key includes encrypting the discoverable data and a result of encrypting the undiscoverable data with the second key with the first key.

WAVELYNX EX. 1003, Pg. 739
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

9. The method of claim 1 wherein encrypting the undiscoverable data with a second key includes encrypting the undiscoverable data and a result of encrypting the discoverable data with the first key with the second key.

10. The method of claim 1 further comprising establishing an identifier.

11. The method of claim 10 further comprising appending the identifier to at least one of a result of encrypting the discoverable data with the first key and a result of encrypting the undiscoverable data with the second key.

12. The method of claim 11 further comprising mutating the identifier.

13. The method of claim 1 further comprising generating a hash of at least a portion of the data.

14. The method of claim 13 further comprising encrypting the hash with the second key.

15. A system for encrypting data including discoverable data and undiscoverable data, the system comprising:

a sender having a first key and a second key, the second key being substantially underivable from the first key, the sender configured to encrypt the discoverable data with the first key, to encrypt the undiscoverable data with the second key, and to receive at least one of a mutated first key and a mutated second key.

16. The system of claim 15 further comprising an authenticator configured to establish the first key and the second key.

17. The system of claim 16 wherein the authenticator is further configured to establish the mutated first key and the mutated second key.

18. The system of claim 15 wherein the sender further has a third key.

19. The system of claim 18 wherein the third key is based on at least one of the first key and the second key.

20. The system of claim 18 wherein the third key is based on the result of an exclusive OR of a fourth key and at least one of the first key and the second key.

WAVELYNX EX. 1003, Pg. 740
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

21. The system of claim 18 wherein the sender is further configured to encrypt at least a portion of the discoverable data with the third key.

22. The system of claim 21 wherein the sender is further configured to receive a mutated third key.

23. The system of claim 15 wherein the sender is further configured to encrypt the discoverable data and a result of encrypting the undiscoverable data with a second key with the first key.

24. The system of claim 15 wherein the sender is further configured to encrypt the undiscoverable data and a result of encrypting the discoverable data with the first key with the second key.

25. The system of claim 15 wherein the sender further has an identifier.

26. The system of claim 25 wherein the sender is further configured to append the identifier to at least one of a result of encrypting the discoverable data with the first key and a result of encrypting the undiscoverable data with the second key.

27. The system of claim 26 wherein the sender is further configured to receive a mutated identifier.

28. The system of claim 15 wherein the sender is further configured to generate a hash of at least a portion of the discoverable data.

29. The system of claim 28 wherein the sender is further configured to encrypt the hash with the second key.

30. A storage device for storing data, the device configured to encrypt discoverable data with a first set of keys and undiscoverable data with a second set of keys, the second set of keys being substantially underivable from the first set of keys.

31. The storage device of claim 30 further configured to communicate with an authenticator to receive a mutation of at least one of the first set of keys or the second set of keys.

WAVELYNX EX. 1003, Pg. 741
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

32. The storage device of claim 30 further configured to require authorization before activating.

33. The storage device of claim 32 wherein the authorization includes at least one of a password and a personal identification number.

34. A method of transmitting messages between an entity and an authenticator comprising:

establishing an identifier associated with the entity;

establishing a first key to encrypt only undiscoverable data, the first key known only to the entity and the authenticator;

establishing a second key to encrypt only discoverable data, the second key known only to the entity and the authenticator;

encrypting request parameters with the second key to create an encrypted request;

generating a hash of the request parameters;

encrypting the hash of the request parameters with the first key to create an encrypted request hash;

the entity sending the encrypted request and the encrypted request hash to the authenticator; and

mutating at least one of the first key and the second key.

35. The method of claim 34 wherein generating a hash of the request parameters includes generating a hash of the request parameters and the identifier.

36. The method of claim 35 further comprising appending the identifier to at least one of the encrypted message and the encrypted message hash.

37. The method of claim 36 further comprising establishing a key function that extracts a portion of the first key based on the identifier.

WAVELYNX EX. 1003, Pg. 742
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

38. The method of claim 37 wherein encrypting the hash of the request parameters with the first key includes encrypting the hash of the request parameters with a result of the key function.

39. The method of claim 34 further comprising establishing an initialization vector based on the first key and the identifier.

40. The method of claim 39 wherein generating a hash of the request parameters includes generating a hash of the request parameters based on the initialization vector.

41. The method of claim 34 further comprising establishing an extraction function based on the first key and the identifier.

42. The method of claim 41 wherein generating a hash of the request parameters includes generating a hash of the request parameters and a result of the extraction function.

43. The method of claim 34 further comprising mutating the identifier to create a mutated identifier and sending the mutated identifier to the entity.

44. The method of claim 43 further comprising mutating the second key to create a mutated second key and encrypting the mutated second key and response parameters with the second key to create an encrypted response.

45. The method of claim 44 further comprising the generating a random mutation value.

46. The method of claim 45 wherein mutating at least one of the first key and the second key includes mutating at least one of the first key and the second key using the mutation value.

47. The method of claim 46 further comprising generating a hash of the mutated identifier, the mutated second key, the mutation value, and the response parameters.

48. The method of claim 47 wherein generating a hash of the mutated identifier, the mutated second key, the mutation value, and the response parameters includes generating a hash of the mutated identifier, the mutated second key, the mutation value, and the response parameters based on an initialization vector.

WAVELYNX EX. 1003, Pg. 743
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

49. The method of claim 48 wherein generating a hash of the mutated identifier, the mutated second key, the mutation value, and the response parameters includes generating a hash of the mutated identifier, the mutated second key, the mutation value, the response parameters, and a result of the extraction function.

50. The method of claim 49 further comprising encrypting the mutation value and the hash of the mutated identifier, the mutated second key, the mutation value, and the response parameters with the first key to create an encrypted response hash.

51. The method of claim 50 wherein encrypting the mutation value and the hash of the mutated identifier, the mutated second key, the mutation value, and the response parameters with the first key includes encrypting the mutation value and the hash of the mutated identifier, the mutated second key, the mutation value, and the response parameters with a result of a key function.

52. The method of claim 51 further comprising appending the mutated identifier to at least one of the encrypted response and the encrypted response hash.

53. The method of claim 52 further comprising sending the encrypted response and the encrypted response hash to the entity.

WAVELYNX EX. 1003, Pg. 744
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

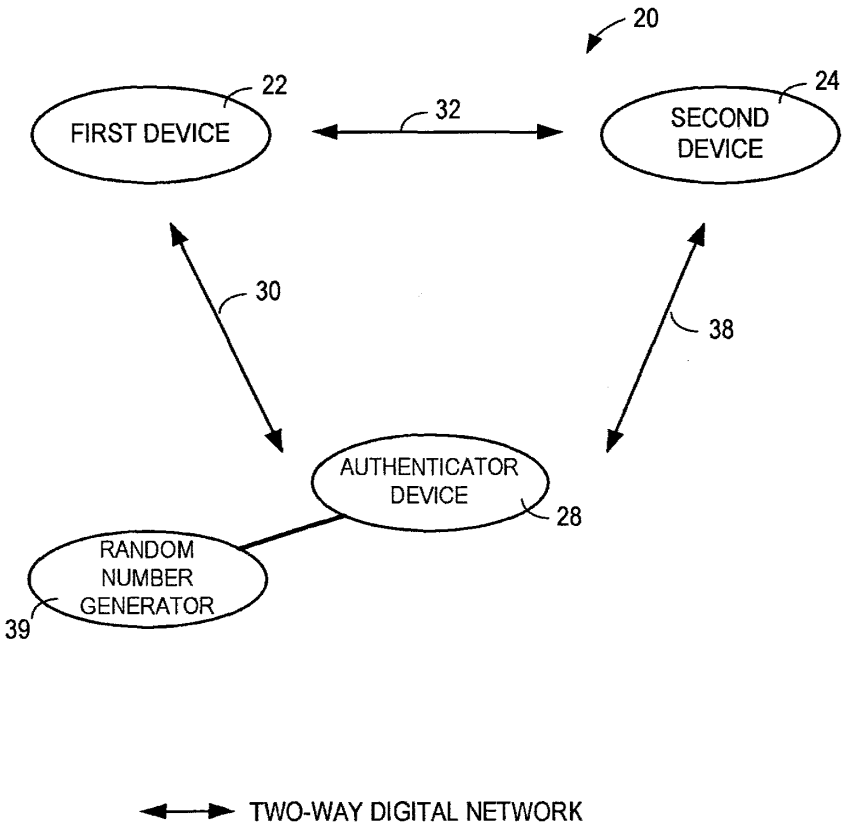
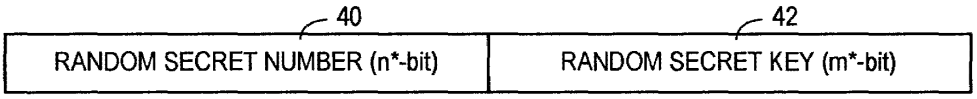

20
22
FIRST DEVICE
32
24
SECOND DEVICE
30
38
AUTHENTICATOR DEVICE
28
RANDOM NUMBER GENERATOR
39
TWO-WAY DIGITAL NETWORK

FIG. 1

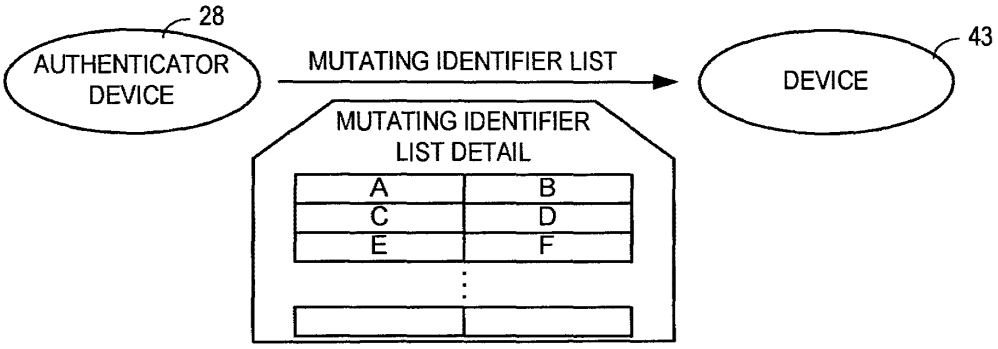


* IN ONE IMPLEMENTATION n AND m ARE BOTH 256 BITS, IN ANOTHER IMPLEMENTATION m IS MUCH LARGER THAN n

FIG. 2

TWO EXEMPLARY METHODS OF PACKAGING MUTATING IDENTIFIERS.

1) THE AUTHENTICATOR DEVICE GENERATES A LIST OF MUTATING IDENTIFIERS



THE LIST MAY BE DISTRIBUTED EITHER BY HAND, PUBLIC KEY ENCRYPTION, OR ENCODING WITH A MUTATING IDENTIFIER FROM A PREVIOUSLY DISTRIBUTED LIST.

FIG. 3A

2) AUTHENTICATOR DEVICE GENERATES A SINGLE MUTATING IDENTIFIER

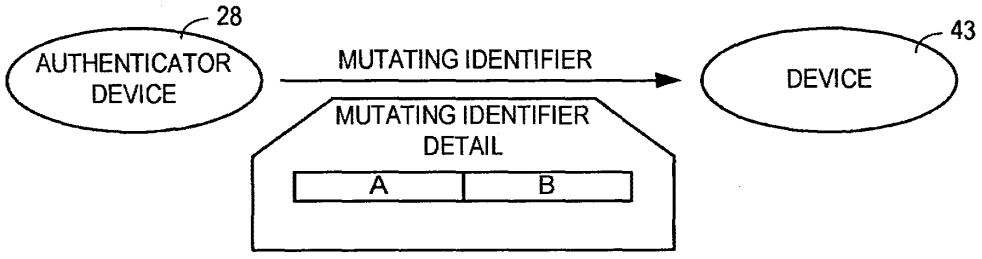


FIG. 3B

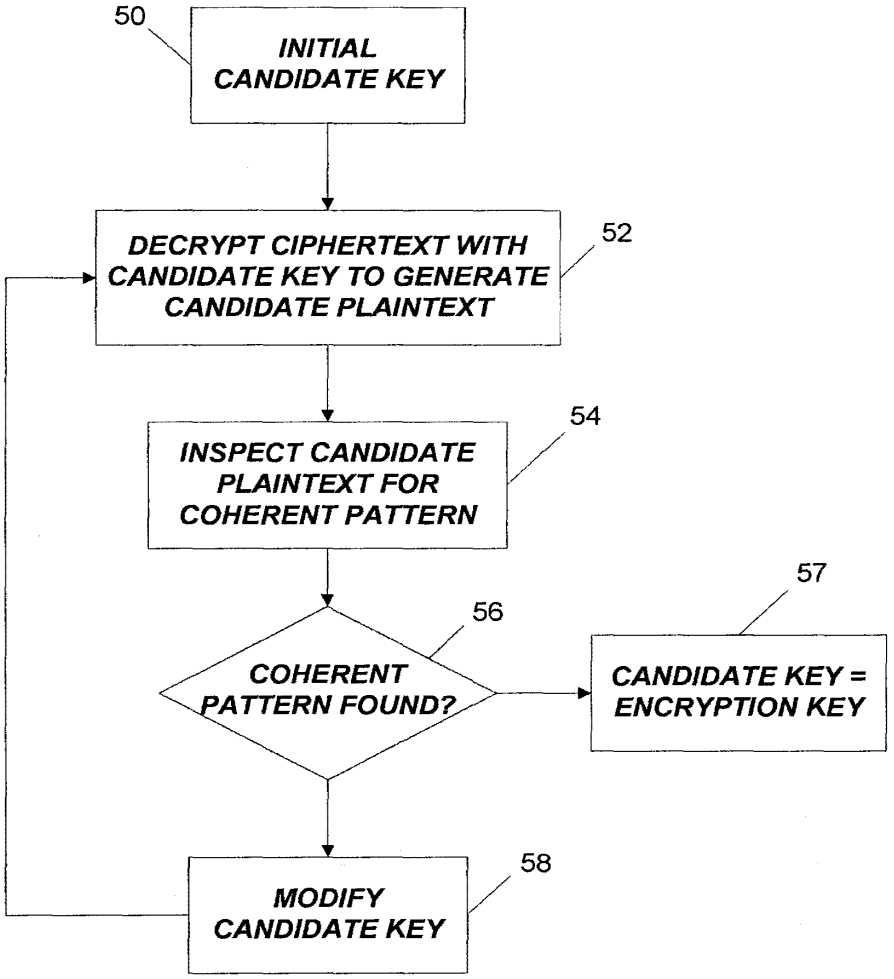

50
INITIAL CANDIDATE KEY
DECRYPT CIPHERTEXT WITH CANDIDATE KEY TO GENERATE CANDIDATE PLAINTEXT
52
INSPECT CANDIDATE PLAINTEXT FOR COHERENT PATTERN
54
COHERENT PATTERN FOUND?
56
CANDIDATE KEY = ENCRYPTION KEY
57
MODIFY CANDIDATE KEY
58

FIG. 4

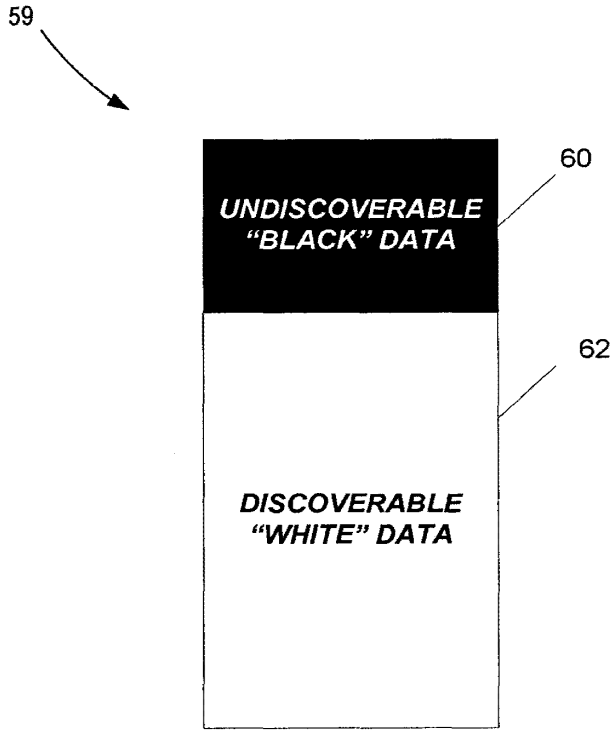

59
60
UNDISCOVERABLE
"BLACK" DATA
62
DISCOVERABLE
"WHITE" DATA

FIG. 5

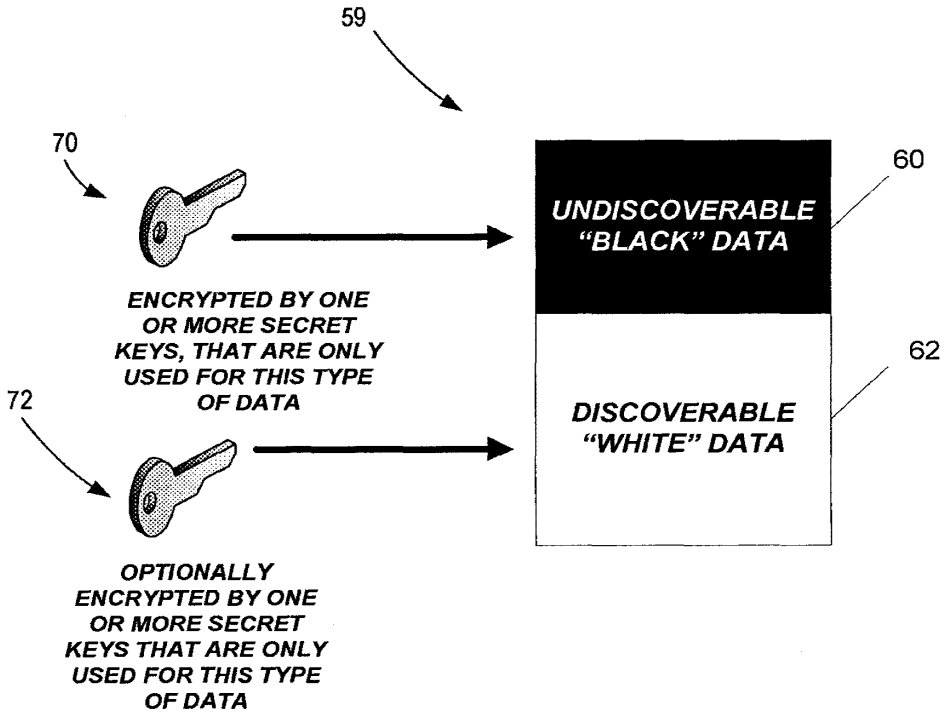

59
70
60
UNDISCOVERABLE "BLACK" DATA
ENCRYPTED BY ONE OR MORE SECRET KEYS, THAT ARE ONLY USED FOR THIS TYPE OF DATA
62
72
DISCOVERABLE "WHITE" DATA
OPTIONALLY ENCRYPTED BY ONE OR MORE SECRET KEYS THAT ARE ONLY USED FOR THIS TYPE OF DATA

FIG. 6

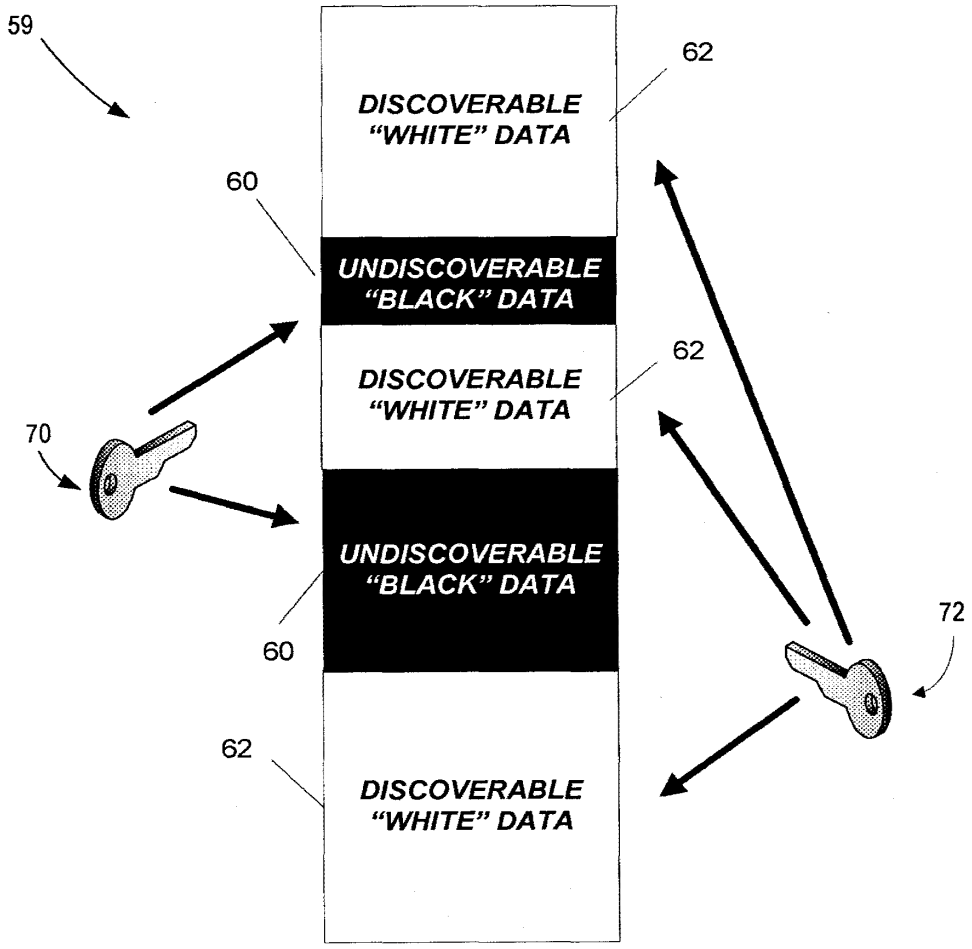

59
62
DISCOVERABLE "WHITE" DATA
60
UNDISCOVERABLE "BLACK" DATA
62
DISCOVERABLE "WHITE" DATA
70
UNDISCOVERABLE "BLACK" DATA
72
60
62
DISCOVERABLE "WHITE" DATA

FIG. 7

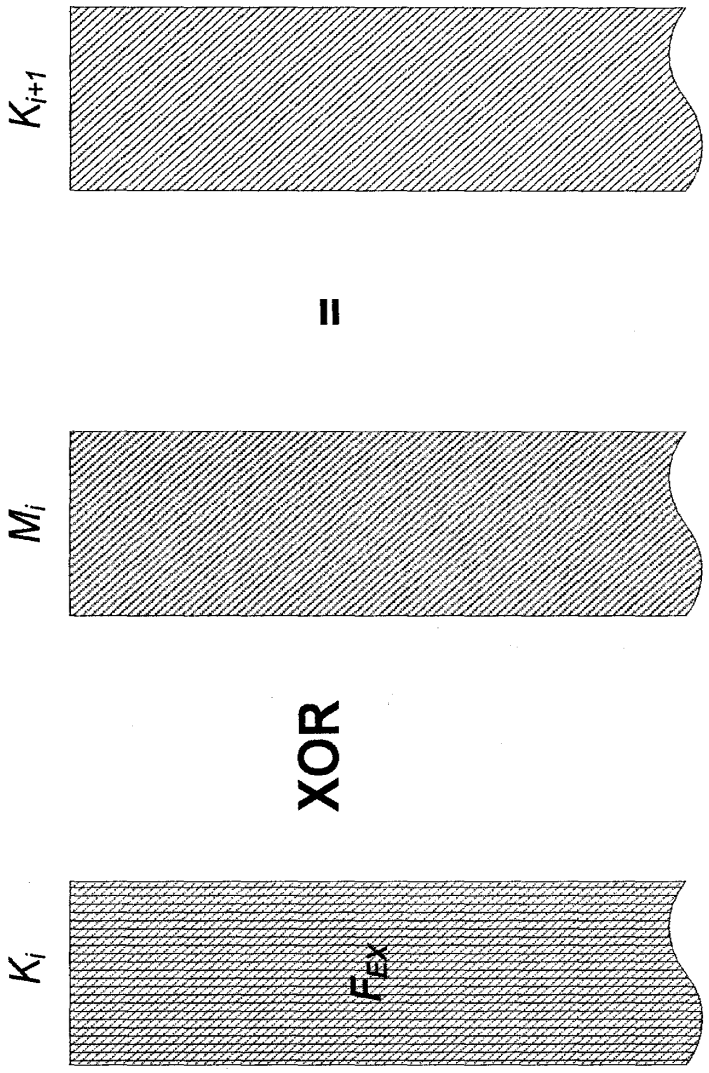


FIG. 8

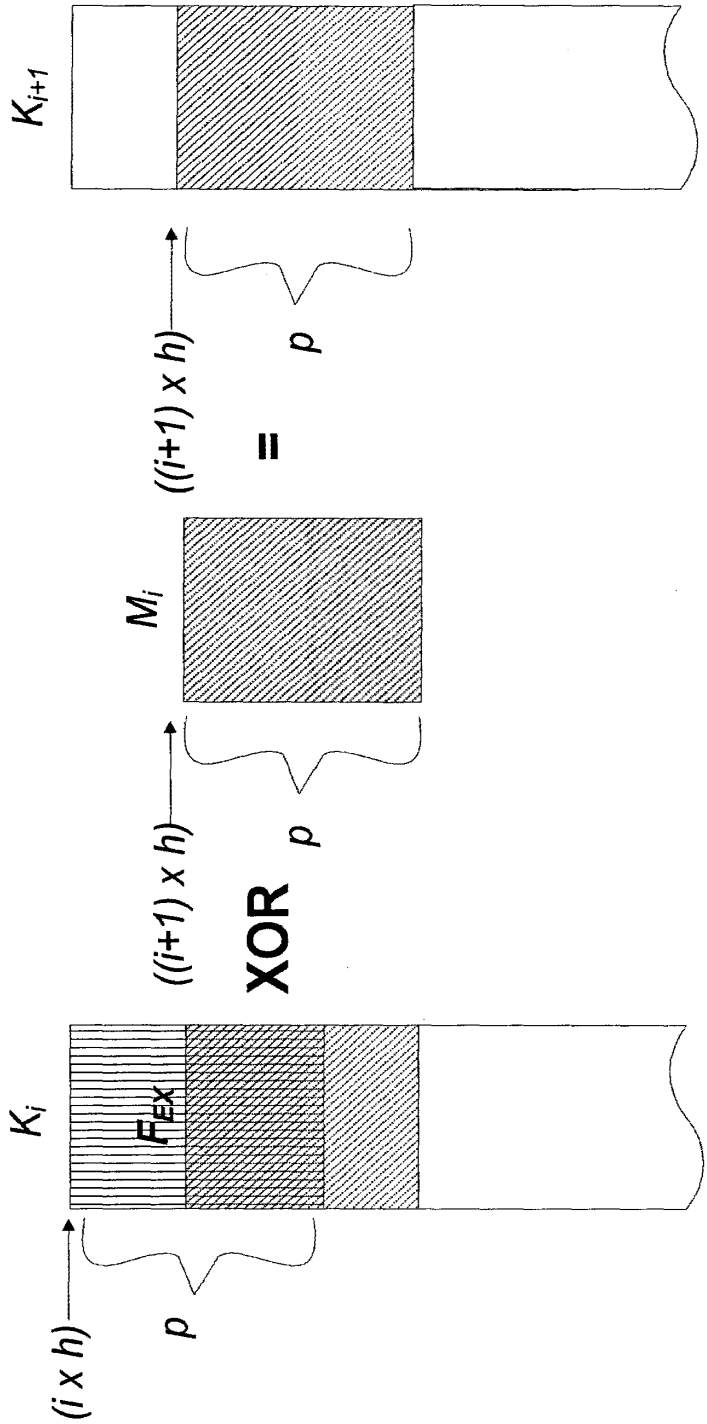


FIG. 9

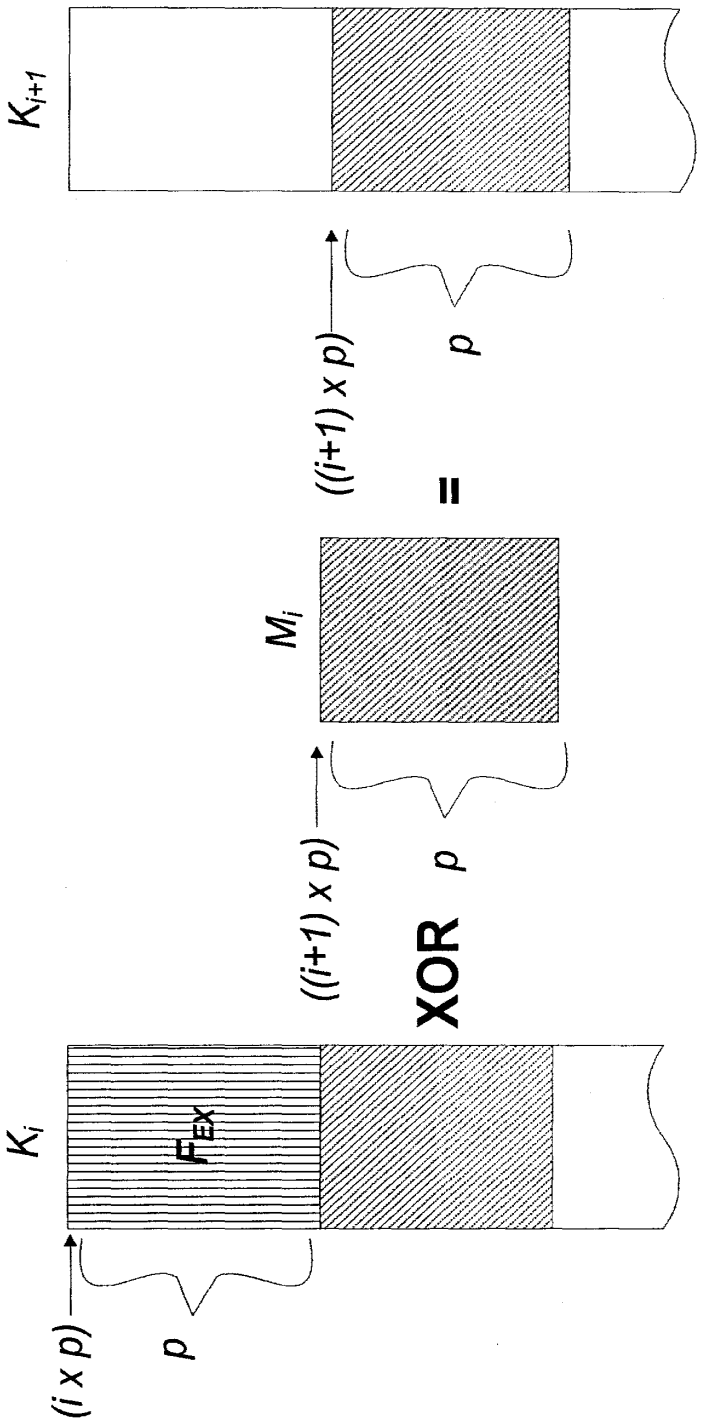


FIG. 10

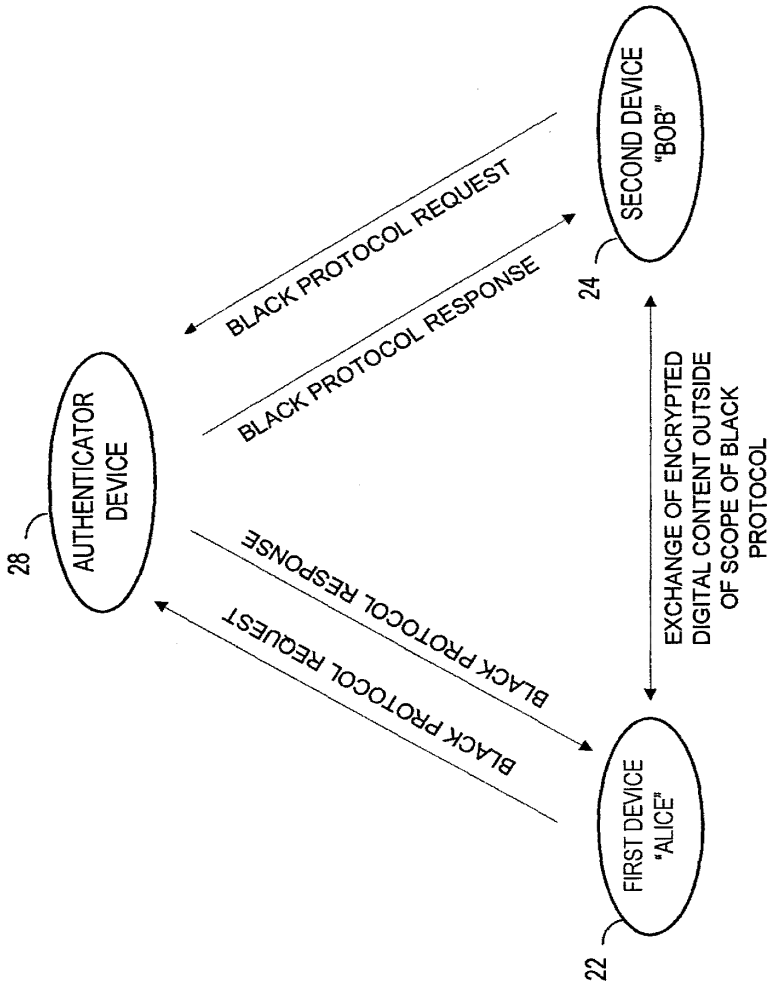

SECOND DEVICE "BOB"
24
BLACK PROTOCOL REQUEST
BLACK PROTOCOL RESPONSE
AUTHENTICATOR DEVICE
28
BLACK PROTOCOL RESPONSE
BLACK PROTOCOL REQUEST
FIRST DEVICE "ALICE"
22
EXCHANGE OF ENCRYPTED DIGITAL CONTENT OUTSIDE OF SCOPE OF BLACK PROTOCOL

FIG. 11

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 October 2007 (18.10.2007)

PCT

(10) International Publication Number
WO 2007/117315 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/US2006/061683

(22) International Filing Date:
6 December 2006 (06.12.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/303,181 15 December 2005 (15.12.2005) US

(71) Applicant *(for all designated States except US)*: **SYMBOL TECHNOLOGIES, INC.** [US/US]; One Symbol Plaza, Holtsville, New York 11742 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **CABANA, Joe** [US/US]; 5 Vernon Lane, Centereach, New York 11720 (US). **PAUL, Christopher** [US/US]; 20 Folger Street, Bayport, New York 11705 (US). **CORDES, Kevin** [US/US]; 1 Liberty Lane, Miller Place, New York 11764 (US).

(74) Agents: **LORENZ, Timothy J.** et al.; Ingrassia, Fisher & Lorenz, P.c., 7150 E. Camelback Road, Suite 325, Scottsdale, AZ 85251 (US).

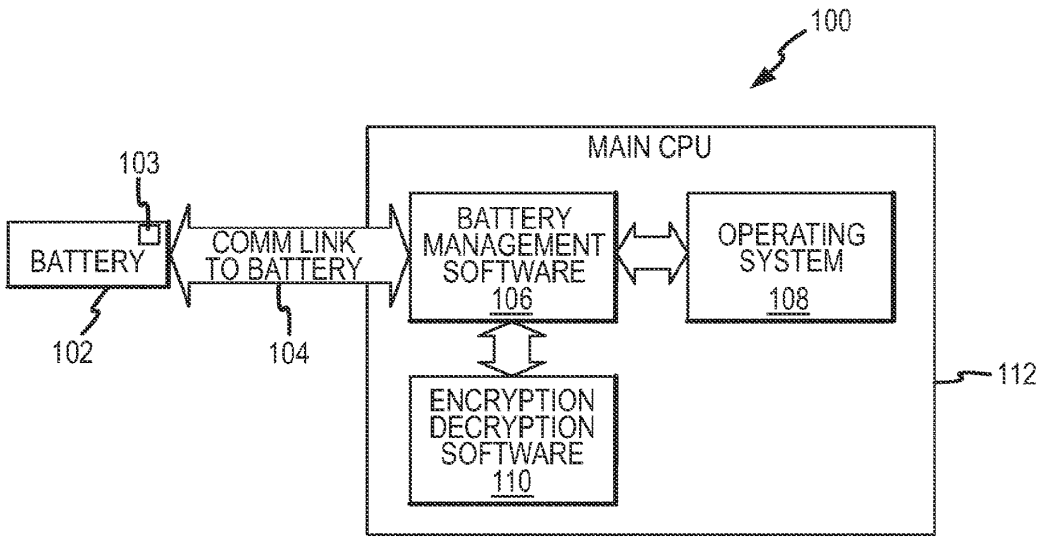
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) **Title:** METHODS AND APPARATUS FOR POWER SOURCE AUTHENTICATION



(57) **Abstract:** A power source management system disposed within a device is configured to communicate with a power source and perform an authentication procedure to determine whether the power source is an approved power source. The authentication procedure may involve the use of a challenge-response procedure using keys resident in the device and the power source. If it is determined that the power source is not an approved power source, the power source management system prevents use of or disables certain functionality of the device.

WO 2007/117315 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHODS AND APPARATUS FOR POWER SOURCE AUTHENTICATION

TECHNICAL FIELD

[0001] The present invention generally relates to the use of batteries and other such power sources and, more particularly, to systems and methods for preventing the use of unapproved power sources.

BACKGROUND

[0002] Many devices, particularly those with critical power requirements, are designed to use a specific power source (e.g., batteries) manufactured by a designated, authorized battery manufacturer. Nevertheless, other parties may manufacture and market what are known as "cloned" batteries that may fit the intended device, but which are not authorized for use with that device.

[0003] The use of such cloned batteries poses two problems. First, when a cloned battery is used, and that battery does not work properly, this failure reflects poorly on the party that manufacturers the device in which the battery is used, even though that party does not control the quality of the cloned battery. Second, cloned batteries are undesirable because they may incorporate inferior materials, might not be manufactured to the same standards, and/or might not be subject to the same level of quality control. Furthermore, cloned batteries are generally not submitted for UL (Underwriters Laboratories) approval, as any economically practical design submitted for UL approval requires submission of all chargers designed for use with that battery, which would be impracticable. The lack of UL approval has significant insurance implications for the end-customer.

[0004] Accordingly, for these and other reasons, there is a need for systems and methods that help prevent the use of unapproved power sources.

BRIEF SUMMARY

[0005] In accordance with one embodiment of the present invention, a battery management system (or "power source management system") within a device is configured to communicate with a power source and perform an authentication procedure wherein the battery management system determines whether the power source is an approved power source. Approved power sources are preconfigured with an appropriate cryptographic key or keys. If the system determines that the power source is not an approved power source, it disables or otherwise limits use of the device. The

WAVELYNX EX. 1003, Pg. 758
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

authentication procedure involves, in the illustrated embodiment, the use of a challenge-response scheme, wherein the device generates a random number, sends it to the power source for encryption and response, then compares that response with a response generated by the device itself using the same device-resident key. In this way, the present invention helps prevent the use of unapproved batteries and/or power sources.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] A more complete understanding of the present invention may be derived by referring to the detailed description and claims when considered in conjunction with the following figures, wherein like reference numbers refer to similar elements throughout the figures.

[0007] FIG. 1 is a battery authentication system in accordance with one embodiment of the present invention;

[0008] FIG. 2 is a battery authentication system in accordance with another embodiment of the present invention;

[0009] FIG. 3 is a battery authentication system in accordance with another embodiment of the present invention;

[0010] FIG. 4 is a battery authentication system in accordance with another embodiment of the present invention;

[0011] FIG. 5 is a battery authentication system in accordance with another embodiment of the present invention;

[0012] FIG. 6 is a battery authentication system in accordance with one embodiment of the present invention; and

[0013] FIG. 7 is flowchart depicting an authentication procedure in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION

[0014] The following detailed description is merely illustrative in nature and is not intended to limit the invention or the application and uses of the invention. Furthermore, there is no intention to be bound by any expressed or implied theory presented in the preceding technical field, background, brief summary or the following detailed description. In addition, the invention may be described in terms of functional and/or logical block components and various processing steps. It should be appreciated that such

WAVELYNX EX. 1003, Pg. 759
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

block components may be realized by any number of hardware, software, and/or firmware components configured to perform the specified functions.

[0015] In general, a power source management system in accordance with the present invention is configured to communicate with a power source and perform an authentication procedure wherein the power source management system determines whether the power source is an approved power source – i.e., a power source preconfigured with an appropriate cryptographic key or keys. In this regard, the terms "battery" and "power source" are often used interchangeably, and thus the term "battery" is not intended as a limitation of the present invention. If the system determines that the battery is not an approved power source, it disables or otherwise limits use of the device. The authentication procedure involves, in the illustrated embodiment, the use of a challenge-response scheme, wherein the device generates a random number, sends it to the battery for encryption and response, then compares that response with a response generated by the device itself using a device-resident key. The invention is not limited to challenge/response authentication, however, and encompasses any convenient authentication procedure.

[0016] Referring to FIG. 1, a battery authentication system in accordance with one embodiment of the present invention generally includes a CPU 112, a battery management system 106, an operating system 108, and an encryption/decryption system 110. Battery management system 106 communicates with a battery logic subsystem 103 of battery 102 via a communication link 104, and is configured to communicate with a battery and perform an authentication procedure wherein the battery management system determines whether the battery is an approved power source. Approved power sources are preferably pre-configured with an appropriate cryptographic key or keys (e.g., within battery logic subsystem 103). If the system determines that the battery is not an approved power source, it disables or otherwise limits use of the device.

[0017] Battery 102 comprises any power source now known or later developed, including various primary and secondary batteries, fuel cells, and any other portable power source. It also includes various non-portable power sources such as AC adaptors, power supplies, etc. Battery logic subsystem 103 disposed within battery 102 includes suitable hardware and/or software (e.g., microcontrollers, memory devices, etc.) capable of performing the functions described herein.

[0018] CPU 112 refers to the CPU of a computer, terminal, portable bar-code reader, personal data assistant, cellular telephone, or any other type of device. Software

WAVELYNX EX. 1003, Pg. 760
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

implementations executed by CPU 112 (or a separate power management processor, as described below) may be implemented using any suitable software code now known or later developed, including, for example, assembly language, C, or the like.

[0019] Communication link 104 includes any convenient connection between the battery and CPU, and may implement any suitable protocol or communication method. In one embodiment, communication takes place through the terminal electrodes. Suitable connection systems include, for example, I2C, Dallas 1 Wire, and SMBUS.

[0020] Having thus given an overview of one embodiment of the present invention, an exemplary method for battery authorization will now be described. As shown in FIG. 7, the system begins by initiating battery authorization (702). Initiation may take place, for example, when the battery is first inserted, or may take place at any arbitrary or scheduled time during which the battery is connected to the system.

[0021] Next, in step 704, the terminal (i.e., battery management software 106 implemented within a computer terminal or the like) generates a random number. The length of this random number may be selected depending upon the desired level of security in accordance with known cryptographic principles. The generated random number is then transmitted from the terminal to the battery as a challenge (step 706).

[0022] The battery encrypts the received random number (step 708) and sends the encrypted random number back to the terminal as a response to the challenge (step 710). This encryption may be private key, public/private key, or any other encryption algorithm.

[0023] The terminal then compares the response generated by the battery with the expected response (step 712). That is, the terminal performs its own encryption operation on the generated random number, and it is this number that the terminal compares with the battery's response. In one embodiment, the SHA-1/HMAC algorithm as employed in the Texas Instruments BQ26100 Integrated Circuit is used for encryption.

[0024] If the battery's response matches the expected response, then the battery is successfully authenticated and normal operation of the terminal can proceed (step 716). If, on the other hand, the battery is not successfully authenticated, then use of the terminal is disabled or otherwise limited (step 718). For example, the terminal may stop reporting battery charge level, disable recharging of the battery, disable use of certain functionality, reduce input/output capabilities, and/or disable use of the display.

WAVELYNX EX. 1003, Pg. 761
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0025] Further details regarding authentication protocols and other such matters may be found in a number of standard reference books, including, for example, Bruce Schneier, Applied Cryptography, 2d ed. (1996).

[0026] In an alternate embodiment, a dedicated power management system is used. That is, referring to FIG. 1, the challenge-response of the illustrated embodiment is generated by the main CPU of the terminal. In FIG. 2, however, the challenge and response is generated by a dedicated power management processor 202. Power management processor 202 includes encryption/decryption software 110 and battery management software 106, which communicates with main CPU 112 through a second communication link 203.

[0027] In another embodiment, shown in FIG. 3, a separate secure IC 302 generates the challenge and checks the response via an interface with battery management software 106. A combination of these two embodiments is shown in FIG. 4, in which a separate encryption IC is used in connection with a dedicated power management processor 202.

[0028] When a separate IC 302 is used to handle the encryption/decryption process (as in FIGS. 3 and 4), the system can be made more secure by allowing IC 302 to monitor the communication between battery 102 and the main CPU 112 or power management processor 202. The IC can then interrupt battery communications if it detects that the system is using a non-approved battery. Such embodiments are shown in FIGS. 5 and 6.

[0029] While at least one example embodiment has been presented in the foregoing detailed description, it should be appreciated that a vast number of variations exist. It should also be appreciated that the example embodiment or embodiments described herein are not intended to limit the scope, applicability, or configuration of the invention in any way. Rather, the foregoing detailed description will provide those skilled in the art with a convenient road map for implementing the described embodiment or embodiments. It should be understood that various changes can be made in the function and arrangement of elements without departing from the scope of the invention as set forth in the appended claims and the legal equivalents thereof.

WAVELYNX EX. 1003, Pg. 762
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

CLAIMS

What is claimed is:

1. A method of preventing the use of an inappropriate power source in conjunction with a device, said method comprising the steps of:

performing an authentication procedure wherein the device determines whether the power source is an approved power source;

reducing a functionality of the device in the event that said authentication procedure determines that the power source is not an approved power source.

2. The method of claim 1, wherein said step of reducing said functionality includes disabling operation of the device.

3. The method of claim 1, wherein said step of reducing said functionality includes limiting operation of the device.

4. The method of claim 1, wherein said authentication procedure includes a challenge-response procedure.

5. The method of claim 1, wherein said challenge-response procedure includes the steps of:

generating, at the device, a random number;

transmitting said random number to the power source;

encrypting, at the power source, said random number to produce a first encrypted value;

encrypting, at the device, said random number to produce a second encrypted value ;

transmitting said first encrypted value to the device;

comparing, at the device, said first and second encrypted values.

6. The method of claim 5, wherein said first encrypted value is encrypted based on a key resident within said power source.

WAVELYNX EX. 1003, Pg. 763
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

7. The method of claim 5, wherein said second encrypted value is encrypted based on a key resident within said device.

8. A power source management system disposed within a device, said power source management system configured to communicate with a power source and perform an authentication procedure wherein the power source management system determines whether said power source is an approved power source and prevents full operation of said device in the event that said authentication procedure determines that the power source is not an approved power source.

9. The system of claim 8, wherein said authentication procedure includes a challenge-response procedure.

10. The system of claim 9, wherein said challenge-response procedure includes the steps of:

generating a random number;
transmitting said random number to said power source;
receiving, from said power source, a first encrypted value;
encrypting said random number to produce a second encrypted value; and
comparing said first and second encrypted values.

11. The system of claim 10, wherein said second encrypted value is based on a key resident within said device.

12. The system of claim 8, wherein said power source management system is disposed within a CPU of said device.

13. The system of claim 8, wherein said power source management system is disposed within a power management processor that is external to a CPU of said device.

14. The system of claim 8, wherein the power source management system communicates with an external encryption IC.

WAVELYNX EX. 1003, Pg. 764
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

15. The system of claim 14, wherein said external encryption IC monitors communication from said power source to said power source management system.

16. A power source comprising:

a processor;

a memory coupled to said processor;

a communication link coupled to said processor, wherein said processor is configured to receive a challenge number via said communication link, encrypt said challenge number utilizing a key resident in said memory to form a response, and transmit said response via said communication link.

WAVELYNX EX. 1003, Pg. 765
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

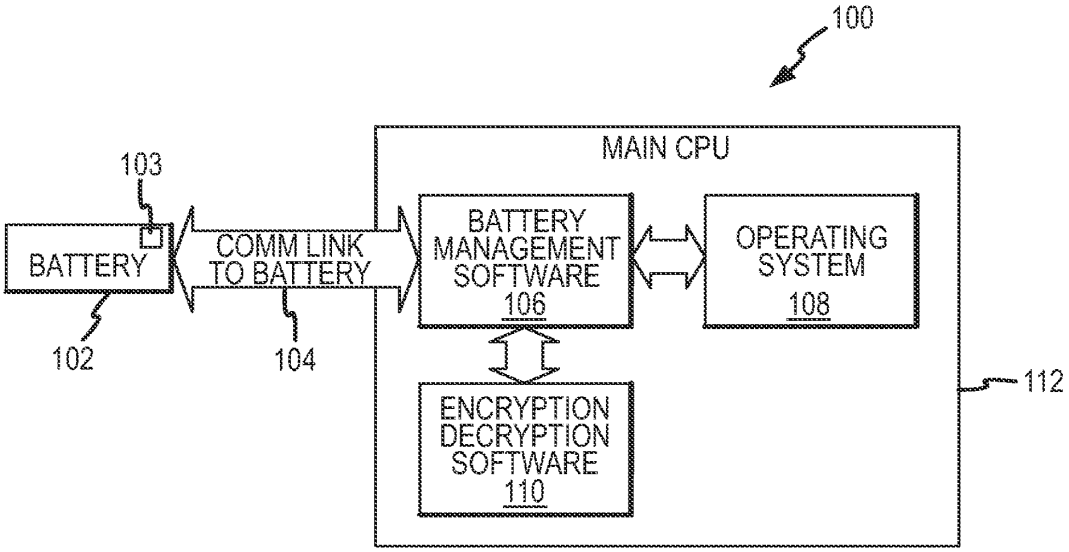

100
MAIN CPU
103
BATTERY
COMM LINK TO BATTERY
BATTERY MANAGEMENT SOFTWARE 106
OPERATING SYSTEM 108
102
104
ENCRYPTION DECRYPTION SOFTWARE 110
112

FIG.1

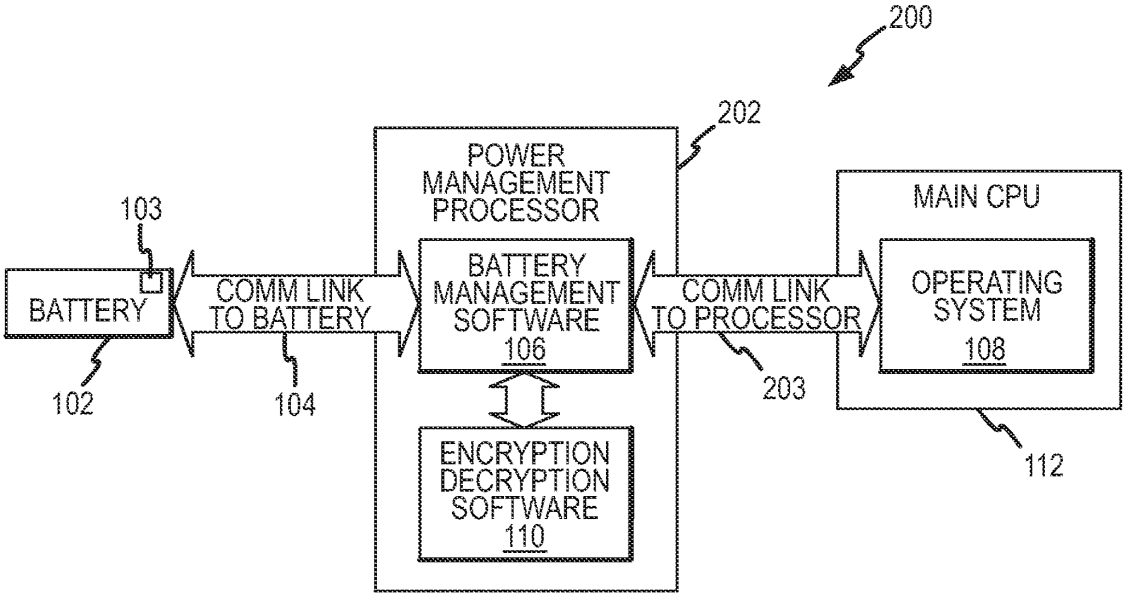

200
202
POWER MANAGEMENT PROCESSOR
MAIN CPU
103
BATTERY
COMM LINK TO BATTERY
BATTERY MANAGEMENT SOFTWARE 106
COMM LINK TO PROCESSOR
OPERATING SYSTEM 108
102
104
203
ENCRYPTION DECRYPTION SOFTWARE 110
112

FIG.2

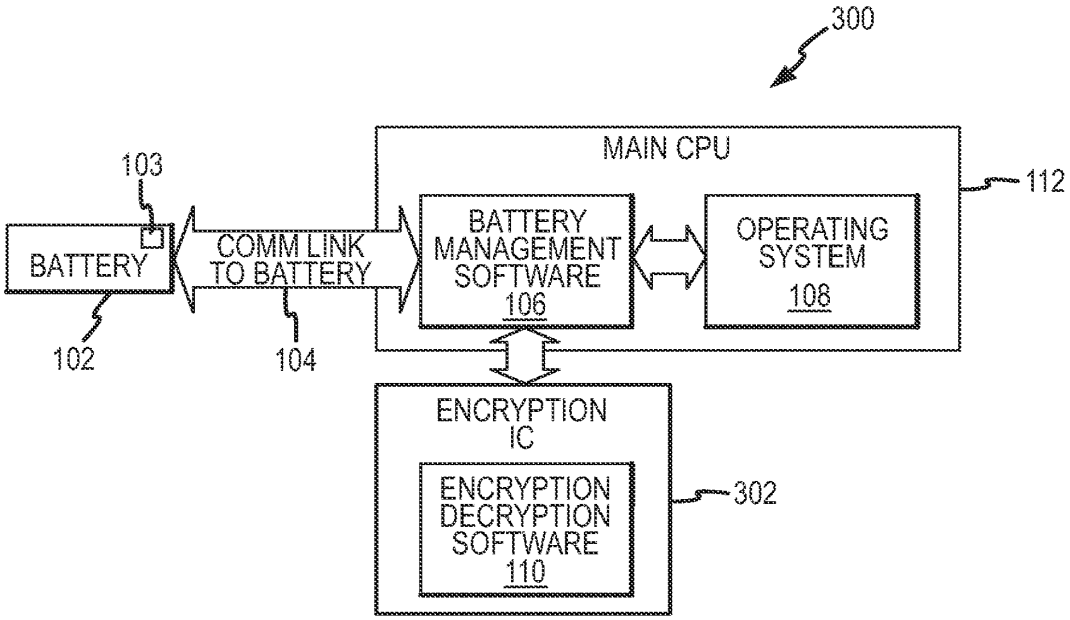

300
MAIN CPU
112
103
BATTERY
COMM LINK TO BATTERY
BATTERY MANAGEMENT SOFTWARE 106
OPERATING SYSTEM 108
102
104
ENCRYPTION IC
ENCRYPTION DECRYPTION SOFTWARE 110
302

FIG.3

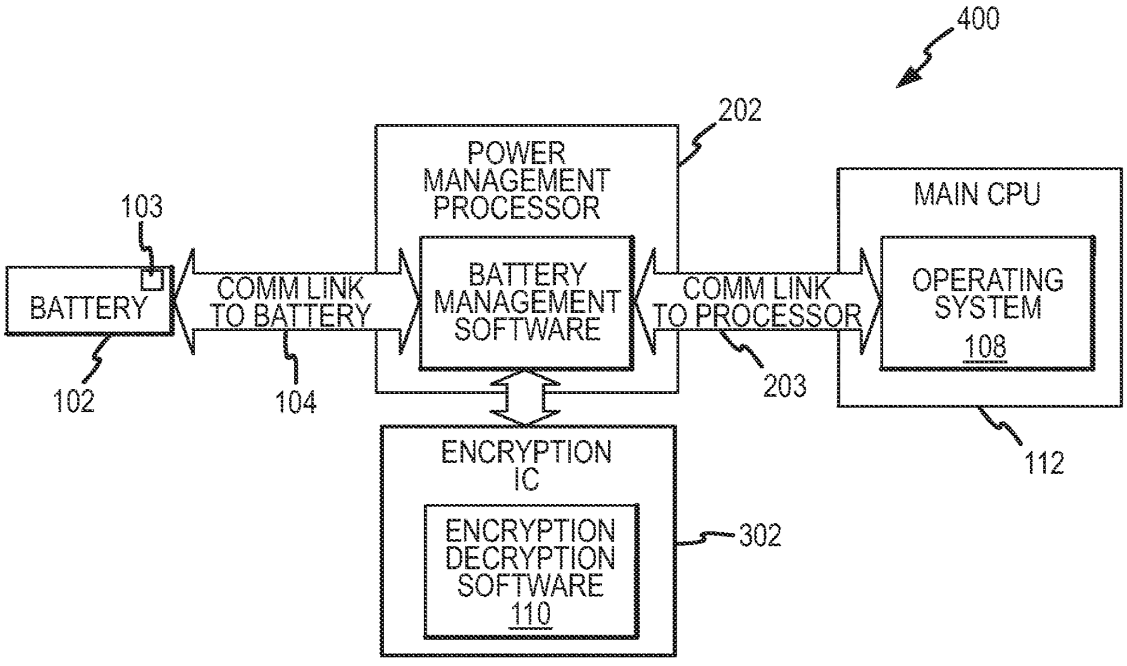

400
202
POWER MANAGEMENT PROCESSOR
MAIN CPU
103
BATTERY
COMM LINK TO BATTERY
BATTERY MANAGEMENT SOFTWARE
COMM LINK TO PROCESSOR
OPERATING SYSTEM 108
102
104
203
112
ENCRYPTION IC
ENCRYPTION DECRYPTION SOFTWARE 110
302

FIG.4

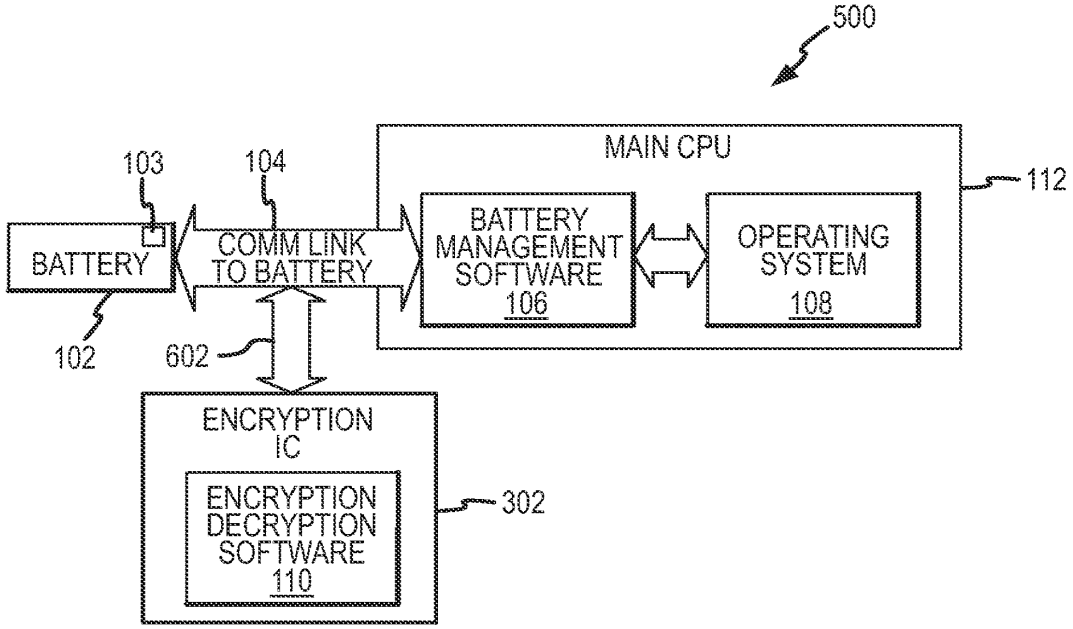

500
MAIN CPU
112
103
104
BATTERY
COMM LINK TO BATTERY
BATTERY MANAGEMENT SOFTWARE 106
OPERATING SYSTEM 108
102
602
ENCRYPTION IC
ENCRYPTION DECRYPTION SOFTWARE 110
302

FIG.5

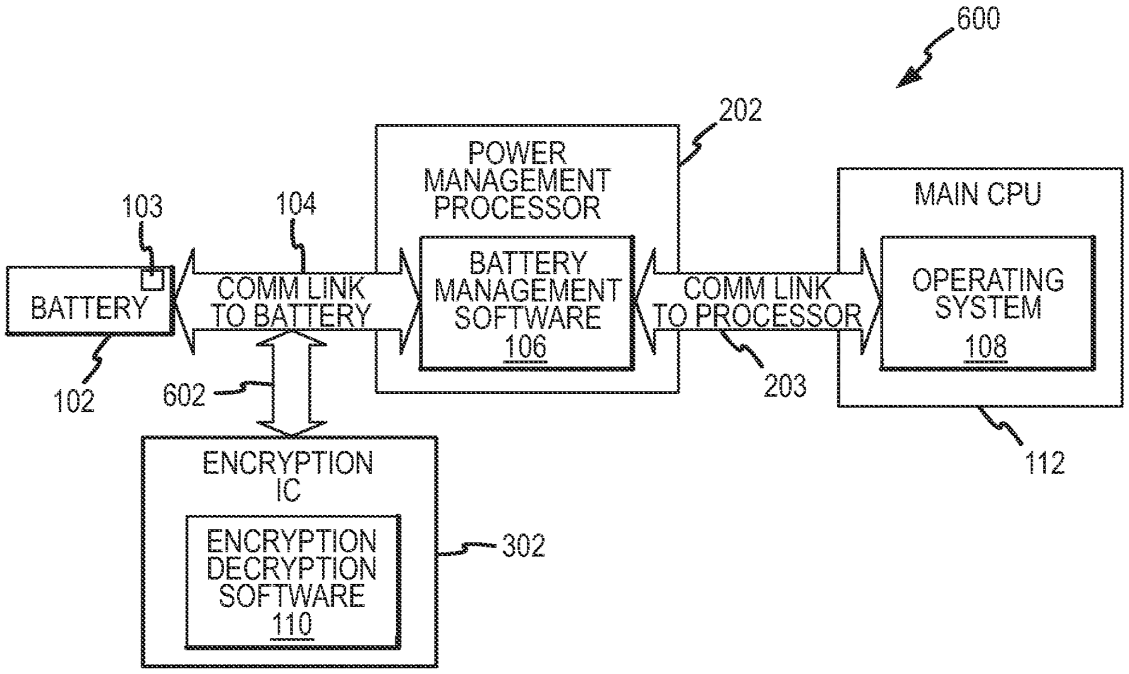

600
202
POWER MANAGEMENT PROCESSOR
MAIN CPU
103
104
BATTERY
COMM LINK TO BATTERY
BATTERY MANAGEMENT SOFTWARE 106
COMM LINK TO PROCESSOR
OPERATING SYSTEM 108
102
602
203
112
ENCRYPTION IC
ENCRYPTION DECRYPTION SOFTWARE 110
302

FIG.6

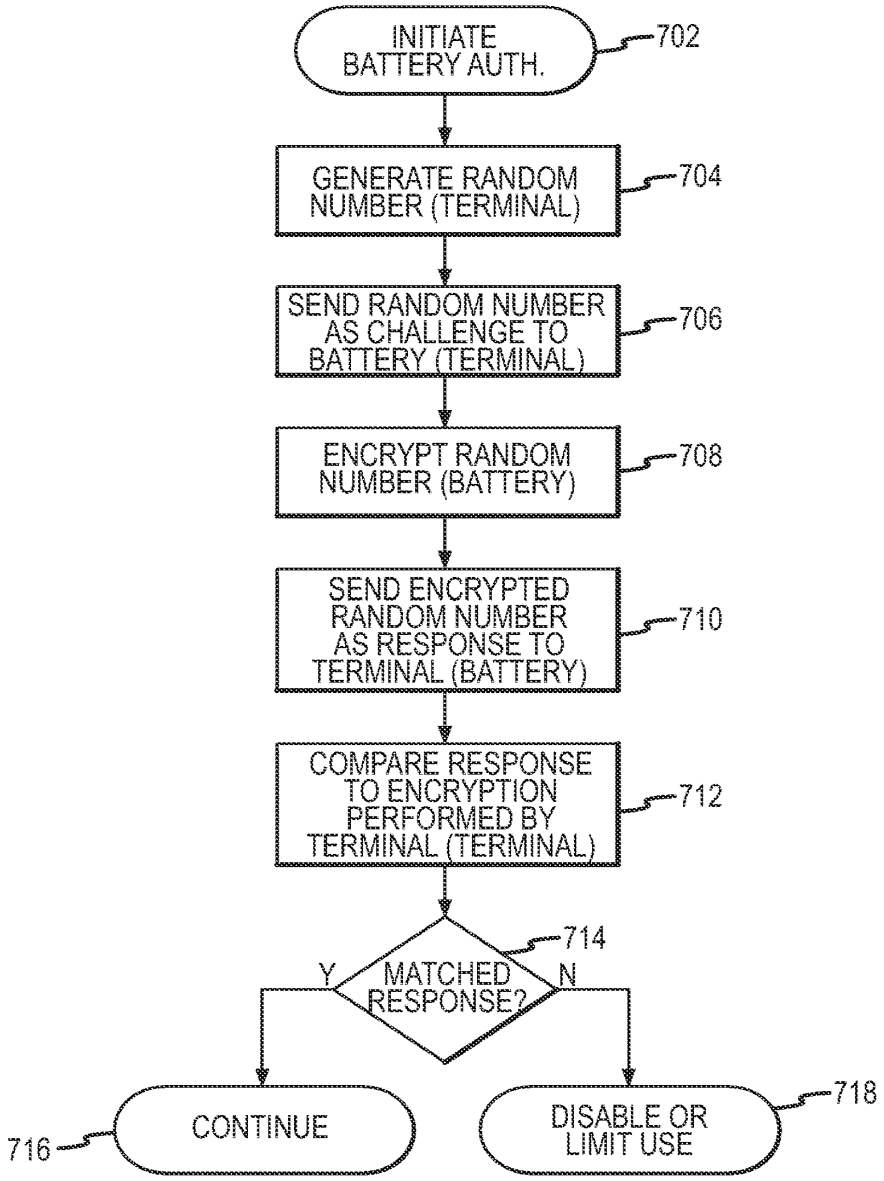

INITIATE BATTERY AUTH.
702
GENERATE RANDOM NUMBER (TERMINAL)
704
SEND RANDOM NUMBER AS CHALLENGE TO BATTERY (TERMINAL)
706
ENCRYPT RANDOM NUMBER (BATTERY)
708
SEND ENCRYPTED RANDOM NUMBER AS RESPONSE TO TERMINAL (BATTERY)
710
COMPARE RESPONSE TO ENCRYPTION PERFORMED BY TERMINAL (TERMINAL)
712
MATCHED RESPONSE?
714
Y
N
CONTINUE
716
DISABLE OR LIMIT USE
718

FIG.7

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 October 2007 (25.10.2007)

PCT

(10) International Publication Number
WO 2007/120215 A2

(51) International Patent Classification:
G06Q 99/00 (2006.01)

(21) International Application Number:
PCT/US2006/045110

(22) International Filing Date:
22 November 2006 (22.11.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/286,890 23 November 2005 (23.11.2005) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier applications:

US	11/286,890 (CON)
Filed on	23 November 2005 (23.11.2005)
US	10/854,604 (CIP)
Filed on	26 May 2004 (26.05.2004)
US	10/248,894 (CIP)
Filed on	27 February 2003 (27.02.2003)
US	60/360,023 (CIP)
Filed on	27 February 2002 (27.02.2002)

(71) Applicant *(for all designated States except US)*: **IMAGINEER SOFTWARE, INC.** [US/US]; 3452 North Hackett Avenue, Milwaukee, WI 53211 (US).

(72) Inventors; and

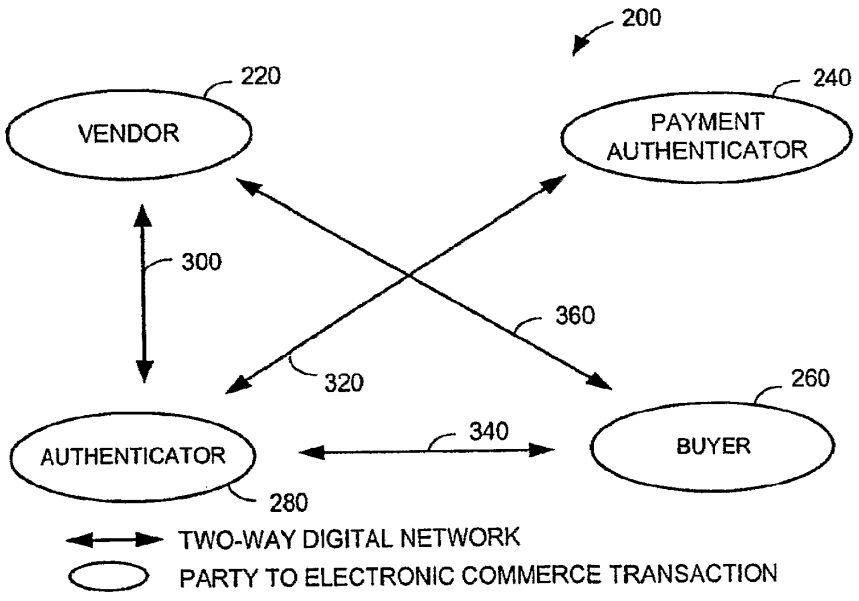
(75) Inventors/Applicants *(for US only)*: **SELLARS, William** [US/US]; 3452 North Hackett Avenue, Milwaukee, WI 53211 (US). **MALINA, Richard** [US/US]; 7914 East Field Circle, Mequon, WI 53097 (US). **COCHRAN, William** [US/US]; 307 West Washington Street, Champaign, IL 61820 (US).

(74) Agent: **STETTNER, Derek, C.**; Michael Best & Friedrich LLP, 100 East Wisconsin Avenue, Suite 3300, Milwaukee, WI 53202-4108 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, ME, MG, MK, MN, MW, MX,

[Continued on next page]

(54) Title: SECURE ELECTRONIC COMMERCE USING MUTATING IDENTIFIERS



(57) Abstract: Methods and systems for performing electronic commerce using mutating identifiers. One method can include encrypting buyer transaction data with a first mutating identifier, transmitting the buyer transaction data to an authenticator device, decrypting the buyer transaction data, generating a payment request, encrypting the payment request with a third mutating identifier, and transmitting the payment request to a payment authenticator device.

WO 2007/120215 A2

MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US (patent), UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SECURE ELECTRONIC COMMERCE USING MUTATING IDENTIFIERS

RELATED APPLICATIONS

[0001] This application claims priority from U.S. Patent Application Serial No. 11/286,890 filed on November 23, 2005, which is a continuation-in-part of U.S. Patent Application Serial No. 10/854,604 filed on May 26, 2004, which is a continuation-in-part of U.S. Patent Application Serial No. 10/248,894 filed on February 27, 2003, which claims priority to U.S. Provisional Patent Application Serial No. 60/360,023 filed on February 27, 2002, the entire contents of which are all hereby incorporated by reference.

BACKGROUND OF THE INVENTION

[0002] Embodiments of the invention relate to the distribution of content (such as text, audio, video, multi-media materials, and the like). More particularly, the invention relates to the distribution of such content in a manner that ensures that the copyrights and other similar legal rights of the content owner are respected.

[0003] More and more content is being delivered in digital form, and more and more digital content is being delivered over private and public networks, such as Intranets, the Internet, cable TV networks, and the like. For consumers of such content, digital versions (as opposed to analog, paper copy, and other forms) provide various advantages such as enhanced fidelity, improved and greater playback options, interactivity, and others. Online or network delivery generally offers greater convenience and timeliness. Online delivery is also cheaper than other delivery methods, and this benefits content publishers.

[0004] Most current and potential digitally distributed content is, like most books, distributed in a manner such that the publisher or owner generally gives or sells the content to a consumer, but continues to restrict rights to use the content even after the content is under the sole physical control of the consumer. For instance, a content owner typically retains copyright to the content so that the consumer cannot legally reproduce or publish the content without permission. Digital content, as opposed to older forms of media, allows a content owner to adjust pricing according to whether the consumer is allowed to make a persistent copy, or is just allowed to view the content as it is delivered.

WAVELYNX EX. 1003, Pg. 772
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0005] Despite the worthwhile attributes of digital and network distribution, content owners are still generally reluctant to distribute content, particularly high-value content, via networks because unauthorized duplication, piracy, and distribution of digital content (e.g., as was done by users of Napster) is very easy. Unlike analog recorders, photocopiers, and other older devices, current technology permits unlimited, pristine copies of digital content to be made. And, in most instances, copies of the digital content can be made very quickly or nearly instantaneously. Furthermore, even current protective measures such as public key encryption and the content scrambling system ("CSS"), which is used for digital versatile discs, have been defeated.

[0006] Furthermore, once a key is gained or discovered, either legally or illegally, that unlocks the encrypted content, that piece of content is forever compromised. This free access to the content allows the possessor of the key to make and distribute endless copies of the decrypted content. In the event that a key is used to circulate illegal copies of a content item, it is generally impractical to track the illegally distributed content back to the individual who initially owned the key.

[0007] In addition to distributing content, the availability and popularity of the networked systems, such as the Internet, has changed the landscape of commerce. Electronic commerce ("e-commerce") has grown into a large and important part of business. Providing electronic commerce capabilities has become a necessity for businesses to compete nation-wide and worldwide.

[0008] In addition to providing electronic commerce capabilities, businesses are also encouraged to provide secure electronic commerce capabilities. Most buyers will not provide billing or confidential information to complete an electronic commerce transaction unless they are promised that their data will be kept secure. Many current electronic commerce transaction protocols use a variation of a public key exchange protocol. The protocol establishes a secure connection between the buyer and the seller. Once a secure connection is established, the confidential information (e.g., a credit card number, account number, and the like) is communicated from the buyer to the seller. The seller then uses the information to obtain payment from a company (e.g., a credit card company or financial institution).

[0009] However, in the above protocol, once the credit card or confidential information is provided to the seller, the seller has the information indefinitely. There have been instances

WAVELYNX EX. 1003, Pg. 773
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

where the security of that information has been compromised. Such security compromises pose several problems, including consumer wariness of electronic transactions.

SUMMARY OF THE INVENTION

[0010] In light of the above, there is a need to provide a method and system of distributing content that ensures that the rights of content owners are respected. There is also a need to provide a mechanism that would track thieves and deter them from illegally distributing content. Furthermore, there is a need to provide a method and system of performing secure electronic commerce.

[0011] The present invention provides among other things a multiple party system for distributing content. In one embodiment, three parties are involved in the system: a consumer, a content provider, and an authenticator. The distribution of content from the content provider to the consumer is carried out using a predefined protocol, mutating IDs, and licenses. The authenticator controls the distribution of mutating IDs and verifies the identity of both parties.

[0012] In another embodiment, four parties are involved in a multiple party system for distributing content. The system includes a consumer, a service provider, an authenticator, and a content provider. The distribution of content from the content provider to the consumer, through the services of the service provider, is carried out using a predefined protocol, mutating IDs, and licenses. The authenticator controls the distribution of mutating IDs and verifies the identity of one or more parties.

[0013] Embodiments also provide methods for distributing content. One method for distributing content includes having a content provider make a request to send content to a consumer. The request includes an encrypted identifier for the content to be sent. In response to the request, the consumer encrypts the request to create an authentication request and sends the authentication request to an authenticator. The authenticator checks the authentication request and, if valid, informs the content provider to send the encrypted content identified in the initial request to the consumer. The authenticator sends a decryption key to the consumer so that the consumer may decrypt and view or consume the content.

WAVELYNX EX. 1003, Pg. 774
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0014] In another embodiment, the method includes having a consumer make a request for content to a service provider and relaying the request to a content provider. In response to the request, the content provider creates a license that includes identifying information concerning the service provider and encrypted information identifying the requested content. The license is sent to the service provider. The service provider encrypts the license from the content provider and sends this message to the consumer. The consumer encrypts the message to create an authentication request and sends the authentication request to an authenticator. The authenticator checks the authentication request and, if valid, informs the content provider to send encrypted content specified in the license to the consumer. The authenticator sends a decryption key to the consumer so that the consumer may decrypt and consume the content.

[0015] Embodiments of the invention further provide a method for distributing content where each copy of the content is uniquely watermarked. The method includes having a consumer make a request for content to a service provider and relaying the request to a content provider. In response to the request, the content provider creates a license that includes identifying information concerning the service provider and encrypted information identifying the requested content and a unique watermark. The license is sent to the service provider. The service provider encrypts the license from the service provider and sends this message to the consumer. The consumer encrypts the message to create an authentication request and sends the authentication request to an authenticator. The authenticator checks the authentication request and, if valid, informs the content provider to send encrypted content, watermarked with the watermark specified in the received license, to the consumer. The authenticator sends a decryption key to the consumer so that the consumer may decrypt and view the watermarked content.

[0016] In another embodiment, the invention provides a content distribution system that includes an authenticator and a content provider and a consumer. The content provider has content and a content identifier, and generates a first mutating identifier associated with the content identifier. The system also includes a consumer device that is operable to send requests to receive content to and to receive content from the content provider. The consumer device receives the first mutating identifier from the content provider in the form of a request, generates a second mutating identifier associated with the first mutating identifier, and distributes the second mutating identifier to the authenticator. The authenticator

WAVELYNX EX. 1003, Pg. 775
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

validates the request and thereafter informs the consumer of the validity of the request. If the request is valid, the content provider thereafter encrypts the content and sends the encrypted content to the consumer while the authenticator sends a decryption code to the consumer.

[0017] In another embodiment, the invention provides a content distribution system that includes an authenticator and a content provider. The content provider has content and a content identifier, and generates a first mutating identifier associated with the content identifier. The system also includes a consumer device that is operable to generate a request for the content. In addition, the system includes a service provider. The service provider receives the request from the consumer device, receives the first mutating identifier from the content provider, generates a second mutating identifier associated with the first mutating identifier, and distributes the second mutating identifier to the consumer device. The consumer device generates a third mutating identifier associated with the second mutating identifier, and distributes the third mutating identifier to the authenticator. The authenticator validates the request and thereafter informs the service provider of the validity of the request. If the request is valid, the content provider thereafter encrypts the content and sends the encrypted content to the consumer while the authenticator sends a decryption code to the consumer.

[0018] Additional embodiments provide methods of performing electronic commerce using mutating identifiers. One method can include encrypting buyer transaction data with a first mutating identifier, transmitting the buyer transaction data to an authenticator device, decrypting the buyer transaction data, generating a payment request, encrypting the payment request with a third mutating identifier, and transmitting the payment request to a payment authenticator device.

[0019] Another embodiment provides methods for establishing communication between a first device and a second device. One method can include generating a request for a transaction key, encrypting the request with a first mutating identifier of the first device, transmitting the encrypted request to an authenticator device, generating a transaction key, generating a first message including the transaction key, and encrypting the first message with the first mutating identifier of the first device.

[0020] Embodiments also provide methods of performing electronic commerce using mutating identifiers. One method can include encrypting buyer transaction data with a first

WAVELYNX EX. 1003, Pg. 776
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

mutating identifier, transmitting the buyer transaction data to an authenticator device, encrypting buyer credentials with a first transaction key, and transmitting the buyer credentials to a payment authenticator device. The method can also include encrypting vendor transaction data with a second mutating identifier, transmitting the vendor transaction data to an authenticator device, encrypting vendor credentials with a second transaction key, and transmitting the vendor credentials to a payment authenticator device. Furthermore, the method can include decrypting the buyer transaction data, decrypting the vendor transaction data, generating a payment request for a payment authenticator device, encrypting the payment request with a third mutating identifier, and transmitting the payment request to the payment authenticator device. In addition, the method can include decrypting the payment request; decrypting the buyer credentials; decrypting the vendor credentials; generating a first response based on the buyer credentials, the vendor credentials, and the payment request; generating a second response based on the buyer credentials, the vendor credentials, and the payment request; transmitting the first response to the buyer device, and transmitting the second response to the buyer device.

[0021] Yet another embodiment provides electronic commerce systems. One system can include a vendor device; a buyer device configured to encrypt buyer transaction data with a first mutating identifier and to transmit the buyer transaction data to an authenticator device; and a payment authenticator device configured to approve or decline a payment request, to generate a first response for the buyer device, to generate a second response for the vendor device, to transmit the first response to the buyer device, and to transmit the second response to the vendor device. The authenticator device can be configured to decrypt the buyer transaction data, to generate a payment request for a payment authenticator device, to encrypt the payment request with a third mutating identifier of the payment authenticator device, and to transmit the payment request to the payment authenticator device.

[0022] One alternative embodiment of the invention involves encryption, but not mutating identifiers. One exemplary method includes encrypting buyer transaction data with a first buyer key, transmitting the buyer transaction data to an authenticator device, encrypting buyer credentials with a first transaction key, and transmitting the buyer credentials to a payment authenticator device. The method can also include encrypting vendor transaction data with a vendor key, transmitting the vendor transaction data to an authenticator device, encrypting vendor credentials with a second transaction key, and

WAVELYNX EX. 1003, Pg. 777
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

transmitting the vendor credentials to a payment authenticator device. Furthermore, the method can include decrypting the buyer transaction data, decrypting the vendor transaction data, generating a payment request for a payment authenticator device, encrypting the payment request with a payment request key, and transmitting the payment request to the payment authenticator device. In addition, the method can include decrypting the payment request; decrypting the buyer credentials; decrypting the vendor credentials; generating a first response based on the buyer credentials, the vendor credentials, and the payment request; generating a second response based on the buyer credentials, the vendor credentials, and the payment request; transmitting the first response to the buyer device, and transmitting the second response to the buyer device.

[0023] Although mutating identifiers are preferred, the alternative described in the previous paragraph can be implemented with, for example, asymmetric encryption schemes. In addition, it might be possible to use symmetric systems where keys are not mutated at all or only episodically mutated. The other electronic commerce protocols, systems, and schemes described herein may also be modified in this way; that is, modified to use asymmetric encryption, non-mutating keys, or episodically mutated keys.

[0024] Embodiments of the invention will become apparent by consideration of the detailed description and accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] In the drawings:

[0026] Fig. 1 is a schematic illustration of a system of one exemplary embodiment of the invention where four entities are involved in communication.

[0027] Fig. 2 is a schematic illustration of a system of another exemplary embodiment of the invention where three entities are involved in communication.

[0028] Fig. 3A is an illustration of a bit stream (called a "mutating ID") used in one embodiment of the invention.

[0029] Figs. 3B and 3C are illustrations of ways to distribute mutating IDs.

WAVELYNX EX. 1003, Pg. 778
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0030] Fig. 4 is a schematic illustration of a license structure for one exemplary embodiment of the invention.

[0031] Fig. 5 is a schematic illustration of a portion of the system shown in Fig. 1.

[0032] Fig. 6 is a schematic illustration of a communication protocol used in the system shown in Fig. 1.

[0033] Fig. 7 is a schematic illustration of a portion of the system shown in Fig. 1, illustrating license distribution to multiple service providers.

[0034] Fig. 8 is a schematic illustration of a mutating identifier cycle used in one form of the invention.

[0035] Fig. 9a is an exemplary schematic illustration of content key management in one embodiment of the invention.

[0036] Fig. 9b is an exemplary schematic illustration of data flows that occur when content is requested under the situation illustrated in Fig. 9a.

[0037] Fig. 10a is an exemplary schematic illustration of content key management in another embodiment of the invention.

[0038] Fig. 10b is an exemplary schematic illustration of data flows that occur when content is requested under the situation illustrated in Fig. 10a.

[0039] Fig. 11 is an exemplary illustration of a content request.

[0040] Fig. 12 is an exemplary illustration of a content request showing an approval phase.

[0041] Fig. 13 is an exemplary illustration of a content request showing a delivery phase.

[0042] Fig. 14 is a schematic illustration of a license structure for another exemplary embodiment of the invention where watermarking is used.

[0043] Fig. 15 is a schematic illustration of a communication protocol used in the system shown in Fig. 1 with the addition of watermarking the content.

WAVELYNX EX. 1003, Pg. 779
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0044] Fig. 16 is a schematic illustration of exemplary embodiments of the devices used to distribute content where three entities are involved in a communication.

[0045] Fig. 17 is a schematic diagram of hardware inside one of the peripherals shown in Fig. 16.

[0046] Fig. 18 is a schematic illustration of a system of one exemplary embodiment of the invention where four entities are involved in a communication to perform electronic commerce.

[0047] Fig. 19 is a schematic illustration of a communication protocol used in the system shown in Fig. 18.

[0048] Fig. 20 is a schematic illustration of a system of another exemplary embodiment of the invention where four entities are involved in a communication to perform electronic commerce.

[0049] Fig. 21 is a schematic illustration of a communication protocol used in the system shown in Fig. 20.

[0050] Fig. 22 illustrates a brute force attack according to one embodiment of the invention.

[0051] Fig. 23 illustrates types of data included in data to be encrypted according to one embodiment of the invention.

[0052] Figs. 24 and 25 illustrate an encryption strategy for protecting the security of undiscoverable data according to one embodiment of the invention.

DETAILED DESCRIPTION

[0053] Before embodiments of the invention are explained in detail, it is to be understood that the invention is not limited in its application to the details of the construction and the arrangements of the components set forth in the following description or illustrated in the drawings. The invention is capable of still other embodiments and of being practiced or being carried out in various ways. Also, it is to be understood that the phraseology and

WAVELYNX EX. 1003, Pg. 780
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

terminology used herein is for the purpose of description and should not be regarded as limiting.

[0054] In particular it should be understood that the invention is implemented using various computer devices, such as personal or home computers, servers, and other devices that have processors or that are capable of executing programs or sets of instructions, including special purpose devices such as set top boxes. In general, the invention may be implemented using existing hardware or hardware that could be readily created by those of ordinary skill in the art. Thus, the architecture of exemplary devices will not be explained in detail, except to note that the devices will generally have a processor, memory (of some kind), and input and output devices. In some cases, the devices may also have operating systems and application programs that are managed by the operating systems. The hardware devices will also generally need some ability, depending on the role of the device in the particular embodiment of the invention implemented, to compress or decompress data and to encode data or decode encrypted data. In many instances, a decompression capability may be provided using available codecs, such as hardware-implemented MPEG codecs. A decryption capability may be provided using a decryption hardware or software module capable of decrypting data that is encrypted using the encryption algorithm chosen. One encryption algorithm that is suitable for use in embodiments of the invention is the Rijndael algorithm, an example of which is available at http://www.esat.kuleuven.ac.be/~rijmen/rijndael/rijndaelref.zip.

[0055] Fig. 1 illustrates an exemplary system 20 configured to distribute content over a network. In reality, one or more networks or communication systems such as the Internet, the telephone system, wireless networks, satellite networks, cable TV networks, and various other private and public networks could be used in various combinations to provide the communication links desired or needed to create embodiments or implementations of the invention, as would be apparent to one of ordinary skill in the art. Thus, the invention is not limited to any specific network or combinations of networks. However, it is preferred that the networks or communication system used have the ability to support secure communications such as communications with data encrypted with a version of Rijndael encryption, secured socket layer ("SSL") communications, or others. Furthermore, data can be transferred from one party to another with wire, digital satellite service ("DSS"), or physical media being physically carried from one party to another.

WAVELYNX EX. 1003, Pg. 781
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0056] In the embodiment shown in Fig. 1, the system 20 includes four participants: a content owner or provider 22, a service provider 24, such as a cable company, Internet service provider, or the like, a consumer 26, and an authenticator 28. Although, only one content provider, service provider, and consumer are shown, in most implementations of the invention numerous content providers, service providers, and consumers will be involved. Further, there could even be multiple authenticators, although only one is required. In practice, it is likely that the following relationship will exist: number of authenticators < number of content providers < number of service providers < number of consumers, but again there is no limit on the number of participants or any requirement of a particular relationship between the numbers of the various types of participants.

[0057] In another embodiment shown in Fig. 2, the system 20 includes three participants: the content owner or provider 22, the consumer 26, and the authenticator 28. Although, only one content provider and consumer are shown, in most implementations of the invention numerous content providers and consumers will be involved. Further, as mentioned before, there could be more than one authenticator, although only one is required. Again there is no limit on the number of participants or any requirement of a particular relationship between the numbers of the various types of participants.

[0058] The participants 22, 24, 26, and 28 are connected to each other via two-way links 30, 32, 34, 36, and 38. These links may be a constructed from all or part of the networks mentioned above. The system 20 uses a key-based encryption algorithm and currently available algorithms such as the Rijndael algorithm may be used. The ultimate choice for the algorithms used will depend on a variety of factors including a trade off between the strength of the algorithm (in terms of being broken) and speed (in terms of a processor's capability to perform the mathematical operations required by the chosen algorithm).

[0059] In one embodiment of the invention, it is assumed that the consumer has a decoding processor or similar device, which may, for example, take the form of a "set top box," home computer, or other device. In this same embodiment, it is assumed that the decoding processor is in a "hostile environment," in the sense that consumer may possibly want to tamper or otherwise circumvent rights management features of the decoding processor. Thus, it is preferred that the decoding processor be housed in a container that has the ability to detect intrusions to its interior. It is also preferable, that the decoding processor

WAVELYNX EX. 1003, Pg. 782
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

have "persistent" memory such as non-volatile RAM, EPROM, or other storage where data remains intact after the removal of power. The persistent memory is used to store identification information, which may be a "mutating ID," that, preferably, changes over time.

[0060] In preferred embodiments, the system 20 uses a random number generator to generate certain numbers used by a protocol implemented or followed by the system. It is preferred that the random number generator produce numbers that are as truly random as is possible with the particular technology used to implement the invention. In one embodiment, communication traffic, such as requests from customers to obtain content, is used to create random numbers. Such requests occur, in general, in an unpredictable manner. Thus, the random numbers generated based on such traffic are also truly or nearly truly random, as opposed to pseudo random numbers generated with algorithmic methods.

[0061] In the exemplary embodiment shown, each of the parties 22-28 in the system 20 has different responsibilities and it is assumed that each party trusts the authenticator 28. Further, it is preferable that content provider 22, service provider 24, and consumer 26 be assigned a changing or mutating identifier ("ID"), which is explained further below.

Content Provider ("Alice")

[0062] The content provider 22 is an entity such as a movie studio, recording company, or any other entity that wishes to distribute content electronically. In general, it is assumed that the content provider 22 owns the copyright or other intellectual property rights in the content or is authorized by the owner of such rights to distribute the content. It is assumed that the content provider 22 wants to be reimbursed fairly for each copy of its content distributed using the system 20. It is also assumed that the content provider 22 wants to track each provided copy of its content to both the service provider 24 and consumer 26 to which it is allocated. Thus, in one embodiment of the invention, the system 20 is configured such that the content provider 22 can produce a virtual inventory of licenses for content using a list of mutating IDs (generally created as needed), where each license gives authorization to view or in some cases maintain a copy of the content provided. The virtual inventory (or set of licenses) may be assigned to various distribution entities, such as one or more service providers. As the virtual inventory is consumed, the consumption is tracked in order to record, log, or note which service provider 24 provided copies of content to one of its

WAVELYNX EX. 1003, Pg. 783
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

consumers. For instance, tracking of the virtual memory allows a content provider such as a movie studio, which has sold distribution rights to a cable company and a satellite broadcasting company, to determine which of the entities, the cable company or the satellite broadcasting company, distributed the content to the subject consumer. In preferred embodiments, the content provider 22 is the sole encrypter of its content and controls the decoding of the content by, for example, denying a request, as explained further below.

[0063] In another embodiment of the invention, the system 20 is configured such that the content provider 22 can produce a list of watermarks, apply a unique watermark to each copy of its content, and produce a virtual inventory of licenses for watermarked content using a list of mutating IDs (generally created as needed), where each license gives authorization to view or in some cases maintain a copy of the watermarked content provided. The virtual inventory (or set of licenses) may be assigned to various distribution entities, such as one or more service providers. As the virtual inventory is consumed, the consumption is tracked in order to record, log, or note which service provider 24 provided content copy to one of its consumers and, furthermore, which consumer received a particular watermarked piece of content. For instance, tracking of the virtual memory allows a content provider such as a movie studio, which has sold distribution rights to a cable company and a satellite broadcasting company, to determine which of the entities, the cable company or the satellite broadcasting company, distributed the content to the subject consumer. The tracking system also allows the consumption activity of individual consumers to be recorded or logged since a watermarked content can be mapped to a specific consumer.

Service Provider ("Bob")

[0064] In the embodiment shown, the service provider 24 distributes content of the content providers. However, service providers may also have several additional responsibilities, including identification of themselves and the subject content with mutating IDs. In the embodiments described, a user such as the consumer 26 without the appropriate mutating IDs cannot decode content. In many scenarios, the service provider 24 provides the requested content off of or from a storage device that is local to the service provider 24. However, the invention is not limited to any particular location of the content and content may be retrieved from storage of the content provider 22 and then transferred to the consumer 26. In preferred embodiments of the invention, every service provider 24 sees each request

WAVELYNX EX. 1003, Pg. 784
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

from consumers in the system 20 and receives authentication from the authenticator 28. In some embodiments, any one of the service providers in a particular embodiment may be responsible for shipping or transferring the content to the consumer. This allows content providers to avoid having to deliver their content to service providers, if they so desire. In other words, the service provider 24 need not possess (such as by maintaining encrypted copies in local storage) the content that is ordered by consumers.

Consumer ("Carol")

[0065] It is assumed that at least some consumers may wish or attempt to view or consume content without paying for it. Therefore, measures are provided to prevent unauthorized consumption of content. The mutating IDs mentioned above provide one mechanism by which content decoding and, therefore, consumption is controlled. By encapsulating multiple mutating IDs, the set top box is able to prove to the authenticator 28 that 1) the set top box is a licensed decoder of the content, 2) the service provider 24 is a licensed distributor of the content, and 3) the content itself was licensed for use by the content provider 22 for the consumer, as will be explained in greater detail.

Authenticator ("Trent")

[0066] The authenticator 28 is the repository that holds the data necessary to decode a particular piece of content, or watermarked content if applicable. In the embodiments discussed, the authenticator 28 verifies the consumer 26, the service provider 24, and the content by their mutating IDs before sending any decoding information to the subject consumer 26. The authenticator 28 is also the source of mutating IDs and keeps track of such IDs using a database or similar mechanism.

Mutating ID

[0067] An exemplary mutating ID 38 is shown in Fig. 3a. The mutating ID 38 is an identifier having two portions: a first portion 40 and a second portion 42. The first portion 40 is an identifying number, which is a random number. The second portion 42 is an encoding/decoding key, also a random number and preferably a symmetric cipher key. As implemented in the embodiments discussed herein, a mutating ID can be used only once and then can not be used again. Mutating IDs are generated and tracked by the authenticator 28.

WAVELYNX EX. 1003, Pg. 785
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Because the mutating IDs are one-time-use mechanisms, once a service provider or consumer or other entity has used its supply of mutating IDs, additional mutating IDs must be obtained from the authenticator 28. The data in a mutating ID is chosen at random with an equal probability of all possible mutating IDs. Once a request or decode for specific content has been made, the three mutating IDs (consumer, service provider, content) are discarded and new mutating IDs are generated for additional transactions, in a manner described in more detail below. Information regarding how mutating IDs are distributed to parties in the system 20 is also provided in Figs. 3b and 3c. Specifically, entities of the system can receive either numerous mutating IDs or a single mutating ID depending on the desired use. In one embodiment of the invention, a provider 43, who may perform functions of both the content provider and the service provider, may request multiple number/key pairs from the authenticator. Each provider 43 requires numerous number/key pairs since each copy of a content item must have a unique license made from a unique number/key pair. The authenticator creates as many mutating IDs as the provider 43 requires and sends a list of pairs 26 back to the provider 43. The provider 43, knowing the quantity of number/key pairs requested and the size of each pair, breaks the list into individual number/key pairs. In other embodiments of the invention, each entity of the content distribution system needs a mutating ID to identify itself to the authenticator, and after using a mutating ID an entity requires a new mutating ID from the authenticator. A single, new mutating ID is sent to the entity, which replaces the entity's previous mutating ID.

[0068] As should be apparent, embodiments of the invention are symmetric key systems. Symmetric key systems commonly encounter key management issues as the number of entities of the system grows. For example, a network of n entities requires $n(n-1)/2$ keys to enable all entities to communicate with one another. Thus, for a system of 1000 entities, where every entity wishes to send identical content to every other entity, almost a half million keys are required.

[0069] Disclosed embodiments, however, do not require a separate key for every pair of entities of the system. As will be illustrated, each entity and content distributed receives one key, which is mutated after each use. For a system of 1000 entities, only 2000 keys are required compared to the almost half of a million keys with previous symmetric key systems. Also, the authenticator is not required to store the entire bit string of the mutating ID. The

WAVELYNX EX. 1003, Pg. 786
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

authenticator may use a hash function or simply a positional index to map each key partition of the mutating ID into a memory storage location based on the corresponding number.

[0070] Other differences between embodiments of the invention and prior security systems relate to speed and reduced vulnerability to certain attacks. The use of symmetric keys also allows fast computation (as compared to public key systems) and reduces the effectiveness of chosen plaintext attacks. Embodiments of the invention are relatively fast because of they use symmetric keys rather than public keys. The fundamental concept behind public key systems is the use one-way functions. One-way functions are easy to compute but hard to reverse. Public key systems use trapdoor one-way functions that provide a key to compute the one-way function in the opposite direction. Public key systems provide public keys for each participant that are freely accessed and used as a one-way function to apply to a message. Public key systems also provide private keys (which are underiveable, at least as originally thought, from the public key) to each individual participant to compute the message given the calculation of the one-way function. The security of public key systems relies on the fact that the private key cannot be derived from the public key. In order to maintain this requirement, the one-way functions used in public key systems are complex. The added complexity, however, comes at the cost of added computation time. Public key systems are often 1000 times slower than symmetric key systems.

[0071] Regarding reduced vulnerability to attacks, a chosen-plaintext attack occurs when an intruder has access to an encryption key or process, chooses specific plaintext to encrypt, and attempts to gain knowledge from the encrypted text. In public-key systems an individual's public key is known to all participants in a communication system. Any intruder can encrypt an endless number of messages using an individual's public key. If an attacker encrypts possible messages with an individual's public key and then intercepts an encrypted message sent to the individual, the intruder can compare the intercepted message with messages he or she has created. If an interception message matches an encrypted message created by the intruder, the message has been compromised and the intruder can now read a message that was not intended for him or her. This attack is easy and effective if a small number of possible messages exist, but even if the number of possible messages is more than the intruder is able to encrypt or compare with intercepted encrypted messages, just knowing that an intercepted encrypted message does not correspond to a particular message can provide useful information to the intruder. In either situation, the intruder will not be able to

WAVELYNX EX. 1003, Pg. 787
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

deduce the private key of the individual but he or she may be able to deduce the message, or information regarding the message, sent to the individual. Since the embodiments of the invention utilize a symmetric key system, chosen-plaintext attacks are not applicable because encryption keys are not public knowledge.

[0072] There is another problem with prior symmetric key systems and public key systems. Once an unauthorized entity gains access to an authorized key, the unauthorized entity can decode all messages encrypted with the compromised key, and, perhaps more dangerous, can encrypt false messages to deceive other entities of the system. The mutating ID protocol reduces this weakness by mutating each symmetric key after it has been used. Even if a key is compromised, the compromised key cannot be used to generate future messages nor be used to decrypt future message since it is marked by the authenticator as used and is never used again to encrypt messages.

Protocol

[0073] The system 20 uses a protocol to govern communications between entities. Each entity is randomly assigned a one-time use number/key pair, or mutating ID, (such as the identifier or ID 38 shown in Fig. 3A) that is to be tagged onto a previously mutated ID by the authenticator 28. As noted, each mutating ID includes a random number 40 and a random corresponding coding key 42. The one-time-use number/key pair may take the form of a modified hash. In addition to being random, the one-time-use number/key pair or the hash is immediately discarded after each decryption. In other words, the protocol generates a new random number that has never been used when a hash or a one-time-use number/key is needed. In addition to identifying the entity participating in the content distribution system, the one-time-use number/key is also a completely unrelated hash from the entity using it. That is, the hash does not contain any information concerning the identity of the entity. In this way, the identities of the entities are blind to all participants except for the authenticator 28.

[0074] The authenticator 28 also generates encryption keys for the content distributed over the system 20. Any entity wishing to distribute content requests a key. The entity sending the content supplies the authenticator with a function, or any identifying string, of the content to distribute and the authenticator responds with an associated key. The key, like the mutating IDs, is unrelated to the content that it encrypts. The authenticator also has no

WAVELYNX EX. 1003, Pg. 788
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

knowledge of the true content since only a function of the content or a random identifier is provided. The authenticator records the key and the associated function, or identifying string, of the content. The authenticator supplies the key to any authorized entity of the system 20 that makes a legitimate request. A request for the associated key to a content item includes a reference to the function, or identifying string, of the content item. The authenticator 28 looks up the matching key to the function, or identifying string, indicated in the request and returns the found key.

[0075] Certain embodiments of the system 20 are implemented with an encryption algorithm and a random number generator. Preferably, the encryption algorithm is a symmetric key-based encryption algorithm. The keys are permutation identifications, offsets, and skips. All three could be bundled into one object called a "key." Therefore, any key based encryption algorithm can be used in embodiments of the invention. Since introducing a new encryption algorithm can be a very time consuming process, the system 20 was created to allow already existing and tested key-based encryption algorithms to be used.

[0076] With respect to generating a random number, in the embodiment shown, three different procedures are used. Of course, other combinations of random number procedures can also be used to generate a random number. The first random number generation procedure is a standard congruential random number. The second random number generation procedure generates a random number that is used to determine a sampling rate of a random stream.

[0077] Acquiring a random stream can be difficult. By conventional definition, a random stream or a set of numbers is considered random if and only if the set is the most compact way to represent the numbers. For instance, given the set 2, 4, 6, 8, 10, ⊕, a more compact representation would be $\{2i|i \in Z+\}$ or any even number in the positive integer number set. Another way to state this is a set of numbers for which there is no discernable "pattern." The goal of encrypted communication is to remove all patterns from the encrypted data being transmitted so that intelligent guessing cannot be used to decrypt the encrypted transmitted data. In embodiments of the system 20, the authenticator 28 provides all the random numbers used in the distribution and transmission of content. As discussed in more detail, the sequence of generated numbers is a sequence of random numbers or a random stream, or at least a close approximation to a random stream.

WAVELYNX EX. 1003, Pg. 789
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0078] The third process of the random number generator is to ensure that there is no deterministic way to retrieve the next number. The random streams coming into and leaving the authenticator 28 are encrypted and thereby contain encrypted data. This non-deterministic mechanism used to generate the random stream helps ensure that the random number sequence cannot be represented by a more compact expression, hence defining a random stream.

[0079] For example, the authenticator 28 is designed to receive requests to decode content. These requests reach the authenticator 28 in a random order. For instance, suppose consumer X requests the key to movie Y, and consumer W requests the key to song Z, etc. These requests are formalized by the protocol as sequences of numbers that have been arbitrarily chosen either to be intentionally arbitrary or encrypted by an arbitrary key. Since the requests are arbitrary in nature and are processed in an arbitrary fashion, a stream of random numbers is naturally generated. By sampling this stream in a semi-arbitrary way (i.e., congruential random number), a good sequence of random numbers is generated.

[0080] In one embodiment, the protocol used in the system 20 combines a packet assembly disassembly generator or PAD, key pairing, and a RC4 random stream cipher. More specifically, to encrypt the information in a window of data, a PAD series is generated once a PAD generator is chosen. A second random stream P is generated based on the PAD series in the following manner:

$$p_k = pad_{(k_i) \bmod n} \otimes pad_{(k_{i+1}) \bmod n} \otimes \cdots \otimes pad_{(k_{i+j}) \bmod n}$$

where p_k is the k-th element of the P series, and $pad_{(kj)\ modn}$ is the j-th element of the PAD series. That is, every element of the P series is a combination of exclusive-ORing of the elements in the series generated by the PAD generator. The key pair generally includes a public key and private key. These keys do not have any mathematical relation. They are generated randomly and independently of each other. The keys themselves are simply twenty or more digit numbers. Each entity has a unique key pair.

[0081] An exemplary embodiment of the protocol will now be discussed in greater detail. In the embodiment shown in Fig. 1, the content provider 22 performs encryption (and is sometimes referred to as the "encrypter"). The content provider 22 encrypts content with a particular key or set of keys K. The content provider 22 may store or keep the set of keys K

WAVELYNX EX. 1003, Pg. 790
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

or, alternatively, the authenticator 28 may maintain the set of keys. If so maintained, the authenticator 28 maintains the set of keys K in secret. The content of the content provider is assigned a secret identifying label (e.g., a number that, in general, never changes). The label is given to the authenticator 28 and the authenticator associates the label with the keys needed to decode the encrypted content. The association is indirect, as the association process does not provide any entity access to the actual keys. The encrypted content can now be given to the service provider 24 or another entity without fear of unauthorized decoding, because only the content provider and the authenticator have the actual keys needed to decrypt the content. At this point, the content provider 22 creates the virtual inventory for the content. This involves asking the authenticator 28 for as many mutating IDs as the content provider 22 needs or wants to have. Each mutating ID represents a license allowing exactly one use or consumption of the content. As mentioned earlier, the mutating ID is a number and a key. The content provider 22 encrypts the identifying label with the mutating ID key and groups the mutating ID number and encrypted identifying label into a single piece of data called the "encrypted identifier," which is represented here as "$E_{content}$."

[0082] As each piece of content is consumed, the authenticator 28 tracks which service provider 24 the particular content came from and notifies the content provider 22 of each such decoding. Once it receives content, the service provider 24 combines each encrypted piece of content with other identifying data before distribution to the consumer 26. Each service provider 24 keeps a collection of mutating IDs on hand that are used to identify that particular service provider. As with all others, these mutating IDs are created and tracked by the authenticator 28. The service provider 24 also has a list of $E_{content}$ identifiers for each piece of content. Once requested, the service provider 24 picks an unused $E_{content}$ identifier and one of its unused mutating IDs. The service provider encrypts the $E_{content}$ identifier with the key of the chosen mutating ID and attaches the associated number creating a piece of data referred to herein as the "distributable content" or "$E_{distrib}$." The service provider 24 sends the $E_{distrib}$ content to the consumer 26 and awaits a confirmation signal from the authenticator 28 that allows decoding.

[0083] The confirmation signal is received as an encrypted parcel of data that can be decrypted with the mutating ID key used to create the $E_{distrib}$ content. The confirmation signal is an agreed on, secret set of bytes set by the service provider 24 and authenticator 28. Once

WAVELYNX EX. 1003, Pg. 791
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

the confirmation has been received and verified, the service provider 24 can send the encrypted content or $E_{content}$ identifier to the consumer 26.

[0084] As noted above, it is assumed that the consumer 26 wants to cheat the system 20. For that reason, all communications between the consumer 26 and the authenticator 28 are encrypted by some encrypted communication method. The consumer 26 is given only one mutating ID at a time. When the consumer 26 wants to view or receive some content, the consumer 26 makes a selection of the desired content using the set top box or hardware in place at the site of the consumer 26. The hardware device then requests the content from the service provider 24 who sends the $E_{distrib}$ content for a particular decode. Once this has been received by the consumer 26, the $E_{distrib}$ content is encrypted with the consumer's mutating ID key and coupled with the mutating ID number into a consumer identifier referred to as an $E_{consumer}$ identifier. The $E_{consumer}$ identifier is then sent to the authenticator 28 for verification. Upon verification, the authenticator 28 informs the service provider 24 by a secure channel that the consumer's set top box is licensed to consume the subject content. The authenticator 28 also discards the current set top box mutating ID and sends the set top box a new, unused mutating ID.

[0085] The consumer 26 simultaneously receives, from separate sources, the encrypted data and keys necessary to decrypt the data. With this, the decoding device can decode the requested content.

[0086] Generation of mutating IDs and keeping track of them is the primary task of the authenticator 28. Distribution of mutating IDs to content providers and service providers can be handled via any acceptable means to both, as long as they stay secret to the respective receiver. To ensure that consumers do not ever have more than one mutating ID, once an $E_{consumer}$ identifier is verified as accurate, a new mutating ID is encrypted using the consumer's current mutating ID key and sent to the consumer 26. The consumer can then use that ID for the next transaction. Since the authenticator 28 keeps track of all mutating IDs, it can verify that an $E_{consumer}$ identifier is or contains a valid request. To do so, it finds the key associated with the number in the $E_{consumer}$ identifier and decrypt it, revealing the $E_{consumer}$ identifier. If no key can be found, then the authenticator 28 returns a failure. Using the exact same process, the authenticator 28 recovers the $E_{content}$ identifier. If no key can be found, then the authenticator 28 returns a failure. Once again, the authenticator 28 decrypts the $E_{content}$

WAVELYNX EX. 1003, Pg. 792
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

identifier, if possible. If it is possible, the authenticator 28 look ups the confirmation code for the service provider 24, encrypts it with the mutating ID key used by the service provider 24, and returns it to the service provider. A new mutating ID is then sent, followed by the decoding data. The authenticator 28 then notes or otherwise records all the participating parties, time, and content involved in the transaction for billing purposes.

[0087] As was noted above, one aspect of the invention is to ensure that copies of content are not obtained by unauthorized or unlicensed entities. In the embodiments discussed, obtaining an unauthorized copy of the content would require a person or entity to intercept a message and then break the encrypted content. It is assumed that this is impossible, at least in practical terms, because even if the content was intercepted (which would not be a trivial task) decoding it would be very difficult. This is so, in part, because it is possible, using the random numbers and encryption algorithm discussed to create encrypted data that would require a very long time (on the order of years and in the inventors' opinion, thousands of years) to decrypt. Types of attack would generally require correctly guessing or otherwise obtaining an $E_{consumer}$ identifier. However, since each mutating ID is used only once by the controller of that ID, there is no way to divine one $E_{consumer}$ identifier by inspecting another. Further, since the mutating IDs are calculated to be random (to the limit of the random number generator), it is infeasible to calculate one mutating ID from another.

[0088] In addition to providing a high degree of security, the protocol allows for interchangeable parties at each position. In other words, multiple content providers can use multiple service providers to reach multiple consumers using the authenticator 28 agreed upon by all parties. Further, positions can easily be merged or modified to fit many different distribution models. However, it is generally required that the virtual inventory be maintained by the content provider 22 and/or service provider. As noted, the virtual inventory is preferably a list including secret mutating IDs. The virtual inventory may be traded like any other commodity. Thus, rather than a discrete piece of data being the actual digital content of the desired piece of content, it becomes the $E_{consumer}$ identifier constructed by a particular set of events leading to one particular viewing of that content. Since the $E_{consumer}$ identifier cannot be constructed without cooperation from all parties involved, all parties have the ability to negotiate and reach a mutually beneficial arrangement for the consumption of digital content at the demand of the consumer 26. Should one party decide to

WAVELYNX EX. 1003, Pg. 793
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

not participate, communication with the authenticator 28 can prevent decoding by withdrawing the consent of one or more parties.

[0089] As was noted above, the system 20 is not limited to just four parties. More distribution layers can be introduced and verified through the unraveling process in the same manner. For instance, one service provider 24 can distribute to another service provider. The authenticator 28 can continue to unravel an $E_{consumer}$ identifier until it has found a content provider 22. This can be accomplished by the simple recursive algorithm used to create $E_{consumer}$ identifiers. The important distinction is that the implementation of the authenticator 28 controls the ultimate decoding of content based upon the $E_{consumer}$ identifier. Should a content provider 22 or service provider 24 want to control the distribution downstream, it is easy to control by informing the authenticator 28 to deny decoding of content under particular circumstances.

[0090] Exemplary embodiments of the invention will now be described using several examples.

[0091] As with many descriptions of communication protocols, names are assigned to the various entities (or computer systems associated with those entities) used in the protocol. In one embodiment, Alice (*A*), Bob (*B*), and Carol (*C*) represent various participants in a protocol and Trent (*T*) represents a trusted arbiter of communication. The following table, Table 1, is a list of other symbols used in this document to explain multiple embodiments of the protocol.

WAVELYNX EX. 1003, Pg. 794
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Table 1

Symbol	Meaning
A,B,B',C,T	Entities using the protocol.
M	A content item.
X_{id}	Some (not secret) digital identifier of X, like an email address, Account number, or the like.
X_{cred}	Secret information or credentials that identifies X
K_X	A key for a symmetric cipher associated with some entity X.
N_X	A one-use number associated with some key K_X.
$H(X)$	A function that produces a hash of X.
$E(K,X)$	A cipher that encrypts X with K.
$D(K,X)$	A cipher that decrypts X using K.
$W(D,X)$	A watermark function that applies watermark D to X.
$X \rightarrow Y : Z$	A message Z sent from X to Y.
$\{(N^i_X, K^i_X)\}$	A set of number/key pairs of arbitrary size associated with entity X.

[0092] An exemplary embodiment of the protocol involves three of the entities discussed above. The entity Alice or A performs the role of the content provider 22, entity Carol or C performs the role of the consumer 26, and entity Trent or T performs the role of the authenticator 28. Since the proposed protocol relies on a trusted authority, Alice and Carol trust Trent. Also, Alice has two secrets: a number (N_A) and a key for some symmetric cipher (K_A). Similarly, Carol has a secret number (N_C) and key (K_C). Further, all assigned number and keys are assigned and known by Trent.

[0093] For the purposes of this example only, assume Alice wishes to send an electronic mail message P_1 securely to Carol. Of course, Alice could represent any entity needing to distribute a variety of content items. Besides text messages, Alice could distribute music, images, video, data, etc. using the same protocol described herein.

[0094] To start, if Alice wishes to send the message P_1 to Carol, she requests a key for the message P_1 that she will use to encrypt P_1 and Carol will use to decrypt P_1. To do this Alice creates a label for the message P_1 by using a function to create a hash of the message and encrypts the label with her K_A, and prepends N_A.

$$A \rightarrow T: N_A E(K_A, H(P_1))$$

WAVELYNX EX. 1003, Pg. 795
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0095] When Trent receives the request from Alice and since he knows Alice's secrets, he looks up the key associated with N_A and decodes the request for a key to be used with the message P_1. Trent, however, does not receive the message P but only receives a hash of the message P_1. He can then generate a key K_{P1} and associate it with the supplied hash for future reference. Using this protocol, messages are kept secure even from Trent and keys generated by Trent offer no useful information regarding the data the key will encrypt. Trent does not receive information that would allow him to directly learn the message generated by Alice or allow him to corrupt the system by sending messages generated by Alice to unwanted parties. Alice also requires a new mutating ID since each number/key pair is valid for one use. Trent supplies Alice with the key K_{P1} for the message P_1, a new secret number (N'_A), and a new secret key (K'_A). Trent encrypts the three elements with Alice's previous secret key and sends the message back to Alice.

$$T \rightarrow A: E(K_A, N'_A K'_A K_{P1})$$

[0096] Alice notes her new secret number and key received from Trent and encrypts the hash of the message P_1 with her new secret key K'_A, prepends N'_A and sends the message to Carol.

$$A \rightarrow C: N'_A E(K'_A, H(P_1))$$

[0097] Upon receiving the message from Alice, Carol encrypts the message with her secret key K_C, prepends her secret number N_C, and sends the message to Trent.

[0098] $C \rightarrow T: N_C E(K_C, N'_A E(K'_A, H(P_1)))$

[0099] Trent realizes that N_C belongs to Carol and can decrypt the outer encryption performed by Carol. Trent can also decrypt the inner encryption performed by Alice since he recognizes N'_A as being Alice's secret key. Once decrypted, Trent determines that $H(P_1)$ is the hash of the message Alice previously sent and that he previously created a key K_{P1} for. Trent then generates a receipt for Alice indicating that she may send Carol the message P_1 and sends Carol the key previously generated for the message P_1 and the hash of the message P_1. The receipt Trent generates for Alice includes the hash of the message Carol is requesting a decryption key for, in the current example, $H(P_1)$ and the identity of Carol C_{id}. By identifying both the message and the entity receiving the message Alice can be guaranteed

WAVELYNX EX. 1003, Pg. 796
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

that the correct receiver receives the correct message. Alice may choose not to send the message P_1 based on the information contained in the receipt generated by Trent if Carol was not the intended receiver or if the hash of the message is not the intended message for Carol. If Alice chooses not to send the message P_1 to Carol, the message P_1 is not compromised, even though Carol has received the decryption key K_{P1} from Trent for the message. The key Carol receives, which Trent generated, contains no information concerning the message which it would encrypt. Therefor, if Alice does not send the message P_1, Carol will not be able to deduce the contents of the message P_1 simply by having the key K_{P1}. Trent also generates new number/key pairs for both Alice and Carol since both Alice and Carol have used their current secret key and number. Trent concatenates the receipt (Alice's new secret number N''_A and secret key K''_A) encrypts the concatenated elements with Alice's current secret key K'_A, and sends the message to Alice. Trent concatenates the decryption key K_{P1} for the message P (Carol's new secret number N'_C and secret key K'_C) encrypts the concatenated elements with Carol's current secret key K_C, and sends the message to Carol.

$$T \rightarrow A: E(K'_A, N''_A\, K''_A\, H(P_1) C_{id})$$

$$T \rightarrow C: E(K_C, N'_C\, K'_C\, K_{P1} H(P_1))$$

[00100] Upon receiving the message from Trent, Alice can decode the message and determine if the hash $H(P_1)$ returned to her by Trent in the receipt is the same that she sent to Carol. This provides Alice with an opportunity to double-check the message and the intended receiver once more before sending the message. As described above, if any part of the receipt appears fraudulent, for example, if the hash of the message is not the message Alice wished to send Carol or if a receiver other than Carol is requesting decryption keys for the message, Alice can simply not send the message P_1 encrypted with the key Trent generated. If everything appears correct, Alice can encrypt the message P_1 with the key K_{P1} generated by Trent and send the encrypted message to Carol.

$$A \rightarrow C: E(K_{P1}, P_1)$$

[00101] Once Carol has received the encrypted message P_1 from Alice and the decryption key from Trent, Carol can recover the message P_1 by applying the received decryption key to the received encrypted message.

WAVELYNX EX. 1003, Pg. 797
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

$$P_{recovered} = D(K_{P1}, E(K_{P1}, P_1))$$

[00102] Carol can also verify the message received from Alice, if Carol knows the function that Alice initially used to create the label for the message. Specifically, Carol can compare the hash of $P_{recovered}$ with the hash of the message P_1 received from Alice. If the two hashes are identical then Carol can conclude that the encrypted message received from Alice is associated with the initial message hash $H(P_1)$ sent to both Carol and Trent. If the two hashes are not identical, Carol would have reason to believe that fraudulent behavior is at work in the content distribution system.

[00103] The email example presented above can be extended to more general problems. The protocol could be used, for example, to authenticate a computer to use the resources provided by a server across a network. Many users connected on a network use applications that depend on virtual private networks ("VPNs") for secure communication. The security of VPNs is built into the network layer of the communication model as defined by the Open Systems Interconnection ("OSI") Model. By using mutating IDs at the application layer, one could create virtual private applications ("VPAs") instead of VPNs. With the use of VPAs, email, file systems, and other business level end-user applications could authenticate without the need for complex networking. By securing communication at the application layer, more security is provided at the local area network ("LAN") layer. This also simplifies wide area networks ("WANs") and LAN security to end-users.

[00104] Another use of mutating IDs is in authenticating participants to a transaction. Let Alice be a server with resources that Carol, a client computer can utilize. Let Bob represent a user of the client computer Carol who instructs Carol to utilize particular resources. Trent remains the authenticator for the protocol. Assume that Alice, Carol, and Bob all have a secret number N and a secret key K as described previously. Further, assume that Trent knows all secrets and Alice, Carol, and Bob do not know each other's secrets.

[00105] Since Alice needs to service many clients at once she needs a large list of number/key pairs. Assuming that Alice already has one number/key pair, she can negotiate with the authenticator, Trent, to get many number/key pairs. Alice first needs to prove to Trent that she is the appropriate server. To do this, she encrypts some identifier that Trent will recognize as belonging to Alice with a request for x number/key pairs.

WAVELYNX EX. 1003, Pg. 798
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

$A \rightarrow T$: $N_A E(K_A, A_{id}$ Send x number/key pairs)

[00106] Once Trent validates the request, he generates the number/key pairs and encrypts them with K_A to send the list of pairs back to Alice as shown in Fig. 3B.

$T \rightarrow A$: $E(K_A, N^1{}_A K^1{}_A N^2{}_A K^2{}_A \ldots)$

[00107] Alice can now destroy the number/key pair N_A/K_A and Trent marks them as used. This protocol can be run at any time to ensure that a server has enough number/key pairs to service requests.

[00108] A user, Bob, supplies his identifying credentials (e.g., a password, a user identifier) to client software on the client computer, Carol, who encrypts Bob's credentials and her own identity with her current secret key and prepends her current number. This is then sent to the requested server, Alice.

$C \rightarrow A$: $N_C E(K_c, B_{cred} C_{id})$

[00109] Alice then encrypts the received message with one of her keys and prepends the associated number. This is then sent to the authenticator, Trent.

$A \rightarrow T$: $N_A E(K_A, N_C E(K_c, B_{cred} C_{id}))$

[00110] Trent can unwrap the encryption of the message to determine that Bob would like to use services resident on the server, Alice. At this point Trent can verify the identity of Bob and Carol and validate that both Bob and Carol are authorized to use the server, Alice. After identity and authorization credentials are checked and approved, Trent can then generate two messages. The first message is destined for Carol, the client computer, and contains, encrypted with K_c, a new number/key pair, the identity of the server, Alice, and a session key K_S. The second message is destined for Alice, the server, and contains the identity of the user, Bob, the identity of the client computer, Carol, and K_p. All the components are then encrypted with K_A.

$T \rightarrow C$: $E(K_C, A_{id} N'_c K_c K_S)$

$T \rightarrow A$: $E(K_A, B_{id} C_{id} K_S)$

WAVELYNX EX. 1003, Pg. 799
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00111] At this point, the client computer, Carol, knows that the key K_S is safe to use to encrypt all communication with the server, Alice. Furthermore, Alice knows that the identity of the client computer, Carol, and the user, Bob, have been validated by Trent.

[00112] As should be apparent, secure validation between entities within an insecure system using the above embodiment of the invention requires a minimum number of steps. In order for Carol to initiate communication with Alice securely, she sends a single message to Alice who then forwards the request to Trent for validation. After the request has been confirmed, Trent sends both Carol and Alice a mutating ID to use to communicate. By allowing Alice to acquire her number/key pairs needed to issue to a requesting entity at any time through the protocol the number of steps required is also reduced. Alice does not need to wait for an entity to request service to ask for a number/key pair from the authenticator nor does an entity need to wait for Alice to acquire a number/key pair one at a time on a need basis.

[00113] In comparison, current systems used to establish validated communication between two entities require a larger number of steps that when applied to systems with many services and entities requesting service grow at a high rate. Some current systems require an entity to validate its request with multiple entities before it is allowed to directly communicate with a service. Even the number of steps required for simple tasks, such as logging on, can grow quadratically in respect to the number of components, such as entities and services, associated with the system. Many current systems also rely on timestamps and synchronization of clocks between all participating parties. If any internal clocks differ among the entities, entities may be required to re-authenticate themselves and gain new session keys, which further increases the communication needed to validate the use of a service. When a large number of entities require validation with a large number of services the overhead presented with current systems is believed to be greater than that required by the above described mutating ID system under similar conditions.

[00114] Another exemplary embodiment of the protocol involves all of the four entities discussed previously. In this embodiment, Alice or A performs the role of the content provider 22, entity Bob or B performs the role of the service provider 24, entity Carol or C performs the role of the consumer 26, and entity Trent or T performs the role of the authenticator 28. Since the proposed protocol relies on a trusted authority, Alice, Bob, and

WAVELYNX EX. 1003, Pg. 800
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Carol trust Trent. Also, Alice has two secrets: a number (N_A) and a key for some symmetric cipher (K_A). Bob also has two secrets: a number (N_B) and a key for some symmetric cipher (K_B). Similarly, Carol has a secret number (N_C) and key (K_C). Further, all assigned number and keys are assigned and known by Trent.

[00115] For the purposes of this example only, assume Alice is a movie producer with a movie M. Alice would like to use Bob's cable company (e.g., video-on-demand) to distribute the movie M, and Carol, one of Bob's customers, wishes to receive and watch the movie M. Of course, Alice could represent any entity needing to distribute a variety of content items. Alice could distribute email messages, music, images, data, etc. using the same protocol described herein.

[00116] To start, if Alice wishes to distribute her content, she requests numerous number/key pairs from Trent. To do this Alice creates a message P_2 indicating the request, encrypts the request with her K_A, and prepends N_A. By encrypting the request with her secret key, Trent can verify that only authorized persons are granted number/key pairs. Alice then sends the message to Trent.

$$A \rightarrow T: N_A E(K_A, P_2)$$

[00117] Upon receiving the message from Alice, Trent decodes the request for numerous number/key pairs since he knows Alice's secrets. Referring to Fig. 3B, Trent generates a list of licenses and sends the entire list back to Alice encrypted with Alice's key K_A, which is known to both Trent and Alice.

[00118] $T \rightarrow A: E(K_A, \{N^i_A, K^i_A)\})$

[00119] Alice now coordinates with Trent to receive a key to encrypt each copy of her movie M. Alice generates a label for M, in the current embodiment a hash of M is used, which is used as an identifier for the movie M. Alice picks some arbitrary number/key pair j and encrypts the hash of M with K^j_A, prepends N^j_A, and sends the message to Trent.

$$A \rightarrow T: N^j_A E(K^j_A, H(M))$$

WAVELYNX EX. 1003, Pg. 801
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00120] Alice could also include a unique identifier M_{id} for the movie M that is known by the other entities of the protocol in the message sent to Trent. The addition of the identifier provides Trent with a mechanism to match and authorize requests for encrypted content.

$$A \rightarrow T: N^i_A E(K^i_A, H(M) M_{id})$$

[00121] Upon receiving the message from Alice, Trent generates a key K_M associated with the hash of the movie M and sends it back to Alice encrypted with K^i_A. Trent records that the sent key K_M is associated with the hash of the movie M (and the unique identifier M_{id} if provided). Even though Trent is generating the key for encrypted content, he never receives the content. All Trent receives is the hash of the movie M and possibly a unique identifier that is known by all entities of the system. He has no other useful information regarding the content provided by the content provider. Trent has no information that would allow him to directly distribute content without the intervention and authorization of the content provider.

[00122] $T \rightarrow A: E(K^i_A, K_M)$

[00123] Alice can now encrypt M with K_M with the confidence that only she and Trent know K_M. She encrypts M with K_M and sends it to Bob.

[00124] $A \rightarrow B: E(K_M, M)$

[00125] Alice also sends Bob corresponding primary licenses for the encrypted copy. We call the license sent by the content provider a primary license since it may be further authenticated by a service provider before it is distributed. Alice creates each primary license by encrypting the hash of the movie M with a key K^k_A and prepending a number N^k_A of an arbitrarily chosen number/key pair k from the list generated by Trent. Alice sends each primary license to Bob.

[00126] $A \rightarrow B: N^k_A E(K^k_A, H(M))$

[00127] Although Bob has received both the encrypted content and the corresponding primary license, he still is unaware of the decryption key and so cannot distribute copies of M that Alice would be unaware of. He could distribute either the encrypted content or the primary license to as many consumers as he wished without interaction from Alice or Trent, but the consumers would be unable to view encrypted content or use the primary license.

WAVELYNX EX. 1003, Pg. 802
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Bob also has no information regarding the content Alice is providing from either the encrypted copies or the primary license. By providing Bob with licenses, Alice can restrict the number of copies of content Bob can deliver since each license will be marked as used by Trent after it has been issued to a consumer. With a valid license from Alice, a service provider and consumer cannot gain content. Alice should have at least one number/key pair remaining that is known to Trent. If Alice needs more number/key pairs, she could use the one remaining number/key pair to request more from Trent.

[00128] Once Bob has received the encrypted copies of M and the corresponding primary licenses signed by Alice, he further authenticates each primary license sent by Alice to place his ownership on the content before the license is distributed to consumers. Having Bob further authenticate each primary license is not required, but protection is increased for Alice's movie M since Alice can ensure that requests for content items have been properly initiated through authorized distributors. In order to add greater authorization to each primary license, Bob first requests numerous number/key pairs from Trent in a manner similar to how Alice acquired numerous number/key pairs defined above. Bob creates a message P_3 indicating the request, encrypts the request with his K_B, and prepends N_B. Bob then sends the message to Trent.

$$B \rightarrow T: N_B E(K_B, P_3)$$

[00129] When Trent receives the message from Bob, he looks up the key associated with N_B, since he knows Bob's secrets, and decodes the request. He then generates the numerous number/key pairs requested and sends them back to Bob encrypted with K_B.

$$T \rightarrow B: E(K_B, \{N^i_B, K^i_B)\})$$

[00130] Clearly, Bob could request the number/key pairs before or after receiving the encrypted copies from Alice. He does not need to send anything that he receives from Alice in order to request number/key pairs from Trent. He simply needs to be identifiable to Trent as an entity that can request and receive number/key pairs.

[00131] After Bob has received the number/key pairs from Trent he can create a distributable license, a license that can be distributed to requesting consumers, by choosing an arbitrary number/key pair m from the list Bob received from Trent and encrypting an

WAVELYNX EX. 1003, Pg. 803
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

encrypted identifier received from Alice with the key K^m_A and prepending the number N^m_A from the chosen number/key pair m.

$$N^m_A E(K^m_A, N^k_A E(K^k_A, H(M)))$$

[00132] Bob repeats this procedure for however many distributable licenses he wishes to distribute or is authorized to distribute by Alice.

[00133] Now that Bob has the content, we can address the next step of the process. Carol wishes to view content M and sends a request for M to Bob that includes the unique identifier for the movie M.

$$C \rightarrow B: Send\ M_{id}$$

[00134] Carol's request for content may be encoded to protect it from corruption. Without encoding the request, Carol's request could be intercepted and changed, and Carol may receive content that she did not intend to purchase. Carol can protect her request by encoding it with a secret key shared between Bob and her. Carol could also send her request to Bob using one of the previously disclosed embodiments of the invention. Overall, Carol could use any security mechanism to secure her request.

[00135] Bob receives the request from Carol and replies by choosing one of the distributable licenses he previously generated for the movie M identified by the unique identifier M_{id} included in Carol's request. Bob sends the distributable license to Carol and notes that the license has been used and sent to Carol.

$$B \rightarrow C: N^m_B E(K^m_B, N^k_A E(K^k_A, H(M)))$$

[00136] After receiving the distributable license from Bob, Carol encrypts it with her key K_C and prepends her number N_C. Carol sends the encrypted distributable license to Trent.

$$C \rightarrow T: N_C E(K_C, N^m_B E(K^m_B, N^k_A E(K^k_A, H(M))))$$

[00137] Upon receiving the encrypted distributable license from Carol, Trent can unwrap all of the encryption, since he knows Carol's, Bob's, and Alice's secrets, and can determine that Carol received a valid license and needs the decryption key for the content M. Recovering this information allows Trent to generate a receipt for Bob by concatenating

WAVELYNX EX. 1003, Pg. 804
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Bob's primary license with the identity of Carol and encrypting it with the key it was originally signed with K^m_B and prepending the number originally prepended when Bob signed the license N^m_B. Trent sends the receipt to Bob.

$$T \rightarrow B:\ N^m_B\, E(K^m_B, N^k_A\, E(K^k_A, H(M))\, C_{id})$$

[00138] The receipt Bob receives from Trent not only informs Bob that Carol has been authorized as a consumer, but it also directs Bob to send the matching encrypted copy of the content specified by the hash $H(M)$ to Carol. Bob is also able to verify that the distributable license Carol forwarded to Trent is the same distributable license that he initially sent Carol. Carol cannot substitute a different distributable license nor can another consumer use a distributable license that Carol received from Bob. After Bob has inspected the receipt from Trent, he sends the requested encrypted copy to Carol.

$$B \rightarrow C:\ E(K_M, M)$$

[00139] Carol also requires both the decryption key necessary to decipher the received encrypted copy of M and a new number/key pair so she is able to make a second request for content, since she consumed her old mutating ID with her request for the movie M. Trent, knowing the content requested M and the key used to encrypt a copy of M, since he created it, sends the key to Carol along with a new number/key pair to be used for future requests. Trent concatenates the new number/key pair with the needed key for the encrypted copy of M and encrypts the elements with Carol's current key K_C. Trent does not need to prepend Carol's assigned number N_C since Carol only has one number/key pair and does need to look up a matching key given a number from a list of number/key pairs as does Alice and Bob. Thus, Trent sends the following to Carol.

$$T \rightarrow C:\ E(K_C, N'_C K'_C K_M)$$

[00140] Alternatively, if Alice has provided a unique identifier to Trent for the encrypted content, which is known by Carol, Trent can ask Carol for a final commitment for the content before sending the decryption key. Trent can send Carol a new number/key pair and the unique identifier M_{id} corresponding to the content specified in the distributable license Trent received from Carol.

$$T \rightarrow C:\ E(K_C, N'_C K'_C M_{id})$$

WAVELYNX EX. 1003, Pg. 805
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Since the unique identifier is known by all participating entities, Carol can verify that the decryption key she is about to receive from Trent is for the content that she requested. If the identifier does not correspond to the content she requested, Carol knows that her initial request either contained faults or was compromised, and she can stop Trent from providing a decryption key for the content. By requesting final authorization before sending a decryption key, Carol may avoid charges for content that she did not want. Otherwise, if the identifier does match the content requested, Carol can authorize Trent to send the decryption key.

$$C \rightarrow T: E(K_C, Authorize)$$

$$T \rightarrow C: E(K_C, K_M)$$

[00141] Carol now has everything she needs to watch the movie M.

[00142] Trent could also generate a receipt for Alice at this point, allowing her to track a consumer to particular content. Trent can concatenate the label of M and Carol's identity C and encrypt all the elements with Alice's key K^k_A and prepend the number N^k_A used in the license sent to Carol.

$$T \rightarrow A: N^k_A\, E(K^k_A, H(M)C_{id})$$

[00143] Trent also can mark all the number/key pairs encountered during the transaction $(\{K^k_A, N^k_A\}, \{K^m_B, N^m_B\}, \{K_C, N_C\})$ as used so that should he encounter one of them in the future, he will know that something is amiss. That is, Trent can ensure that each number/key pair is used once and only once. Accordingly, any reuse indicates that an entity or person is attempting to defraud the system.

[00144] At this point Alice knows that Bob distributed a copy of the movie M to Carol. However, Alice cannot track a specific copy of the movie M to Bob or Carol. If an illegally distributed copy of the movie M appears in the market it cannot be traced to a specific service provider or consumer. Alice can only generate a list of entities that licensed the content. This freedom from tracking encourages consumers who legally purchase a copy of the movie M to illegally distribute the acquired copy.

[00145] In another embodiment, Alice can trace a particular copy of a content item to a service provider and a consumer by uniquely watermarking each copy created and

WAVELYNX EX. 1003, Pg. 806
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

distributed. By providing Alice a means to track specific copies to specific consumers, consumers who purchase content legally may be deterred from dishonestly replicating the acquired copy.

[00146] In one watermarking embodiment, the entity Alice or *A* again performs the role of the content provider 22, entity Bob or *B* performs the role of the service provider, entity Carol or *C* performs the role of the consumer 26, and entity Trent or *T* performs the role of the authenticator 28.

[00147] Similar to the non-watermarking embodiment with four entities, Alice requests numerous number/key pairs from Trent with an encrypted message P_4.

$$A \rightarrow T: N_A E(K_A, P_4)$$

[00148] When Trent receives the message from Alice he generates a list of number/key pairs and sends the entire list back to Alice.

$$T \rightarrow A: E(K_A, \{N^i_A, K^i_A\})$$

[00149] Once Alice receives the numerous number/key pairs she can choose to further protect her content by applying a unique watermark to each copy of *M*. Alice generates a list of watermarks $D_1, D_2, \ldots, D_n$ and a watermarking scheme *W* for her movie *M*. Alice can then generate many different hashes for *M*, one for each watermark. Consider:

$$M_1 = W(D_1, M)$$

$$M_2 = W(D_2, M)$$

$$M_3 = W(D_3, M) \ldots$$

[00150] In contrast with the non-watermarking embodiment, Alice can now coordinate with Trent to receive a key to encrypt each watermarked copy of her movie *M* instead of receiving a single key to encrypt her movie *M*. Alice generates a label for *M* and a hash of each watermark $D_1, D_2, \ldots, D_n$, and picks some arbitrary number/key pair *j*. Alice encrypts both the hash of *M* and hash of an individual watermark D_1 with K^j_A, prepends N^j_A, and sends the message to Trent.

WAVELYNX EX. 1003, Pg. 807
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

$$A \rightarrow T: N^{j}_{A} E(K^{j}_{A}, H(M)H(D_1))$$

[00151] As noted in the previous embodiment, Alice could also provide a known identifier for the movie M_{id} in the message to Trent so that Trent can use the identifier later to request final authorization from a consumer.

$$A \rightarrow T: N^{j}_{A} E(K^{j}_{A}, H(M)H(D_1)M_{id})$$

[00152] Upon receiving the message from Alice, Trent generates a key K_{M1} associated with the hash of the movie M and sends it back to Alice encrypted with K^{j}_{A}. Trent records that the sent key K_{M1} is associated with the hash of the movie M and the hash of the watermark D_1 (and the known identifier M_{id} if provided). Even though Trent is generating the key for encrypted, watermarked content, he never receives the decrypted, watermarked content. Again, Trent has no useful information regarding the content provided by Alice.

$$T \rightarrow A: E(K^{j}_{A}, K_{M1})$$

[00153] Alice can now encrypt M_1, the watermarked content and send it to Bob.

$$A \rightarrow B: E(K_{M1}, M_1)$$

[00154] Alice also sends Bob the corresponding primary license for the encrypted, watermarked copy. Alice creates the primary license in the same format as she did in the non-watermarking embodiment but concatenates the hash of the watermark to the hash of the movie M.

$$A \rightarrow B: N^{k}_{A} E(K^{k}_{A}, H(M)H(D_1))$$

[00155] Once Bob has received the encrypted, watermarked copies of M and the corresponding primary licenses signed by Alice, he authenticates each primary license as he did in the non-watermarking embodiment. As noted before, Bob's authentication of each primary license is not necessary, but it does offer added protection and security with the communication system. Bob first acquires numerous number/key pairs from Trent by sending Trent a message P_5.

$$B \rightarrow T: N_B E(K_B, P_5)$$

WAVELYNX EX. 1003, Pg. 808
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00156] As before, Trent receives the message from Bob, generates the numerous number/key pairs requested, and sends them back to Bob encrypted with K_B.

$$T \rightarrow B: E(K_B, \{N^i_B, K^i_B)\})$$

[00157] Again, Bob could request the number/key pairs before or after receiving the encrypted, watermarked copies from Alice.

[00158] Similar to the non-watermarking embodiment, after Bob has received the number/key pairs from Trent he can create a distributable license.

$$N^m_A\, E(K^m_A, N^k_A\, E(K^k_A, H(M)H(D_1)))$$

[00159] Bob repeats this procedure for however many distributable licenses he wishes to distribute or is authorized to distribute by Alice.

[00160] When Carol wishes to view content M, she sends a request for M to Bob.

[00161] $C \rightarrow B: Send\ M_{id}$

[00162] As noted before, further encryption could be applied to Carol's request to ensure that her message is not compromised.

[00163] As before, Bob receives the request from Carol and replies by choosing one of the distributable licenses he previously generated. Bob sends the distributable license to Carol.

$$B \rightarrow C: N^m_B\, E(K^m_B, N^k_A\, E(K^k_A, H(M)H(D_1)))$$

[00164] After receiving the distributable license from Bob, Carol encrypts it with her key K_C and prepends her number N_C exactly as she did in the non-watermarking embodiment and sends the encrypted distributable license to Trent.

$$C \rightarrow T: N_C\, E(K_C, N^m_B\, E(K^m_B, N^k_A\, E(K^k_A, H(M)H(D_1))))$$

[00165] Upon receiving the encrypted distributable license from Carol, Trent can unwrap all of the encryption, since he knows Carol's, Bob's, and Alice's secrets, and can determine that Carol received a valid license and needs the decryption key for the content M

WAVELYNX EX. 1003, Pg. 809
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

watermarked with watermark D_1. Similar to the non-watermarking embodiment, Trent generates a receipt for Bob.

$$T \rightarrow B: N^m_B E(K^m_B, N^k_A E(K^k_A, H(M)H(D_1))C_{id})$$

[00166] The receipt Bob receives from Trent not only informs Bob that Carol has been authorized as a consumer, but it also directs Bob to send the matching encrypted copy of the content specified by the hash $H(M)$ and the hash $H(D_1)$ to Carol.

$$B \rightarrow C: E(K_{M1}, M_1)$$

[00167] As in the non-watermarking embodiment, Trent sends the decryption key for the encrypted, watermarked copy of the movie Carol receives from Bob along with a new number/key pair Carol can use for future requests.

$$T \rightarrow C: E(K_C, N'_C K'_C K_{M1})$$

[00168] As mentioned before, Trent could additionally request final authorization at this point if he knows an identifier, which is known to all entities, for the content requested. Trent can provide the identifier to Carol who can verify that the decryption key she is about to receive is for the content she requested. Once Trent receives Carol's verification he can send the needed decryption key.

[00169] Carol now has everything she needs to watch the movie M.

[00170] As before, Trent could also generate a receipt for Alice at this point, allowing her to track a consumer to a particular watermarked copy. Trent can concatenate the label of M, and the hash of the watermark $H(D_1)$, and Carol's identity C and encrypt all the elements with Alice's key K^k_A and prepend the number N^k_A used in the license sent to Carol.

$$T \rightarrow A: N^k_A E(K^k_A, H(M)H(D_1)C_{id})$$

[00171] Trent also can mark all the number/key pairs encountered during the transaction $(\{K^k_A, N^k_A\}, \{K^m_B, N^m_B\}, \{K_C, N_C\})$ and the watermark D_1 as used so any future use of them can be recognized as fraudulent behavior.

WAVELYNX EX. 1003, Pg. 810
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00172] Alternatively in another watermarking embodiment, Bob, rather than Alice, applies the watermark to the content copies. Alice sends Bob a single encrypted version of the movie M and may also provide a list of watermarks to apply. Bob then coordinates with Trent to receive the decryption key for the content. After receiving the decryption key Bob decodes the encrypted content, applies watermarks to the decrypted content, and encrypts the watermarked content with another encryption key different from the first used by Alice. By allowing Bob to watermark the distributable content, less communication is needed and less data is transmitted between the content provider and the service provider. The service provider may also perform the decryption and watermarking on the fly when the request for content is made. By watermarking the content when needed, the amount of storage required by the service provider is reduced since he does not need to retain multiple encrypted, watermarked versions of content. Since Alice provides her content encrypted but unwatermarked, another entity of the protocol must be trusted to decrypt Alice's content, apply a watermark, and encrypt the watermarked content again before it is sent to consumers. Bob could be trusted to perform the decryption and watermarking, but Alice may also insist that Bob use a special piece of hardware that protects the security of her content. By requiring the decryption and watermarking to occur in hardware (which is inaccessible by the service provider) in real-time, the security of Alice's content is preserved. The content is sent to special hardware of the service provider that decodes the content, applies a watermark, and encodes the content again. The hardware prevents the service provider from accessing the decrypted content since the content enters the hardware encrypted and exits the hardware watermarked and encrypted. For the protocol to work, the content provider must trust the security of the special hardware. The authorized special hardware, or decoding entity, that will decode and watermark content provided by Alice is denoted as B' or Brenda in the following detailed example.

[00173] As before Alice wishes to distribute her movie M and wants to track consumers to a specific copy of her movie. However, Bob does not want to receive and store multiple encrypted and watermarked copies of Alice's content. In order to reduce communication and data traffic between Alice and Bob, Bob will watermark and encrypt the content provided by Alice.

[00174] Alice begins by requesting numerous number/key pairs from Trent that she will use, in contrast to the previous two embodiments, to create a list of encrypted watermarks.

WAVELYNX EX. 1003, Pg. 811
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

After Alice receives the number/key pairs from Trent, she then requests an encryption key for the movie *M*. In addition, Alice may specify who should be allowed to decode the movie in order to apply a watermark in her request to Trent. In the current example Brenda will be authorized to apply watermarks. Alice may also include a known identifier for the movie M_{id} that can be used by Trent to validate requests before sending decryption information.

$$A \rightarrow T: N_A E(K_A, H(M)M_{id}B')$$

[00175] Trent responds with an encryption key for the movie K_M.

$$T \rightarrow A: E(K_A, K_M)$$

[00176] Once Alice receives the encryption key K_M she can encrypt the movie *M* and a hash of the movie *H(M)* and send the encrypted content to Bob.

$$A \rightarrow B: E(K_M, H(M)M)$$

[00177] Alice also generates a list of watermarks $D_1, D_2, \ldots, D_n$ to be applied to the copies of the encrypted content. Each watermark will be concatenated with a hash of the movie *H(M)* and possibly a known identifier for the movie M_{id}. Each concatenated list is encrypted with one of the number key pairs Alice received from Trent. Alice sends the list of encrypted watermarks to Bob. Alternatively, Alice could allow Bob to generate the watermarks to be applied to her content. By having Bob create the watermarks even less communication and data is transmitted between Alice and Bob.

$$A \rightarrow B: N^1{}_A E(K^1{}_A, H(M)D_1M_{id}), N^2{}_A E(K^2{}_A, H(M)D_2M_{id}), \ldots, N^3{}_A E(K^3{}_A, H(M)D_3M_{id})$$

[00178] Bob now has a list of encrypted watermarks and the encrypted movie *M* that can only be decrypted by Brenda since that is what was specified by Alice in her request to Trent. In order to prepare to distribute content, Bob and Brenda request numerous number/key pairs from Trent.

[00179] Once Bob receives a request from Carol for the movie *M*, Bob provides Brenda with an encrypted watermark that he received from Alice encrypted with an arbitrarily chosen number/key pair from the list provided by Trent.

$$B \rightarrow B': N^a{}_B E(K^a{}_B, N^1{}_A E(K^1{}_A, H(M)D_1M_{id}))$$

WAVELYNX EX. 1003, Pg. 812
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00180] Brenda chooses an arbitrary number/key pair from the list generated by Trent and further encrypts the doubly-encrypted watermark received from Bob. Brenda then sends the thrice-encrypted watermark to Trent.

$$B' \rightarrow T: N^a{}_{B'}, E(K^a{}_{B'}, N^a{}_B E(K^a{}_B, N^l{}_A E(K^l{}_A, H(M) D_1 M_{id})))$$

[00181] Trent decodes all of the encryption, since he knows everyone's secrets, and returns the hash of the movie $H(M)$, the watermark D_1, the key to decode the movie K_M, a new key to encode the movie $K_{M'}$, and Bob's identity B_{id} to Brenda.

$$T \rightarrow B': E(K^a{}_{B'}, H(M) D_1 K_M K_{M'} B_{id})$$

[00182] Brenda also encrypts the same doubly-encrypted watermark received from Bob with another arbitrarily chosen number/key pair. Brenda sends the second thrice-encrypted watermark to Carol.

$$B' \rightarrow C: N^b{}_{B'}, E(K^b{}_{B'}, N^a{}_B E(K^a{}_B, N^l{}_A E(K^l{}_A, H(M) D_1 M_{id})))$$

[00183] Carol encrypts the received message with her single number/key pair and, as before, forwards the encrypted message to Trent.

$$C \rightarrow T: N_C\, E(K_C, N^b{}_{B'}, E(K^b{}_{B'}, N^a{}_B E(K^a{}_B, N^l{}_A E(K^l{}_A, H(M) D_1 M_{id}))))$$

[00184] Trent unwraps all of the encryption and verifies with Carol that she did intend to get the movie identified in the message.

$$T \rightarrow C: E(K_C, N'_C K'_C M_{id})$$

[00185] Carol responds with authorization if the content identifier M_{id} corresponds to the content she requested.

$$C \rightarrow T: E(K_C, Authorize)$$

[00186] Trent then sends the decryption key K_M to Carol and generates a receipt for Bob. Bob's receipt contains the encrypted watermark Bob initially sent to Carol along with Carol's identity.

$$T \rightarrow B: N^a{}_B E(N^l{}_A E(K^l{}_A, H(M) D_1 M_{id}) C_{id})$$

WAVELYNX EX. 1003, Pg. 813
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00187] Trent also generates a receipt for Alice so that she can associate a watermarked copy of the movie M with Carol. Alice's receipt includes the hash of the movie $H(M)$, the watermark applied D_1, and the identity of the consumer C_{id}. Alice's receipt is encrypted with the number/key pair Alice used to encrypt the watermark that will be associated with the content licensed to Carol.

$$T \rightarrow A: N^1{}_A E(K^1{}_A, H(M) D_1 C_{id})$$

[00188] Upon receiving the receipt from Trent, Bob sends Brenda the encrypted movie that he received from Alice.

$$B \rightarrow B': E(K_M, H(M) M)$$

[00189] Brenda decrypts the encrypted movie since she received the decryption key K_M from Trent. Brenda decodes the movie to recover the movie M, applies the watermark D_1, and encrypts the watermarked movie $W(D_1, M)$ with the new encryption key $K_{M'}$. The hash of the movie $H(M)$ Alice encrypted with the movie allows Brenda to check that the movie has not been switched for another and that the decryption key received from Trent is the matching key for the encrypted content. Once the movie M has been decrypted, watermarked, and encrypted, Brenda sends the encrypted, watermarked content to Carol.

$$B' \rightarrow C: E(K_{M'}, W(D_1, M))$$

[00190] Carol has already received the decryption key from Trent, so she can decrypt the encrypted, watermarked content and view the movie M.

[00191] If Carol decides to make and distribute illegal copies of the content she received, any illegal copy found in the market that Carol distributed can be traced back to Carol by the watermark applied to the content (regardless of the particular embodiment of the invention used to apply the watermark). Both Bob and Alice receive receipts that link a particular watermark to a particular consumer. Knowing this, Carol may be deterred from distributing illegal copies of content, since the copies can be traced back to her. Carol may also keep a closer guard of her copy so that no one else takes her content and distributes it illegally. The rights of the content provider and service provider are enhanced through the use of the watermarking.

WAVELYNX EX. 1003, Pg. 814
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00192] Another feature that could reduce the amount of illegal copying is requiring a real-time decode of the movie or content. Depending on the hardware implementation chosen, a real-time decode may be possible that would require any recording of content downloaded via the proposed system to take the same amount of time required to watch or consume the content at its intended playback speed. The decode output could also be implemented as an analog signal that would reduce the fidelity of subsequent recordings.

[00193] Additional aspects of embodiments of the invention will now be described with respect Figs. 4-16.

[00194] Fig. 5 illustrates the overall process of the consumer 26 requesting content in one embodiment of the invention. First, the consumer 26 makes a request for a particular piece of content (step 50). The request may include an identifier or label associated with the particular piece of content that the consumer found using a search engine or catalog of available content. The label may also be a title of the content such as a movie title or song title. The service provider 24 then requests an encrypted license 52 from the content provider 22 that is associated with the desired content. The service provider sends the license 52 to the customer 26 (step 54). The customer's hardware (e.g., a set top box), encrypts the license 52 with a mutating ID (previously received from the authenticator 28) and sends the now doubly encrypted license 52 to the authenticator 28 (step 56). The authenticator 28 checks the license 52 for validity. If the license is valid, the authenticator 28 so informs the service provider (step 58). The service provider then sends the content to the consumer 26 (step 60). At the same or nearly the same time, the authenticator 28 sends decoding information to the consumer 26 (step 62).

[00195] As should be apparent from the discussion above, the license 52 undergoes multiple transformations, in the sense that the license is encrypted multiple times. Fig. 4 illustrates this process. The content provider creates the license 52 upon receiving a request for particular content. The license is an encrypted version of the randomly determined, secret identifier that the content provider created for the subject content. The now once encrypted version of the license (license 63 in Fig. 4) is sent to the service provider 24. The service provider 24 encrypts the license again (license 64 in Fig. 4) and the now doubly encrypted version is sent to the consumer 26. The doubly encrypted license is encrypted with the mutating ID of the consumer to create a thrice-encrypted version (license 65 in Fig. 4).

WAVELYNX EX. 1003, Pg. 815
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Authentication or verification of the license and ultimately delivery and decoding of information is then performed as noted in the discussion above with respect to Fig. 5. Fig. 6 provides another illustration of the process, including the multiple encryption steps.

[00196] Fig. 7 illustrates that embodiments of the invention may include multiple service providers. Furthermore, it is envisioned that service providers may have their own content or licenses to particular content that other service providers may desire. Thus, various requests for and transfers of licenses (as shown) may occur between service providers in order to make a large variety of content available to the ultimate consumers of the various service providers participating in the system 20.

[00197] Fig. 8 illustrates how mutating IDs are implemented in one embodiment of the invention. As shown, the consumer 26 is assigned a first mutating ID 100. The consumer uses that mutating ID to confirm its identity (as shown in box 102, which corresponds to step 56 in Fig. 5). However, once the license 100 is used, the authenticator modifies the license 100 or, in other words, sends a new key pair to the consumer, as shown in box 104. The new key pair, in effect, creates a new mutating ID 106.

[00198] Figs. 9a and 9b illustrate one way of managing content keys. If desired, the system 20 may be set up or configured such that the content provider 22 provides a list of content keys to the authenticator 28. If the system 20 is implemented in this fashion, a request made by the consumer 26 for content is transferred (through one or more intermediary service providers and/or other content providers) to the content provider 22 (step 110 in Fig. 9b). The content provider then sends a license to the consumer 26 (again, through one or more intermediary service providers) (step 112). The consumer 26 then sends its content license to the authenticator for verification or authentication (step 114). If verified, verification information is sent to the content provider 22 (step 116). The authenticator 28 then sends the content key to the consumer 26, so that the content can be decoded and viewed (step 118).

[00199] Figs. 10a and 10b illustrate another way of managing keys. The overall process is similar to what was described above with respect to Figs. 9a and 9b, except that no keys are sent to the authenticator 28, but maintained with the content provider 22.

WAVELYNX EX. 1003, Pg. 816
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00200] Figs. 11-13 illustrate ways in which service and content providers can share content, if desired. One way of sharing content is for each content provider to hold the rights to certain content to create a license upon receiving a request for the desired content (Fig. 11). Alternatively, content providers may distribute a predetermined number of licenses to downstream providers so that they do not have to respond to each request on an individual basis and so that downstream providers can be assured of having the ability to distribute a certain number of "copies" of certain content, without having to obtain approval, which could possibly be denied, by the act of refusing to supply a license, for each individual request that the downstream provider receives from end consumers (Figs. 12 and 13).

[00201] Fig. 14 illustrates the process of generating and utilizing a license for content, where the content is watermarked. As noted above, a license for content undergoes multiple transformations, in the sense that the license is encrypted multiple times. The content provider 22 creates the license upon receiving a request for particular content or in preparation of future requests by the consumer 26. The license is an encrypted version of the randomly determined, secret identifier 130 that the content provider created for the subject content and a function of the watermark 132 applied to the copy of the content. The now once encrypted version of the license (license 134 in Fig. 14) is sent to the service provider 24. The service provider 24 encrypts the license again (license 136 in Fig. 14) and the now doubly encrypted version is sent to the consumer 26. The doubly encrypted license is encrypted with the mutating ID of the consumer to create a thrice-encrypted version (license 138 in Fig. 14). Authentication or verification of the license and ultimately delivery and decoding of information is then performed as noted in the discussion above with respect to Fig. 5 except the encrypted content and decoding information is specific to the watermarked content identified in the license. Fig. 15 provides another illustration of the process, including the multiple encryption steps.

[00202] Fig. 16 illustrates exemplary embodiments of the devices used to distribute content. The consumer 26 is illustrated as a set top device 150. The set top device 150 is coupled to a display capable of displaying content 151, and provides an interface to the set top device through a remote control 152. A user 153 uses the remote control 152 to interact with the set top device 150. The user 153 uses the remote control 152 to view a list of available content and select content to view on the display 151. The interface accessed

WAVELYNX EX. 1003, Pg. 817
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

through the use of the remote control 152 also indicates when the user 153 makes a selection of content by highlighting or flashing the name of the selected content on the display 151.

[00203] Once the user 153 has selected content, the set top device 150 generates a request for the selected content. The request is then sent to the content provider 22. The content provider 22 device is illustrated as a server that contains a processor 156, a memory module 158, and input/output module 160. The memory module 158 contains the content items of the content provider 22 and at least one mutating identifier. The memory module 158 may contain other mutating identifiers if the content provider 22 has requested numerous mutating identifiers from the authenticator 28. The memory module 158 may contain non-volatile memory such as one or more forms of ROM, one or more disk drives, RAM, other memory, or combinations of the foregoing.

[00204] The content provider 22 receives the request through the input/output module 160 and transfers the request to the processor 156. The processor 156 accesses the memory module 158 to obtain a mutating identifier that it will use to encrypt a license for the content item also stored in the memory module 158. The processor 156 encrypts the license and transfers the license to the consumer 26 through the input/output module 160.

[00205] The consumer 26 receives the encrypted license, processes the encrypted licenses (as will be explained below), and forwards the encrypted license to the authenticator 28. The authenticator is also illustrated as a server containing a processor 162, a memory module 164, and an input/output module 166. Unlike the memory module 158 of the content provider 22, the memory module 164 of the authenticator 28 does not contain content to be distributed. Again, the memory module 162 may contain non-volatile memory such as one or more forms of ROM, one or more disk drives, RAM, other memory, or combinations of the foregoing.

[00206] The processor 162 of the authenticator 28 decrypts the encrypted license received by the input/output module 166 to uncover the content requested and the entities involved in the distribution. Once the identities of the participating entities are known, the processor 162 verifies the identities and generates a receipt for the content provider 22. The processor also accesses the decryption key associated with the content specified in the license and accesses one or more new mutating identifiers from the memory module 164. The processor then directs the receipt, the one or more new mutating identifiers, and the decryption key to be sent to the respective parties by the input/output module 166. The processor 162 of the

WAVELYNX EX. 1003, Pg. 818
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

authenticator also indicates in the memory module 164 that the mutating identifiers used in the current transaction have been used and should be accepted if encountered in the future.

[00207] The consumer 26 receives the decryption key and a new mutating identifier, and the content provider 22 receives the receipt for the requested content item and a new mutating identifier if required. The content provider 22 then sends the encrypted content to the consumer 26 and can record that the consumer 26 received the content.

[00208] Once the consumer 26 receives the encrypted content from the content provider, it now has everything it needs to watch the movie. The set top device 150 can now display the movie on the connected display 151.

[00209] Fig. 17 illustrates hardware that may be used in the set top device 150. In the exemplary configuration shown, the set top device 150 includes a processor 170, a memory module 172, an input/output module 174, and a remote control module 175. The hardware may contain other modules as well.

[00210] The memory module 172 is used to hold the mutating identifier of the consumer 26 and may also be used to hold content, messages, or keys sent by the content provider 22 and/or authenticator 28. The memory module 172 may contain non-volatile memory such as one or more forms of ROM, one or more disk drives, RAM, other memory, or combinations of the foregoing.

[00211] The processor 170 is configured to generate requests, encrypt received messages, access and store data to the memory module 172, and decrypt received content. The input/output module 174 is configured to interface with the other entities of the system (i.e., the content provider 22 and authenticator 28), and the display device 151. The remote control module 175 is configured to interface with the remote control 152 used by the user 153 to initiate requests for content.

[00212] Although a processor is paired with a memory and an input/output module is illustrated for all three devices used in the distribution system, it should be apparent to one skilled in the art that hardware, software, or any combination thereof could be used to communicate and distribute content between participating entities. The processor could be an

WAVELYNX EX. 1003, Pg. 819
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

integrated circuit, a microprocessor, or any combination of hardware and software that can perform the necessary operations for distributing content.

[00213] As should be apparent, the system 20 and the protocol used to implement it can be used in a variety of applications beyond the secure distribution of content. Many different types of communications ranging from e-mail, video and multi-media conferencing, data and telemetry collection, and others may benefit from using all or part of the system 20 to enhance security and trust. Some of these additional applications are now discussed.

Confirmation of Geographic Location

[00214] As is known, innumerous human activities depend on the human participants trusting the other participants. Furthermore, participants must feel comfortable that the other participants are authentic (i.e., not imposters or frauds) and will not renege on promises or commitments made. In times when most activities took place face-to-face many concerns about authenticity were reduced. For example, before the existence of the telephone and Internet, imposters would have to physically meet and fool any person that the imposter intended to dupe. With modern communications, it is often impossible for parties to know who they are actually communicating with or where the parties are located.

[00215] Various biometric and other devices exist that can be used to address concerns of authenticity and trust and many of these can be used with or added into embodiments of the system 20 described. However, the system 20 also has inherent features that lessen concerns of authenticity. One of these is the ability to trace, at least to a relatively specific point, the location of customers ordering content.

[00216] As noted, mutating IDs are implemented in embodiments of the invention and the consumer 26 is assigned the first mutating ID 100. Subsequent mutating IDs are assigned each time the consumer desires to obtain content. Further, each customer 26 has a decoding processor or similar device (e.g., a set top box, home computer, etc.) and the customer's address and name are associated with that hardware. Each service provider and content provider also has an actual physical location and address. Because embodiments of the system 20 rely on unraveling multi-encrypted and encapsulated identifiers, the location of a consumer can, at a minimum, be traced to the service area of the service provider of the consumer.

WAVELYNX EX. 1003, Pg. 820
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00217] For example, if the customer or a thief moved the customer's hardware to a location outside of the service area of the service provider 24 and requested content, the unraveling of the multi-encrypted and encapsulated identifiers would fail, as the new service provider would be unable to send appropriate keys needed for decoding based on the mutating ID stored in the hardware.

Real Time Authentication of User and Real Time Play Back of Content

[00218] As noted, one difficulty with modern communications systems is ensuring the authenticity of the communicating parties. In one embodiment of the invention, the system 20 may be used to encode information such as email messages sent from one party to another. The sending party acts like the content provider/service provider and the receiving party like the customer.

[00219] One way to provide extra insurance of authenticity is to send a receiving party certain random information and require that the recipient process that information and relay it back to the sender before communication of valuable information begins. For example, randomly selected sections of text from the U.S. Constitution, the Gettysburg Address, or for that matter any text from the thousands of manuscripts from the Library of Congress may be sent to the receiving party. Before any valuable content or information is sent to the receiving party, the randomly selected text must be properly decrypted and then a duplicate of the information sent back to the recipient. If the receiving party is unable to do this, then an improper communication connection has been made or the receiving party is an imposter, who may, for example, have intercepted or hacked into the communication link of the true recipient. However, without possessing the proper mutating ID, no decryption of the random text is possible.

[00220] An additional security feature that may be added to the system 20 is real-time play back of content. As noted earlier, one of the problems for copyright and other legal rights holders, is that digital content can be copied (at least theoretically) an infinite number of times and the time needed to make each copy is very short. For example, a CD containing seventy minutes of music can be completely copied in a few minutes. Compressed files may be copied even faster. This makes large-scale illicit copying of content very attractive to a potential criminal. Once a single copy of valuable content is obtained, hundreds and possibly thousands of pristine copies can be made quickly and then sold.

WAVELYNX EX. 1003, Pg. 821
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00221] In embodiments of the system 20, decoding of the content at the consumer 26 or receiver is done in a real-time fashion, meaning that play back in the system 20 occurs no faster than playback of the content as intended for the final viewer or consumer. Thus, if a movie has a running time of 2 hours and twenty minutes, recording of the content sent to the consumer takes the same amount of time, thereby discouraging large scale copying of the same. It should also be noted that the system is configured to restrict the number of decryptions that occur. Generally, only one decryption can be made by the customer. This helps reduce unauthorized copying. It should also be noted that the content may include known copy protection devices or code before it is encrypted. These devices may be used to prevent or reduce unauthorized copying as well.

Data and Telemetry Collection

[00222] As noted, embodiments of the invention may be implemented where just three parties participate. The parties include an authenticator, a sending party (by analogy encompassing the roles and functions of the content provider and service provider), and receiving party (by analogy encompassing the roles and functions of the consumer). As noted, the system 20 may be configured to implement a secure email system. As should be apparent, the system 20 could also be implemented in a variety of other applications where secure communication could be useful, such as collecting data from electric and gas meters, equipment and human monitoring systems, and other data and telemetry collection applications. In general, many existing systems could be readily modified with existing processing and communications hardware to allow communications using the multi-encrypted and encapsulated identifier architecture described herein.

Electronic commerce

[00223] Mutating identifiers can also be used in electronic commerce protocols. In some embodiments, the roles of the four participants (i.e., the content provider, the service provider, the consumer, and the authenticator) can be renamed to include a vender, a buyer, a payment authenticator, and an authenticator. Mutating identifiers can be issued and managed by the authenticator to allow the vendor, the buyer, and the payment authenticator to complete a transaction.

WAVELYNX EX. 1003, Pg. 822
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00224] Fig. 18 illustrates an exemplary system 200 configured to perform electronic commerce. In reality, one or more networks or communication systems such as the Internet, the telephone system, wireless networks, satellite networks, cable TV networks, and various other private and public networks could be used in various combinations to provide the communication links desired or needed to create embodiments or implementations of the invention, as would be apparent to one of ordinary skill in the art. Thus, the invention is not limited to any specific network or combinations of networks. However, it is preferred that the networks or communication system used have the ability to support secure communications such as communications with data encrypted with a version of Rijndael encryption, secured socket layer ("SSL") communications, or others. Furthermore, data can be transferred from one party to another with wire, digital satellite service ("DSS"), or physical media being physically carried from one party to another.

[00225] In the embodiment shown in Fig. 18, the system 200 includes four participants: a vendor 220; a payment authenticator 240, such as a credit card company, a financial institution, or the like; a buyer 260; and an authenticator 280. Although, only one vendor 220, payment authenticator 240, and buyer 260 are shown, in most implementations numerous vendors, payment authenticators, and buyers will be involved. Further, there could be multiple authenticators 280, although only one is required. In practice, it is likely that the following relationship will exist: number of authenticators < number of payment authenticators < number of vendors < number of buyers, but again there is no limit on the number of participants or any requirement of a particular relationship between the numbers of the various types of participants.

[00226] In some embodiments, the vendor 220, the payment authenticator 240, and the buyer 260 are connected to the authenticator 280 via two-way links 300, 320, and 340. The vendor 220 and the buyer 260 are also connected via a two-way link 360. These links may be constructed from all or part of the networks mentioned above. In some embodiments, the link 360 includes a non-secure hypertext transport protocol ("HTTP") link.

[00227] The vendor 220 is an entity, such as a retail company, that wishes to sell its goods and/or services electronically. It is assumed that the vendor 220 wants to be reimbursed fairly for goods and/or services, both referred to as goods hereafter, exchanged using the system 20. Thus, in one embodiment of the invention, the system 200 is configured such that

WAVELYNX EX. 1003, Pg. 823
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

the vendor 220 can produce a bill of sale for goods sold. The bill of sale can include a transaction identifier. In some embodiments, the transaction identifier includes a vendor identifier.

[00228] Buyers and vendors agree on a bill of sale and a price. The buyer 260 can authorize the financing of a transaction for items listed in the bill of sale at the agreed upon price from an account. Buyers, vendors, and payment authenticators can receive an unforgeable receipt of the transaction as described above. It is assumed that at least some buyers may wish or attempt to purchase goods electronically without paying for them or with funds from an account that the buyer is not authorized to manage. It is also assumed that the buyer requires a secure transaction where payment information cannot be compromised. Therefore, measures are provided to prevent unauthorized purchasing of goods and to provide a secure transaction. Mutating IDs provide a mechanism by which purchasing is controlled.

[00229] The payment authenticator 240 is an entity, such as a credit card company or financial institution, which holds accounts that can be used to finance transactions (in terms of money or other payment forms or mechanisms). It is assumed that the payment authenticator 240 can agree to finance an electronic transaction from an account and, therefore, account identifiers are kept confidential. Thus, in some embodiments of the invention, the system 200 is configured such that the buyer 260 and the payment authenticator 240 agree on a secret account identifier for an account of the buyer 260. Further, authorizations for payment of a transaction from an account are encrypted with a mutating ID.

[00230] The authenticator 280 is the repository that holds the data necessary to perform secure electronic transactions. In the embodiments discussed, the authenticator 280 verifies the vendor 220, the payment authenticator 240, and the buyer 260 by their mutating IDs before allowing an e-commerce transaction to take place. The authenticator 280 can verify the receipts of the buyer, the vendor, and the payment authenticator. The authenticator 280 can also perform the above actions without knowing the buyer's account information or the details of the transactions. The authenticator 280 is also the source of mutating IDs and keeps track of such IDs using a database or similar mechanism.

[00231] Exemplary embodiments of the invention will now be described using several examples.

WAVELYNX EX. 1003, Pg. 824
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00232] As with many descriptions of communication protocols, names are assigned to the various entities (or computer systems associated with those entities) used in the protocol. In one embodiment, Bob (*B*), Vera (*V*), and Carol (*C*) represent various participants in a protocol, and Trent (*T*) represents a trusted arbiter of communication. The following table, Table 2, is a list of other symbols used in this document to explain multiple embodiments of the protocol.

Table 2

Symbol	Meaning
B, V, C, T	Entities using the protocol.
S	A bill of sale.
P	Agreed upon price for goods.
R_X	Receipt for X.
M_X	A message for X made by Trent sent via a third part.
x_X, y_X	An account number for X.
X_{id}	Some (not secret) digital identifier of X (*e.g.*, an email address, name, or the like).
X_{cred}	Secret information that identifies a party X, known only by X and T (and C if applicable), and can be randomly chosen by T. Providing knowledge of this data can authenticate X, C, and/or T with each other.
K_X	A key for a symmetric cipher associated with some entity X.
N_X	A one use number (nonce) associated with some key K_X.
$H(X)$	A function that produces a secure hash of X.
$E(K, X)$	A cipher that encrypts X with K.
$X \rightarrow Y : Z$	Message Z sent from X to Y.

[00233] An exemplary embodiment of the protocol involves the four participants discussed above. The entity Bob ("B") performs the role of the buyer 260, the entity Vera ("*V*") performs the role of the vendor 220, the entity Carol ("*C*") performs the role of the payment authenticator 240, and the entity Trent ("*T*") performs the role of the authenticator 280. The protocol involves Bob purchasing goods from Vera. Bob purchases or pays for the goods using an account held by Carol. Trent arbitrates communication between Bob, Vera, and Carol. Since the proposed protocol relies on a trusted authority, Bob, Vera, and Carol trust Trent. Further, all assigned number and keys are assigned and known by Trent. It is assumed

WAVELYNX EX. 1003, Pg. 825
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

that Bob, Vera, and Carol already hold secret number/key pairs (N_B,K_B), (N_V,K_V), and (N_C,K_C), respectively, issued from Trent.

[00234] For the purposes of this example only, assume Bob wishes to purchase goods from Vera. Bob and Vera agree on a bill of sale (S). Bob wishes to pay with funds drawn from an account held with Carol. The account is identified by credentials (B_{cred}). The credentials (B_{cred}) are a secret known or recognizable only to Bob, Carol, and Trent. In some embodiments, the credentials (B_{cred}) represent Bob's account number. In other embodiments, the credentials (B_{cred}) are assigned by Trent. Trent does not have to "know" credentials a priori or before hand for the protocol to work. In some embodiments, Trent only forwards credentials to Carol. Furthermore, in some embodiments, Trent cannot obtain the data (such as an account number) included in credentials. This helps increase security of the protocol.

[00235] Since the credentials (B_{cred}) are known only to Bob, Trent, and Carol, Trent and Carol can use the credentials (B_{cred}) to verify that Bob created a particular message. Carol may also use the credentials (B_{cred}) to verify Bob's account number. In some embodiments, the credentials (B_{cred}) are constructed from a secret known only to Bob and Carol (such as Bob's account number). The credentials (B_{cred}) can also be constructed from details regarding the current transaction. In some embodiments, the credentials (B_{cred}) are determined as follows:

$$B_{cred} = E(H(x), H(S)P)$$

[00236] In the above equation, x is a secret known only to Bob and Carol (such as Bob's account number), S is the bill of sale, and P is the agreed upon price for the goods included in the bill of sale. In some embodiments, Bob constructs his credentials (B_{cred}) from plaintext versions of the bill of sale and/or the price rather than as a hash. Using a hash, however, provides an abstraction of the details of the transaction. It should be understood that additional formulas or mechanisms can be used to determine credentials.

[00237] Since Bob and Carol know x (and the hash function if applicable), Bob and Carol can decrypt the credentials (B_{cred}) and obtain the secure information regarding Bob's account. In some embodiments, Trent and Vera cannot obtain the secure information regarding Bob's account or the details of the transaction, such as the price.

WAVELYNX EX. 1003, Pg. 826
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00238] Bob can generate credentials (B_{cred}) for each transaction, and Carol (who knows Bob's account number and can generate $H(x)$) can decrypt the credentials (B_{cred}) to obtain the bill of sale and the corresponding price. In some embodiments, if Carol holds multiple accounts for Bob each having account numbers $x_1, x_2, \ldots, x_n$, Carol generates a hash for each account number. If one of the hashes can decrypt the credentials (B_{cred}), Carol knows which account to draw funds from. Bob can also prepend an account identifier to the credentials (B_{cred}) to identify a particular account.

[00239] In some embodiments, creating a hash of an account can create hash collisions where $H(x_i) = H(x_j)$ and x_i does not equal x_j. Hash collisions can be detected at account creation and a colliding account number can be regenerated in order to prevent a hash collision.

[00240] As shown in Fig. 19, to begin the purchase process, Vera sends Bob signed vendor transaction data. In some embodiments, the vendor transaction data includes a bill of sale (S) and/or a corresponding total price (P) for the bill of sale (S). In addition to or in place of the plaintext bill of sale (S) and corresponding price (P), the vendor transaction data can include a hash of the bill of sale (S) and/or the price (P). In some embodiments, the vendor transaction data also includes credentials of Vera (V_{cred}). Vera's credentials (V_{cred}) are a secret known or recognizable only to Vera, Carol, and Trent. In some embodiments, as described above, Vera's credentials (V_{cred}) can be constructed from a secret known only to Vera and Carol, such as an account number of Vera. In other embodiments, Trent can assign credentials (V_{cred}) to Vera. Carol and Trent can use Vera's credentials (V_{cred}) to verify that the vendor transaction data was generated by Vera. The vendor transaction data can also include an identifier of a buyer (e.g., Bob) and/or a payment authenticator (e.g., Carol) associated with the transaction. Vera "signs" the vendor transaction data by encrypting the data with her secret key (K_V) and prepending her secret number (N_V). Vera sends the signed vendor transaction data to Bob.

$$V \rightarrow T: S\, N_V\, E(K_V, H(S)P)$$

[00241] Upon receiving the signed data from Vera, Bob provides signed buyer transaction data as Vera did. The buyer transaction data includes a bill of sale, which, when Bob acts correctly and honestly, is identical or equal to the bill of sale signed by Vera. Bob can also

WAVELYNX EX. 1003, Pg. 827
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

include his credentials (B_{cred}) and identities of the participants of the transaction beside himself (i.e., Vera and Carol) in the buyer transaction data. Bob signs the buyer transaction data by encrypting the data with his secret key (K_B) and prepending his secret number (N_B). Bob concatenates the signed buyer transaction data to Vera's signed vendor transaction data and sends the concatenated message to Trent.

$$B \rightarrow T: N_B\, E(K_B,H(S)P)\, B_{cred} V_{id} C_{id} N_V\, E(K_V,H(S)P)$$

[00242] It should be understood that Bob can also initiate the purchase process. In some embodiments, Bob sends Vera signed buyer transaction data including the identities of Vera and Carol. Vera adds signed vendor transaction data to the signed data provided from Bob and forwards the concatenated message to Trent.

[00243] Trent can unroll the concatenated message (since he knows the secret keys of Bob and Vera). In some embodiments, Trent verifies that the buyer transaction data, or a portion thereof, (e.g., the bill of sale, the price, and/or the hashes of the bill of sale and/or price) transmitted from Bob matches the vendor transaction data, or a portion thereof, transmitted from Vera. If the data does not match, it is possible that Vera and Bob have not agreed on a common bill of sale and/or a related price, and Trent can inform Bob and Vera of the discrepancy.

[00244] If the data matches, Trent generates a payment request. In some embodiments, the payment request is transmitted to Carol in order to request payment of a transaction between Bob and Vera. The payment request can include receipts for Vera, Bob, and Carol. Each receipt can include keys of two of the three participants (i.e., the keys of the participants to whom the receipt is not for), the bill of sale, and the price. In some embodiments, each receipt also includes credentials of the recipient and/or the other participants. The recipient can use the credentials to verify that the receipt was generated by Trent. It should be understood that a hash of the keys, the bill of sale, the price, and/or the credentials can be included in place of plaintext data in order to further increase the security and secrecy of the transaction. When hashes are provided, Trent can obtain the hashes but cannot decipher the details of the transaction. Exemplary receipts for Vera, Bob, and Carol may be configured as follows:

$$R_V = E(K_V,H(K_B K_C P)H(S)P)$$

WAVELYNX EX. 1003, Pg. 828
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

$$R_B = E(K_B, H(K_V K_C P) H(S) P)$$

$$R_C = E(K_C, H(K_B K_V P) H(S) P)$$

[00245] The payment request can also include new number/key pairs for Vera, Bob, and Carol.

$$M_V = E(K_V, N'_V K'_V)$$

$$M_B = E(K_B, N'_B K'_B)$$

$$M_C = E(K_C, N'_C K'_C)$$

[00246] The payment request can further include messages for Carol to send to Bob and Vera. In some embodiments, Trent generates one "accepted" message and one "declined" message. Carol sends the "approved" message, or a portion thereof, to Bob and Vera if she honors the request for payment of the transaction, and Carol sends the "declined" message, or a portion thereof, to Bob and Vera if she does not honor the request for payment.

$$M_{approved} = E(K_V, \text{"Approved"}) E(K_B, \text{"Approved"})$$

$$M_{declined} = E(K_V, \text{"Declined"}) E(K_V, \text{"Declined"})$$

[00247] It should be understood that the payment request Trent generates can include more or fewer messages. For example, Trent can generate a message that includes both the receipt and the new number/key pair for each participant. Trent can also generate separate "approved" and "declined" messages from Bob and Vera.

[00248] Trent also decodes Bob's credentials (B_{cred}) and the price of bill of sale (P). In some embodiments, Trent cannot decode Bob's credentials (B_{cred}) and, therefore, cannot obtain confidential information regarding Bob's account held with Carol. For that reason, although Trent obtains Bob's credentials (B_{cred}), Trent cannot obtain Bob's account information.

[00249] The payment request can also include a credential message that includes Bob's credentials (B_{cred}) and the price (P). Trent can re-encrypt the credential message with Carol's secret key (K_C) in order to prevent anyone but Carol from obtaining the data contained in the

WAVELYNX EX. 1003, Pg. 829
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

credential message. Trent can also prepend Carol's secret number (N_C) to the credential message.

$$M_{cred} = E(K_C, B_{cred} P)$$

[00250] The payment request can also include the identities of the transaction participants (other than Carol).

[00251] Trent sends the payment request to Carol. In some embodiments, Trent can also send Carol the messages and receipts included in the payment request individually. In some embodiments, Trent encrypts the payment request (or the individual messages and/or receipts) with Carol's secret key (K_C). Trent can also prepend Carol's secret number (N_C) in order to instruct Carol on how to decode the payment request.

$$T \rightarrow C: N_C\, E(K_C, B_{id} V_{id} (R_C R_B R_V)(M_C M_B M_V)(M_{accept} M_{decline})(M_{cred}))$$

[00252] Carol receives the payment request and determines whether to approve payment for the bill of sale. In some embodiments, Carol determines whether or not to approve payment by determining if Bob's account (identified by B_{cred}) contains enough funds to cover the price (P) of the bill of sale. If Bob's account contains enough funds to cover the price, Carol transfers funds from Bob's account to an account of Vera. In some embodiments, Carol can act as an escrow and can hold funds from Bob's account until Vera notifies Carol that goods included in the bill of sale have been shipped and/or provided to Bob. Once the goods have been provided to Bob, Carol can transfer the funds from Bob's account to an account of Vera. Upon approving payment, Carol can send responses to both Bob and Vera including a receipt, a new number/key pair, and an approval message.

$$C \rightarrow B: E(K_B, \text{"Approved"})\, M_B R_B$$

$$C \rightarrow V: E(K_V, \text{"Approved"})\, M_V R_V$$

[00253] Bob, Vera, add Carol can present Trent with the number they used in this transaction (N_B, N_V, or N_C), the price, and their receipt for transaction verification. For example, Trent can verify that the receipts are identical.

WAVELYNX EX. 1003, Pg. 830
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00254] If Bob's account does not contain enough funds to cover the price, Carol does not transfer funds from Bob's account to Vera's account. Carol does, however, send responses to both Bob and Vera that includes a new number/key pair and a declined message. Carol can also send Bob and Vera a receipt indicating the declined transaction.

$$C \rightarrow B\text{: } E(K_B, \text{"Declined"})\ M_B R_B$$

$$C \rightarrow V\text{: } E(K_V, \text{"Declined"})\ M_V R_V$$

[00255] Fig. 19 schematically illustrates another exemplary communication protocol. The exemplary protocol involves Bob purchasing goods listed on a bill of sale (S) from Vera using an account held by Carol. Again, Trent arbitrates communication between Bob, Vera, and Carol. And, since the proposed protocol relies on a trusted authority, Bob, Vera, and Carol trust Trent. Further, all assigned number and keys are assigned and known by Trent.

[00256] For the purposes of this example only, assume Bob wishes to purchase goods from Vera. In contrast to the previous example, in order for Vera and Bob to exchange goods, Trent first establishes secure communication between Bob and Vera.

[00257] Bob sends Trent a request for a transaction key. The request can include the identity of a vendor that Bob wishes to communicate with. In some embodiments, Bob encrypts his request with his secret key (K_B) and prepends his secret number (N_B).

[00258] To further ensure security, Bob can include additional request-identifying data X in the request. In some embodiments, the data X includes random or psuedorandom data. For example, the data X can include secret credentials of Bob known only to Bob and Trent. The data X can be used to authenticate Trent. Since the request is encrypted with Bob's secret key, only Bob and Trent can decode the message. Trent can prove he is the authenticator by decoding Bob's request and including the data X in his response to Bob. Therefore, Trent proves that he has decoded Bob's request.

[00259] The data X can also be used to associate a response from Trent with a specific request. Trent can also pass the data X to the vendor Bob identifies in his request.

$$B \rightarrow T\text{: } N_B\, E(K_B, V_{id} X)$$

WAVELYNX EX. 1003, Pg. 831
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00260] In some embodiments, Bob can send the request for a transaction key to the vendor that Trent wishes to communicate with (e.g., Vera), and the vendor can concatenate its own request for a transaction key. The request generated by the vendor can include the identity of the buyer that sent the original request for a transaction key to the vendor. The request generated by the vendor can also include request-identifying data Y, which can include random or psuedorandom data. For example, the data Y can include secret credentials of the vendor known only to the vendor and Trent. The vendor can encrypt its request for a transaction key with its secret key (K_V) and can prepend its secret number (N_V). The vendor can then send the concatenated requests for a transaction key to Trent. Trent can use the concatenated request to verify that each party agrees to establishing a transaction key.

[00261] Trent decrypts the request for a transaction key from Bob and generates a secret transaction key K_{BV} that Bob and Vera can use to communicate and negotiate a transaction. Trent can also generate new number/key pairs for Bob and Vera. Trent encrypts the key and the new number/key pairs with Bob's key and Vera's key, respectively, to create two messages. The messages can include the secret data (e.g., credentials) provided by the recipient of the message in the recipient's request for a transaction key. As described above, a recipient can use the secret data to verify that Trent generated the message. Trent sends one message to Bob and one message to Vera. Trent can also send the concatenation of the messages to either Vera or Bob. In some embodiments, a first part of the concatenated message ($N_V\, E(K_V, B_{id} K_{BV} X N'_V K'_V)$) can includes Vera's information, and a second part of the concatenated message ($E(K_B,\, K_{BV} N'_B K'_B)$) includes Bob's information. Whoever receives the concatenated messages (i.e., either Vera or Bob) can take their encrypted message from the concatenated message and can pass the remaining concatenated message to the other participant.

$$T \rightarrow V:\ N_V\, E(K_V, B_{id} K_{BV} X N'_V K'_V) E(K_B,\, K_{BV} N'_B K'_B)$$

[00262] For example, Vera can receive the message and can decrypt the first part of the message. Vera, however, cannot decrypt the second part of the message since it is encrypted with Bob's secret key K_B. Upon decrypting the first part of the message, Vera can recover the secret transaction key K_{BV} generated by Trent for Vera and Bob to communicate. Vera forwards the second part of the message to Bob. As noted above, in some embodiments, if

WAVELYNX EX. 1003, Pg. 832
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Trent includes the data X in the first part of the message, Vera can also forward Bob the data X encrypted with the secret key K_{BV}.

$$V \rightarrow B: E(K_B, K_{BV}N'_BK'_B)E(K_{BV},X)$$

[00263] After Bob and Vera receive their respective messages from Trent and/or each other, Bob and Vera share a secret key (K_{BV}) that they can use to negotiate a transaction. As shown in Fig. 19, Bob and Vera can use the secret key (K_{BV}) to negotiate a transaction and exchange buyer information and vendor information. The buyer information can include shipping information provided by the buyer. Once Bob and Vera agree upon the transaction, Bob and Vera are ready to negotiate payment.

[00264] It is assumed that all transactions associated with an account require the buyer to identify him or herself to the exclusion of others. In traditional commerce, this involves using a card with a particular account number and a signature. In e-commerce, the account number and other data (like a billing address zip code) are used to identify the buyer.

[00265] In some embodiments, the buyer (Bob) is identified with the transaction simultaneously. Instead of using a static account number to identify the buyer, elements of the buyer and the transaction are combined to generate buyer credentials. In some embodiments, elements of the buyer known only to the buyer and a payment authenticator holding the account of the buyer (e.g., x_B and y_B), elements of the vendor (V_{id}), the bill of sale (S), and the agreed upon price (P) are combined to generate credentials of the buyer (B_{cred}). The buyer can compute the credentials at the time of the purchase.

[00266] Let x_B and y_B be parts of an account number of Bob. At the point of sale, Bob can generate his credentials as follows:

$$B_{cred} = E(x_B,y_BV_{id}H(SP)P)$$

[00267] Only individuals with the knowledge of x_B can decode this message. In some embodiments, since only Bob and Carol know x_B (and y_B), this message can be safely passed to Trent since he cannot decode the message and, therefore, he cannot obtain the account information.

WAVELYNX EX. 1003, Pg. 833
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00268] In some embodiments, to ensure that Bob does not cheat on the price, Vera generates similar credentials. Carol can use Vera's credentials to verify that Bob and Vera agree on the bill of sale (S) and the price (P).

$$V_{cred} = E(x_V, y_V B_{id} H(SP) P)$$

[00269] It should be understood that Bob and Vera can construct their credentials using other mechanisms, such as the mechanism described in the previous embodiment of the e-commerce protocol.

[00270] To begin the purchase protocol, Vera sends Bob signed vendor transaction data. The vendor transaction data can include a hash of the bill of sale and the price ($H(SP)$) and Vera's credentials (V_{cred}). By including the price in the hash rather than as plaintext, Trent is unaware of the bill or sale and the corresponding price. Vera encrypts the data with her secret key (K'_V) and prepends the encrypted data with her secret number (N'_V).

$$V \rightarrow B: N'_V E(K'_V, H(SP) V_{cred})$$

[00271] Bob concatenates this message with signed buyer transaction data. The buyer transaction data can include Bob's credentials (B_{cred}) and another hash of the bill of sale and the price ($H(SP)$). The buyer transaction data can also include Carol's identity (C_{id}). To provide his signature, Bob's encrypts the buyer transaction data with his secret key (K'_B) and prepends his secret number (N'_B). Bob sends the concatenated vendor and buyer transaction data to Trent.

$$B \rightarrow T: N'_B E(K'_B, C_{id} H(SP) B_{cred} N'_V E(K'_V, H(SP) V_{cred}))$$

[00272] It should be understood that Bob can also initiate the purchase process. In some embodiments, Bob's sends signed buyer transaction data to Vera. Vera can concatenate signed vendor transaction data to the received data and can forward the concatenated data to Trent.

[00273] In some embodiments, as part of restrictions on Trent to ensure security, Trent cannot know the details of the transaction. However, he can identify the transaction data in order to verify receipts. Trent can use the transaction data to determine if Bob and Vera have agreed on a price and a bill of sale for a transaction. By creating a hash of the bill of sale and

WAVELYNX EX. 1003, Pg. 834
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

price, Trent receives the transaction data but cannot determine the details of the bill of sale and/or the related price.

[00274] Upon receipt of the message including the vendor and buyer transaction data, Trent decodes the data and recovers the buyer transaction data and the vendor transaction data. In some embodiments, Trent verifies that Bob and Vera have agreed on a bill of sale and price. If the bill of sale and/or price were provided as a hash, Trent determines whether the hash provided by Vera matches the hash provided by Bob.

[00275] In some embodiments, Trent cannot decode Bob's credentials (B_{cred}) because Trent does not know Bob's account number (x_B *and* y_B). Similarly, Trent cannot derive Vera's credentials (V_{cred}).

[00276] After decoding the transaction data, Trent generates a payment request. Trent can transmit the payment request to Carol in order to request payment for the transaction between Bob and Vera. The payment request can include receipts for Bob, Vera, and Carol. Each receipt can provide a signed version of transaction data. In some embodiments, the receipt includes a hash of the bill of sale and the price. Each participant receipt is encrypted by the two number/key pairs of the other two participants. For example, Carol's receipt for the transaction can be as follows:

$$R_C = N'_B E(K'_B, N'_V E(K'_V, B_{id} V_{id} H(SP)))$$

[00277] Since the receipt requires the secret keys of both Bob and Vera, only Trent can construct this value. In this sense, the receipt is perceivably unforgeable.

[00278] Similarly, Trent can create receipts for Bob and Vera.

$$R_B = N_C E(K_C, N'_V E(K'_V, C_{id} V_{id} H(SP)))$$

$$R_V = N'_B E(K'_B, N_C E(K_C, B_{id} C_{id} H(SP)))$$

[00279] . The payment request can also include the identities and credentials of Bob and Vera, transaction data, Carol's receipt, a new number/key pair for Carol, and/or a hash of the bill of sale and/or the price. Trent can encrypt the payment request with Carol's current secret key (K_C) in order to prevent anyone but Carol from obtaining the data contained in the request. Trent can also prepend Carol's secret number (N_C) to the request. In some

WAVELYNX EX. 1003, Pg. 835
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

embodiments, Carol has secret number/key pairs issued by multiple authenticators, and Trent can prepend Carol's secret number in order to identify the secret key Carol should use to decode the request.

$$M_C = N_C E(K_C, B_{id} B_{cred} V_{id} V_{cred} H(SP) R_C N'_C K'_C)$$

[00280] In some embodiments, the payment request further includes messages for Carol to send to Bob and Vera. In some embodiments, Trent generates two messages for Bob and two messages for Vera. Each pair of messages includes an "accepted" message and a "declined" message. Carol sends the "accepted" messages to Bob and Vera if she honors the request for payment of the transaction. Carol sends the "declined" messages to Bob and Vera if she does not honor the request for payment. The "accepted" messages include the receipts for Bob and Vera so that if the request for payment is honored, Bob and Vera receive receipts of the transaction. Both the "accepted" and the "declined" messages can also include new number/key pairs for both Bob and Vera. Separate messages can also be generated and sent, including the new number/key pairs and/or the receipts, such as those below.

$$M_B = E(K'_B, \text{"Approved"} R_B N''_B K''_B)$$

$$M'_B = E(K'_B, \text{"Declined"} R_B N''_B K''_B)$$

$$M_V = E(K'_V, \text{"Approved"} R_V N''_V K''_V)$$

$$M'_V = E(K'_V, \text{"Declined"} R_V N''_V K''_V)$$

[00281] Trent sends the payment request to Carol.

$$T \rightarrow C: M_C M_B M'_B M_V M'_V$$

[00282] Once Carol receives the request, she can retrieve Bob's and Vera's credentials (B_{cred} and V_{cred}). Carol can decrypt both of the credentials since she knows the account numbers of Bob and Vera. Carol can also verify that both Bob and Vera agree on the bill of sale and the price. Further, Carol can determine whether to honor the request for payment. If Carol honors the request for payment, she sends responses to Bob and Vera that each include an "approved" message.

$$C \rightarrow B: M_B$$

WAVELYNX EX. 1003, Pg. 836
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

$C \rightarrow V: M_V$

[00283] Conversely, if Carol does not honor the request for payment, Carol sends responses to Bob and Vera that each include a "declined" message.

$C \rightarrow B: M'_B$

$C \rightarrow V: M'_V$

[00284] In Carol honors the request for payment, Carol also transfers funds between Bob's account and an account of Vera as indicated by the price P. In some embodiments, Carol can act as an escrow and can hold funds from Bob's account until Vera notifies Carol that goods included in the bill of sale have been shipped and/or provided to Bob.

[00285] The above protocol describes a secure e-commerce protocol involving a small number of communications and connections. However, in some situations application of the protocol is inappropriate or inefficient. For example, in the above protocol, Vera sends her account information through both Bob and Trent before it reaches Carol. While this is perfectly safe, it may be preferred to avoid distributing account information even in a secure, encrypted form.

[00286] The above protocol can be modified so that both Bob and Vera communicate sensitive account information directly to Carol. Fig. 20 illustrates another embodiment of the system 200 that includes the buyer 260, the vendor 220, the payment authenticator 240, and the authenticator 280. In comparison to Fig. 18, the system 200 shown in Fig. 20 connects each participant to every other participant of the system 200. The vendor 220, the payment authenticator 240, and the buyer 260 are each connected to the authenticator 280 via the two-way links 300, 380, and 360. The buyer 260 is also connected to the vender 220 and the payment authenticator 240 via two-way links 360 and 380. Furthermore, the vendor 220 is connected to the payment authenticator 240 via a two-way link 400. These links may be a constructed from all or part of the networks mentioned above.

[00287] Fig. 21 illustrates a communication protocol used in the system shown in Fig. 20. After Bob and Vera agree on a bill of sale and a method of payment, they each communicate directly with Carol rather than indirectly through Trent. In some embodiments, this can be accomplished by establishing secret transaction keys between Carol and Bob and between

WAVELYNX EX. 1003, Pg. 837
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Carol and Vera. For example, Bob sends Trent a request for a transaction key (K_{BC}) that allows him to directly communicate with Carol. Vera can also request a transaction key (K_{CV}) from Trent so that she can also directly communicate with Carol. After the transaction keys (K_{BC} and K_{CV}) are established, the credentials of Vera and Bob can be directly sent to Carol.

[00288] Since the credentials can be provided directly to Carol, the above protocols can be modified so that account data is not sent indirectly through Trent. Once Vera and Bob agree on the bill of sale (S) and the price (P), Vera signs the vendor transaction data and sends the signed data to Bob. In some embodiments, the vendor transaction data includes the hash of the bill of sale and the price ($H(SP)$).

$$V \rightarrow B: N'_V E(K'_V, H(SP))$$

[00289] In contrast to the above protocol, rather than including her credentials in the vendor transaction data, Vera sends her credentials directly to Carol encrypted with the secret key (K_{CV}) issued by Trent.

$$V \rightarrow C: E(K_{CV}, V_{cred})$$

[00290] Upon receiving the signed vendor transaction data from Vera, Bob concatenates the signed vendor data received from Vera with buyer transaction data signed with his secret key/number pair. In some embodiments, the buyer transaction data includes a hash of the bill of sale and the price. The buyer transaction data can also include Carol's identity. Bob sends the concatenated buyer and vendor transaction data to Trent.

$$B \rightarrow T: N'_B E(K'_B, C_{id} H(SP) N'_V E(K'_V, H(SP)))$$

[00291] Also, since Bob can securely and directly communicate with Carol, Bob sends Carol his credentials (which can include the details of the transaction). Bob encrypts his credentials with the secret key he shares with Carol (K_{BC}).

$$B \rightarrow C: E(K_{BC}, B_{cred})$$

WAVELYNX EX. 1003, Pg. 838
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00292] Trent receives the concatenated transaction data and creates a payment request for Carol. The payment request can include the same information as described in the above protocol except for the credentials of Bob and the credentials of Vera.

[00293] After receiving the payment request from Trent and the credentials from Bob and Vera, Carol has the same information as she received in the previous protocol, but has received the information through different channels or connections. Even if someone illegally obtains and resends one of the messages provided to Carol, Carol must receive and verify messages from each participant before she approves payment. Furthermore, the payment request from Trent includes a mutating identifier so that Carol will be unable to decode an illegally re-sent message from Trent since the mutating identifier has mutated or changed since its previous use.

[00294] Carol now decides whether to honor the request for payment for the transaction. If Carol honors the request for payment, Carol sends responses to Bob and Vera that include an "approved" message. If Carol does not honor the request for payment, Carol sends responses to Bob and Vera that include a "declined" message. In some embodiments, Carol encrypts the responses to Bob and Vera with the secret keys issued by Trent to further secure the transaction. For example, if Carol honors the request for payment, she sends the following responses:

$$C \rightarrow B: E(K_{BC}, M_B)$$

$$C \rightarrow V: E(K_{CV}, M_V)$$

[00295] If Carols honors the request, Carol also transfers funds from Bob's account to an account of Vera. In some embodiments, however, Carol acts as an escrow and waits until she receives notification from Vera and/or Bob that the goods included in the transaction are shipped and/or provided to Bob before transferring funds from Bob's account to Vera's account.

[00296] On the other hand, if Carol does not honor the request for payment, she sends the following encrypted responses:

$$C \rightarrow B: E(K_{BC}, M'_B)$$

WAVELYNX EX. 1003, Pg. 839
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

$C \rightarrow V$: $E(K_{CV}, M'_V)$

[00297] As should be understood, the above protocols can be used to perform secure commerce, both electronic commerce and non-electronic commerce. Furthermore, it should be understood that the roles of authenticator and payment authenticator can also be combined. For example, each payment authenticator can provide their own mutating identifiers to their clients (individuals for whom they hold accounts for).

[00298] It should also be understood that the above protocols (or portions thereof) can be combined. For example, electronic commerce transactions can be included in digital content purchases from a content provider or a service provider. Additionally, electronic commerce transactions can be watermarked to guarantee uniqueness in transaction data and corresponding receipts. Additional combinations and configurations are also possible.

[00299] To ensure the best possible security of transmitted data, the secret keys of mutating IDs (e.g., K_B, K_C, and K_V) need to remain secret. For example, if Trent provides Bob with a new mutating ID encrypted with Bob's current secret key (e.g., K_B), an eavesdropper who has determined Bob's current secret key can obtain Bob's new mutating ID provided by Trent. The eavesdropper can then use new mutating ID to send false data and/or obtain the plaintext of future data exchanged between Bob and Trent.

[00300] Theoretically, it is possible (although as noted in earlier parts of this document, highly improbable) that one or more eavesdroppers could determine (or attempt to determine) a key used to encrypt particular data by performing a brute force attack. As shown in Fig. 22, a brute force attack includes decrypting ciphertext with every possible key until a key is found that produces coherent or recognizable data (e.g., human readable data). As shown in Fig. 22, an eavesdropper determines an initial or zero candidate key (step 400). The eavesdropper then uses the candidate key to decrypt ciphertext (step 402). After decrypting the ciphertext, the eavesdropper can inspect the result (i.e., the candidate plaintext) to determine if the ciphertext decrypted with the candidate key produces coherent plaintext or a coherent pattern (step 404). If the eavesdropper obtains or knows the plaintext (or a portion or pattern thereof) corresponding to obtained ciphertext, the eavesdropper can more easily determine whether a correct candidate key has been found. For example, if the eavesdropper obtains ciphertext and knows that the ciphertext includes an individual's name followed by a

WAVELYNX EX. 1003, Pg. 840
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

4-digit personal identification number ("PIN"), the eavesdropper can apply candidate keys until a candidate key produces the plaintext including the individual's name. The eavesdropper can then assume, with some certainty, that remaining information included in the generated plaintext corresponds to the PIN.

[00301] As shown in Fig. 22, if the eavesdropper finds a coherent pattern in candidate plaintext generated by decrypting ciphertext with a particular candidate key (step 406), the eavesdropper knows, with some certainty, that the current candidate key equals or is the key used to generate the ciphertext (step 407).

[00302] If the eavesdropper does not find a coherent pattern in the candidate plaintext generated by decrypting ciphertext with a particular candidate key (step 406), the eavesdropper can modify the candidate key (e.g., increment the candidate key) (step 408), and can use the modified candidate key to decrypt the ciphertext (step 402) and inspect the generated candidate plaintext for coherent plaintext or a coherent pattern (step 404). Given enough processing power and time, the eavesdropper can continue this process until a particular candidate key generates candidate plaintext with coherent plaintext or a coherent pattern and, therefore, determine the key used to generate the ciphertext.

[00303] However, if the eavesdropper has no knowledge of the plaintext or a pattern of the plaintext (i.e., has no content hint), the eavesdropper's ability to determine whether a correct candidate key has been found is greatly reduced and, perhaps, eliminated. For example, if plaintext includes a random number encrypted with a particular key, no matter how many keys the eavesdropper attempts in a brute force attack, the eavesdropper will have no way to determine whether candidate plaintext is the true plaintext corresponding to the ciphertext. Decrypting an encrypted random number with any candidate key will produce a random number that is equally likely to be the original random number as every other random number produced by every other candidate key.

[00304] Referring to the transaction key exchange example described above involving Bob, Vera, and Trent, if any portion of an encrypted message are recognizable, known, become known, or include any content hints, an eavesdropper could possibly perform a plaintext or partial-plaintext attack on the encrypted message and uncover a secret key of Bob or Vera used to encrypt the message. For example, assume that Bob sends the following message to Trent that is intercepted by an eavesdropper.

WAVELYNX EX. 1003, Pg. 841
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

$B \rightarrow T$: N_B $E(K_B, V_{id} X)$

[00305] The eavesdropper can perform a brute force attack on the intercepted message because Vera's identifier V_{id} and the format of the above message are known or public. Thus, the eavesdropper can obtain Bob's secret key K_B and the data X. Furthermore, once the eavesdropper obtains Bob's current secret key K_B, the eavesdropper can use Bob's current secret key K_B to obtain all data encrypted with Bob's current secret key K_B, such as his next mutating ID (e.g., N_B' and K_B').

[00306] An eavesdropper can use other knowledge about an encrypted message or the communication protocol used to generate an encrypted message to perform brute force attacks. For example, an eavesdropper can use the mutating ID number (e.g., N_B), which is passed in the clear, to perform a brute force attack. An eavesdropper could also use knowledge of the algorithm used to generate the mutating ID numbers to perform a brute force attack.

[00307] As pointed out above, keys used to encrypt undiscoverable data (i.e., data that is random or has no content hints) cannot be determined or discovered using a brute force attack, since an eavesdropper will be unable to determine when a correct candidate key is found. Keys used to encrypt discoverable data (i.e., data that is known, may be later disclosed, is recognizable, or has a known or easily guessed format), however, can (theoretically) be determined using a brute force attack. When the discoverable data and the undiscoverable data are encrypted together or with the same encryption key, the key determined through a brute force attack using the discoverable data is also the key used to encrypt the undiscoverable data and, therefore, the undiscoverable data can be discovered.

[00308] To increase the security of encrypted data and to prevent eavesdroppers from obtaining encryption keys using brute force attacks, embodiments of the invention provide an encryption strategy that protects the security of undiscoverable data, such as the secret keys included in mutating IDs.

[00309] Fig. 23 illustrates types of data that can be included in data that is to be encrypted. Data 420 that is to be encrypted and transmitted to a particular receiver is separated into types or classes of data. A first class of data includes an undiscoverable or secret data class 430. The secret data class 430 can include data that is kept secret and only known by authorized

WAVELYNX EX. 1003, Pg. 842
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

entities. For example, the secret data class 430 can include the secret keys of the mutating IDs and/or the credentials of the entities, which may be both random and known only by an authenticator and/or a payment authenticator and the holder of the credentials and the secret keys.

[00310] A second class of data includes a discoverable data class 440. The discoverable data class 440 can include data that is known, may later be known, is recognizable (e.g., human-readable), or has a known or easily guessed format. In some embodiments, the discoverable data class 440 is further divided into a number of subclasses. For example, the discoverable data class 440 can include a first discoverable data subclass 442 that includes discoverable data of a first type. Discoverable data of a first type can include data that has easily distinguished characteristics, such as standardized headers, known patterns, or other publicly available formats. In some embodiments, the first discoverable data subclass 442 includes content transmitted between entities and the numbers (e.g., N_B, N_V, and N_C) of the mutating IDs provided by an authenticator.

[00311] The discoverable data class 440 can also include a second discoverable data subclass 444 that includes discoverable data of a second type. Discoverable data of a second type can include keys used to encrypt messages that include discoverable data of the first type. In some embodiments, the second discoverable data subclass 444 includes the secret keys (e.g., K_B, K_V, and K_C) of the mutating IDs. For example, as described above with respect to the transaction key exchange protocol, Bob sends Trent a message that includes Vera's publicly-known identifier V_{id} encrypted with his secret key K_B.

$$B \rightarrow T: N_B E(K_B, V_{id} X)$$

[00312] Since Vera's identifier V_{id} is publicly known and, therefore, is discoverable data of the first type, Bob's secret key K_B can be considered discoverable data of the second type because it encrypts discoverable data of the first type.

[00313] The discoverable data class 440 can further include a third discoverable data subclass 446 that includes otherwise undiscoverable data that becomes discoverable because it is encrypted with keys that are considered discoverable data of the second type. For example, as shown in the above message sent from Bob to Trent, the data X can be randomly assigned to Bob and known only to Bob and Trent. Therefore, the data X is undiscoverable.

WAVELYNX EX. 1003, Pg. 843
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

However, since the data X is encrypted with Bob's secret key K_B (which is discoverable data of the second type because it is used to encrypt discoverable data (e.g., V_{id})), the data X can be considered discoverable data of the third type.

[00314] Figs. 24 and 25 illustrate an encryption strategy for protecting the security of undiscoverable data. To protect the security of the undiscoverable or secret data, separate keys are used to encrypt the different types of data (hereinafter referred to as "separate encryption protocols"). For example, one or more keys (e.g., one or more mutating IDs) can be used to encrypt the undiscoverable data and one or more keys (e.g., one or more mutating IDs) can be used to encrypt the discoverable data. As described below, since the same keys are never used to encrypt undiscoverable data and discoverable data, discoverable data of the third type is eliminated.

[00315] As shown in Fig. 24, data included in the secret data class 430 can be encrypted with one or more keys 450 that are only used to encrypt data included in the secret data class 430 (hereinafter referred to in this example as "undiscoverable data keys 450"). Optionally, data included in the discoverable data class 440 can be encrypted with one or more keys 460 that are only used to encrypt data included in the discoverable data class 440 (hereinafter referred to in this example as "discoverable data keys 460"). It should be understood that the undiscoverable data keys 450 used to encrypt data included in the secret data class 430 cannot be determined from (or are unrelated to) the discoverable data keys 460 used to encrypt data included in the discoverable data class 440. It should also be understood that the data 420 does not need to be separated and placed in contiguous blocks of data according to the data class the portions of the data 420 belong to. As shown in Fig. 25, data included in the secret data class 430 and the discoverable data class 440 can be divided into a number of portions that are mixed together.

[00316] As noted above, by separating the discoverable data from the undiscoverable data and encrypting the undiscoverable data with undiscoverable data keys 450 that are different from the discoverable data keys 460 used to encrypt the discoverable data, discoverable data of the third type is eliminated since the discoverable data keys 460 are never used to encrypt undiscoverable data. Therefore, even if a discoverable data key 460 used to encrypt discoverable data is determined using a brute force attack, the determined discoverable data key 460 cannot be used to obtain undiscoverable or secret data.

WAVELYNX EX. 1003, Pg. 844
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00317] For example, assume that Bob wants to request a transaction key from Trent to be used to communicate with Vera, and, therefore, Bob sends Trent the data X and Vera's identifier V_{id}. Also assume that Bob has a mutating ID that includes a number N_{B1} and a corresponding key K_{B1} to be used only with undiscoverable data (for this example hereinafter respectively referred to as the "undiscoverable data number N_{B1}" and the "undiscoverable data key K_{B1}") and a mutating ID that includes a number N_{B2} and a key K_{B2} to be used only with discoverable data of the first type and the second type (for this example hereinafter respectively referred to as the "discoverable data number N_{B2}" and the "discoverable data key K_{B2}"). Bob can encrypt Vera's identifier V_{id} with his discoverable data key K_{B2} because Vera's identifier V_{id} is discoverable data of the first type (i.e., it is publicly known). Similarly, Bob can encrypt the data X with his undiscoverable data key K_{B1}. Since Vera's identifier V_{id} is encrypted separately from the data X, the data X is no longer considered discoverable data of the third type as described above because the data X is not encrypted with a key that encrypts discoverable data. The data X, therefore, becomes undiscoverable data. In some embodiments, Bob also appends the undiscoverable data number N_{B1} and the discoverable data number N_{B2} to the separately encrypted portions in order to identify himself and the separate portions of the message to Trent. Bob can send the resulting message to Trent.

$$B \rightarrow T\text{: } N_{B1}E(K_{B1}, X)\ N_{B2}E(K_{B2}, V_{id})$$

[00318] If an eavesdropper obtains the above transmission sent from Bob to Trent, the eavesdropper (theoretically) could obtain the discoverable data key K_{B2}, which encrypted Vera's identifier V_{id}, using a brute-force attack. Knowing the second secret key K_{B2}, however, does not give the eavesdropper access to the undiscoverable data key K_{B1} or, consequently, the data X or any other data also encrypted with the undiscoverable data key K_{B1}. As noted above, Bob could choose not to encrypt Vera's identifier V_{id} and send it as plaintext.

[00319] To complete the key exchange, assume Vera has a mutating ID that includes a number N_{V1} and a corresponding key K_{V1} to be used only with undiscoverable data (hereinafter respectively referred to in this example as the "undiscoverable data number N_{V1}" and the "undiscoverable data key K_{V1}"), a mutating ID that includes a number N_{V2} and a key K_{V2} to be used only with discoverable data (hereinafter respectively referred to in this

WAVELYNX EX. 1003, Pg. 845
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

example as the "discoverable data number N_{V2}" and the "discoverable data key K_{V2}"), and data Y known only to Vera and Trent.

[00320] Trent, who knows the mutating IDs assigned to Bob and Vera, can decrypt the message from Bob and can generate a transaction key K_{BV} for Bob and Vera. Since the transaction key K_{BV} could potentially be used by Bob and/or Vera to encrypt discoverable data of the first type, the transaction key K_{BV} can be considered discoverable data of the second type and can be encrypted with Bob's and Vera's discoverable data keys (e.g, K_{B2} and B_{V2}). Trent provides the transaction key K_{BV} to Bob and Vera in separate responses.

[00321] In some embodiments, Trent's response to Bob includes the transaction key K_{BV}, the data X, a mutating ID that includes a new number N_{B1}' and a new key K_{B1}' for Bob to be used only with undiscoverable data (hereinafter respectively referred to in this example as the "new undiscoverable data number N_{B1}"" and the "new undiscoverable data key K_{B1}""), and a mutating ID that includes a new number N_{B2}' and a new key K_{B2}' for Bob to be used only with discoverable data of the first type and the second type (hereinafter respectively referred to in this example as the "new discoverable data number N_{B2}"" and the "new discoverable data key K_{B2}""). Since the transaction key K_{BV} can be used to encrypt discoverable data of the first type, Trent encrypts the transaction key K_{BV} with Bob's current undiscoverable key K_{B1}. Since the data X is also undiscoverable data, Trent also encrypts the data X with Bob's current undiscoverable data key K_{B1}. Furthermore, since the new undiscoverable data key K_{B1}' for Bob is considered undiscoverable data (e.g., because it is a random, secret key that will only be used to encrypt undiscoverable data), Trent encrypts the new undiscoverable data key K_{B1}' with Bob's current undiscoverable data keys K_{B1}. The new undiscoverable data number N_{B1}' and the new discoverable data number N_{B2}', however, are discoverable data of the first type since they are passed in the clear. Trent, therefore, encrypts the new undiscoverable data number N_{B1}' and the new discoverable data number N_{B2}' with Bob's current discoverable data key K_{B2}. Trent also uses Bob's current discoverable data key K_{B2} to encrypt Bob's new discoverable data key K_{B2}' since it is used to encrypt discoverable data of the first and second type. Optionally, Trent's response can include Vera's identifier V_{ID}, and Trent can use Bob's current discoverable data key K_{B2} to encrypt Vera's identifier V_{id} since it is discoverable data of the first type. In some embodiments, Trent appends Bob's current undiscoverable data number N_{B1} to the portion of the response encrypted with Bob's current undiscoverable data key K_{B1} and appends Bob's current discoverable data number N_{B2} to the

WAVELYNX EX. 1003, Pg. 846
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

portion of the response encrypted with Bob's current discoverable data key K_{B2} in order to identify the separate portions of the response. Trent can then send the response to Bob.

$$T \rightarrow B: N_{B1}E(K_{B1}, K_{B1}'X)\ N_{B2}E(K_{B2}, N_{B1}'\ N_{B2}'\ K_{B2}'\ V_{id}\ K_{BV})$$

[00322] Trent can generate and send a similar response to Vera.

$$T \rightarrow V: N_{V1}E(K_{V1}, K_{V1}'\ Y)\ N_{V2}E(K_{V2}, N_{V1}'\ N_{V2}'\ K_{V2}'\ B_{id}\ K_{BV})$$

[00323] The protocol described above can be generalized for any protocol on a message-by-message basis. For example, let $N_xE(K_x,D_1D_2...)$ be a message to be transmitted wherein N_x is an optional parameter. To transmit the message using the separation encryption protocols described above, divide the data D_1, D_2,... into undiscoverable data and the three types of discoverable data based on the original format of the message. For example, let D_1*, D_2*,...be undiscoverable data, let D_1^+, D_2^+,...be discoverable data of the first type and the second type, and let $D_1^\#$, $D_2^\#$,...be discoverable data of the third type. The message can then be constructed by encrypting the undiscoverable data (i.e., D_1*, D_2*,...) and the discoverable data of the third type (i.e., $D_1^\#$, $D_2^\#$,...) with a first key *K1* and encrypting the discoverable data of the first type and the second type (i.e., D_1^+, D_2^+,...) with a second key *K2*, wherein the first key *K1* and the second key *K2* are different and computationally underivable from each other.

[00324] In some embodiments, an entity may want to further protect discoverable data of the second type. For example, as described above, when Trent provides the transaction key K_{BV} to Bob, Trent encrypts the transaction key K_{BV} with Bob's current discoverable data key K_{B2}. The discoverable data key K_{B2}, however, is also used to encrypt Vera's identifier V_{id}, which is publicly known. An eavesdropper, therefore, can perform a brute force attack on the portion of Trent's response encrypted with Bob's current discoverable data key K_{B2} and obtain, not only K_{B2}, but also the transaction key K_{BV}. Using this technique, an eavesdropper can obtain the transaction key K_{BV} before it is used by either Bob or Vera or without Bob and/or Vera using the transaction key K_{BV} to encrypt discoverable data.

[00325] To attempt to overcome the above issue, discoverable data of the second type can be encrypted with a key separate from the key used to encrypt discoverable data of the first type. In some embodiments, an entity is assigned a separate mutating ID that includes a

WAVELYNX EX. 1003, Pg. 847
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

number and secret key that are only used to encrypt discoverable data of the second type. For example, assuming that Bob has an alternate mutating ID with an alternate number N_{B3} and an alternate key K_{B3} to be used only to encrypt discoverable data of the second type, the above response sent from Trent to Bob can be modified to include the transaction key K_{BV} encrypted with the alternate key K_{B3}.

$$T \rightarrow B: N_{B1}E(K_{B1}, K_{B1}' X)\ N_{B2}E(K_{B2}, N_{B1}'\ N_{B2}'\ K_{B2}'\ V_{id})\ N_{B3}E(K_{B3}, K_{B3}'K_{BV})$$

[00326] By separately encrypting the transaction key K_{BV} and the discoverable data of the first type (e.g., N_{B1}', N_{B2}', and V_{id}), an eavesdropper cannot obtain the transaction key K_{BV} by performing a brute force attack on Trent's response to Bob. It should be understood that the discoverable data key K_{B2}' and/or other discoverable data of the second type can also be encrypted with the alternate key K_{B3} rather than being encrypted with the discoverable data key K_{B2} in order to prevent a brute force attack on the response from revealing the discoverable data of the second type.

[00327] In some embodiments, rather than receiving the alternate mutating ID for encrypting discoverable data of the second type, an entity receives a single alternate key that may or may not mutate after each use or on another schedule (e.g., after a number of uses, after a specific time span, etc.). The alternate key is a secret key known only to an authenticator and the entity assigned the alternate key. In some embodiments, an entity uses its alternate key to directly encrypt discoverable data of the second type. In other embodiments, an entity uses its alternate key and undiscoverable data key to generate a new key to be used to encrypt discoverable data of the second type. For example, assume that Bob wants to initiate a transaction key exchange with Vera and, therefore, Bob sends Trent Vera's identifier V_{id} and the data X. Also assume that Bob has a mutating ID that includes a number N_{B1} and a corresponding data key K_{B1} to be used only with undiscoverable data (for this example hereinafter respectively referred to as the "undiscoverable data number N_{B1}" and the "undiscoverable data key K_{B1}"), a mutating ID that includes a number N_{B2} and a key K_{B2} to be used only with discoverable data of the first type (for this example hereinafter respectively referred to as the "discoverable data number N_{B2}" and the "discoverable data key K_{B2}"), and an alternate key L_B used only to encrypt discoverable data of the second type, such as the transaction key K_{BV}. In some embodiments, Bob uses his discoverable data key K_{B2} as his alternate key L_B. In other embodiments, Bob can use the data X as his alternate key L_B.

WAVELYNX EX. 1003, Pg. 848
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Bob's undiscoverable data key K_{B1}, discoverable data key K_{B2}, and alternate key L_B are all known by Trent and can mutate (i.e., be reassigned by Trent) after each use or on another schedule.

[00328] As described above, to initiate the transaction key exchange, Bob generates a request that includes the data X and Vera's identifier V_{id}. Bob can separate and encrypt the request as described above. Bob sends the request to Trent.

[00329] When Trent receives the request, Trent can generate the requested transaction key K_{BV}. To send Bob the transaction key K_{BV}, Trent can generate an *XOR* of Bob's undiscoverable data key K_{B1} and Bob's alternate key K_{B3} (or L_B) in order to generate a new encryption key.

$$XOR(K_{B1}, K_{B3})$$

[00330] The *XOR* operation performs a bitwise exclusive "or" of Bob's undiscoverable data key K_{B1} and Bob's alternate key K_{B3}. Therefore, the *XOR* operation generates a bit string equal in length to the length of the longer key between Bob's undiscoverable data key K_{B1} and Bob's alternate key K_{B3}. Each bit position of the generated bit string is equal to "0" if the corresponding positions of the Bob's undiscoverable data key K_{B1} and Bob's alternate key K_{B3} are the same (i.e., both "0" or both "1") or set equal to "1" if the corresponding positions of the Bob's undiscoverable data key K_{B1} and Bob's alternate key K_{B3} are the not the same. The *XOR* operation is generally irreversible unless one of the inputs (e.g., Bob's undiscoverable data key K_{B1} or Bob's alternate key K_{B3}) is known.

[00331] Trent uses the generated bit string (i.e., the result of the *XOR* operation) to encrypt the transaction key K_{BV}. Trent sends the encrypted transaction key K_{BV} in a response to Bob. Trent can separate and encrypt the response as described above.

$$T \rightarrow B: N_{B1}E(K_{B1}, K_{B1}'X)\ N_{B2}E(K_{B2}, N_{B1}'\ N_{B2}'\ K_{B2}'\ V_{id})\ E(XOR(K_{B1}, K_{B3}), K_{BV})$$

[00332] Bob can perform the same *XOR* operation as Trent performed in order to generate the decryption key and obtain the transaction key K_{BV}. Even if an eavesdropper were to obtain the transaction key K_{BV} and, through a brute force attack, obtain the encryption key (i.e., the bit string resulting from the *XOR* operation), the eavesdropper cannot use the stolen key to obtain Bob's undiscoverable data key K_{B1}, Bob's alternate key K_{B3}, or the data X.

WAVELYNX EX. 1003, Pg. 849
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Also, if at least one of Bob's undiscoverable data key K_{B1} and Bob's alternate key K_{B3} mutate after each use or on another schedule, the stolen key is not used to encrypt data included in any future messages. Therefore, the stolen key cannot be used to obtain data included in such messages.

[00333] In some embodiments, the different keys used to encrypt discoverable and undiscoverable data are nested in order to provide further security (hereinafter referred to as "nested separate encryption protocols"). For example, assume that Bob sends the following message to Trent as described above.

$$B \rightarrow T: N_{B1}E(K_{B1}, X)\ N_{B2}E(K_{B2}, V_{id})$$

[00334] The message includes two distinct portions. For example, the message includes a first portion that includes the encrypted undiscoverable data $N_{B1}E(K_{B1}, X)$ and a second portion that includes the encrypted discoverable data $N_{B2}E(K_{B2}, V_{id})$. Since the portions are encrypted separately and concatenated together, an eavesdropper can attack the portions separately.

[00335] For example, an eavesdropper can perform a brute force attack on the second portion (i.e., since Vera's identifier V_{id} is publicly known) and can obtain Bob's discoverable data key K_{B2}. Once the eavesdropper has Bob's discoverable data key K_{B2}, the eavesdropper can obtain the details of the request Bob sent to Trent. Perhaps more damaging, the eavesdropper can also perform a man-in-the-middle attack. A man-in-the-middle attack occurs when an eavesdropper intercepts a transmitted data and attempts to read and/or to alter the data and often does so without the data sender and the intended data receiver knowing. For example, using Bob's stolen discoverable data key K_{B2}, an eavesdropper can alter or reconstruct the second portion of Bob's message by replacing the second portion of Bob's message with their identifier encrypted with Bob's discoverable data key K_{B2}. The eavesdropper can then send the altered message to Trent.

[00336] When Trent obtains the altered message, he decrypts each portion and performs the requested processing. Since the first portion and the second portion of the message are essentially unrelated, Trent has no way of determining whether the credentials included in the second portion are legitimate.

WAVELYNX EX. 1003, Pg. 850
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[00337] To overcome the above situation, Bob can use the undiscoverable data key K_{B1} and the discoverable data key K_{B2} to perform nested encryption. For example, assume that Bob generates a request for Trent that includes the data X and Vera's identifier V_{id}. Bob can encrypt the data X (i.e., the undiscoverable data) with the undiscoverable data key K_{B1}. In some embodiments, Bob appends his undiscoverable data number N_{B1} to the encryption result. Bob can then append Vera's identifier V_{id} to the encrypted data X and encrypt the result with the discoverable data key K_{B2}. Bob can then append his discoverable data number N_{B2} to the result.

$$B \rightarrow T: N_{B2}E(K_{B2}, V_{id}\, E(K_{B1}, X))$$

[00338] The above request is dependent on both the undiscoverable data and the discoverable data since both forms of data are included in the final encryption step. It should be understood that the nesting can be performed in either direction. For example, Bob can encrypt Vera's identifier V_{id} with the discoverable data key K_{B2}, append the data X to the result, and encrypt the combination with the undiscoverable data key K_{B1}.

$$B \rightarrow T: N_{B1}E(K_{B1}, X\, E(K_{B2}, V_{id}))$$

[00339] As shown in the above messages, in some embodiments, an entity may only need one mutating ID number (e.g., N_{B1}) to identify itself as the message originator when the mutating ID keys (e.g., K_{B1} and K_{B2}) are nested. For example, Bob can be assigned one mutating ID that includes one number N_B and one secret key K_B, which Bob can use to encrypt solely undiscoverable data or solely discoverable data and a single, separate key, such as an alternate key L_B, which Bob can use to encrypt the type of data not encrypted by the secret key K_B of the mutating ID. Bob can use the alternate key L_B to encrypt data nested within the data encrypted with the secret key K_B. Bob can then append his number N_B to the result.

$$B \rightarrow T: N_BE(K_B, X\, E(L_B, V_{id}))$$

[00340] It should be understood that the protocols (e.g., mutating IDs protocols, separation encryption protocols, and nested separate encryption protocols) described above can include features that make brute force attacks difficult or reduce the usefulness of such attacks. For example, the above protocols can use strong encryption techniques that may require days,

WAVELYNX EX. 1003, Pg. 851
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

months, or years before a brute force is successful and a key is discovered. Thus, what might appear as avenues of attack in a theoretical sense may be difficult or impossible in a practical sense.

[00341] Also, since a number of messages may be transmitted between an entity and the authenticator during the time required for an eavesdropper to perform a brute force attack, the mutating ID associated with the message intercepted by the eavesdropper can be different from a current mutating ID assigned to the entity. Therefore, an eavesdropper attempting to uncover a current mutating ID assigned to an entity must track and store every message transmitted between that entity and the authenticator in order to trace through messages and obtain a current mutating ID key. For example, if an eavesdropper attempts to determine a mutating ID of an entity using a brute force attack, the authenticator may reassign or mutate the mutating ID multiple times while the eavesdropper is performing the brute force attack. If the eavesdropper desires the key of the most recent mutating ID, once the eavesdropper obtains a past and now invalid mutating ID, he or she must use the uncovered mutating ID to decrypt each message transmitted by the authenticator to the entity and obtain each mutated or reassigned mutating ID in order to determine the currently assigned mutating ID. The tracking and storing requirements to perform such a trace are likely to be very challenging to any eavesdropper attempting to perform a brute force attack.

[00342] In addition, an eavesdropper may have to perform multiple brute force attacks in order to discover "useful" information. For example, in the separation encryption protocol examples described above, an eavesdropper may be able to perform a brute force attack on a message encrypted with a transaction key (e.g., K_{BV}) and, therefore, may be able to obtain the transaction key (e.g., K_{BV}). To obtain a secret key of an entity using the transaction key, however, the eavesdropper is required to perform a second brute force attack on the message sent from the authenticator that included the now known transaction key, which was encrypted with the entity's secret key. As described above, a single brute force attack may require days, months, or year to compute and, therefore, a second brute force attack may double the time needed to obtain "useful" information (e.g., a secret key of a mutating ID). Furthermore, an eavesdropper may have to perform nested brute force attacks in order to obtain "useful" information, where an eavesdropper is required to try each of a first candidate key and, for each first candidate key, try each of a second candidate key. For example, in the above nested separate encryption protocol examples, to obtain the data X from the following

WAVELYNX EX. 1003, Pg. 852
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

message, an eavesdropper would be required to try every candidate key for the outer encryption result $E(K_B, XE(L_B, V_{id})$ and, for each candidate key of the outer encryption result, try every candidate key for the nested encryption result $E(L_B, V_{ID})$.

$$B \rightarrow T: N_B\, E(K_B, XE(L_B, V_{id}))$$

[00343] It should be understood that if a single brute force attack requires N processing time units, a single nested brute force attack (i.e., a single brute force attack nested inside another brute force attack) can require approximately N^2 processing time units, which may make a brute force attack impractical or impossible for an eavesdropper in terms of processing requirements.

[00344] As can be seen from the above, various embodiments provide a system and method of distributing content and information with features to protect security and the legal rights of content owners. Embodiments also provide a system and method of performing electronic commerce with features to protect security and the confidential, financial information of transaction participants. Additional features and aspects of embodiments of the invention are set forth in the attached claims.

WAVELYNX EX. 1003, Pg. 853
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

CLAIMS

1. A method of performing electronic commerce using mutating identifiers.

2. The method of claim 1, wherein the method further comprises:

encrypting buyer transaction data with a first mutating identifier;

transmitting the buyer transaction data to an authenticator device;

decrypting the buyer transaction data;

generating a payment request;

encrypting the payment request with a third mutating identifier; and

transmitting the payment request to a payment authenticator device.

3. The method of claim 2, further comprising encrypting vendor transaction data with a second mutating identifier and transmitting the vendor transaction data to the authenticator device.

4. The method of claim 3, further comprising decrypting the vendor transaction data.

5. The method of claim 2, wherein generating a payment request includes generating a payment request including an identity of a vendor device and vendor credentials.

6. The method of claim 2, further comprising transmitting a second receipt to a vendor device.

7. The method of claim 2, further comprising transmitting a fifth mutating identifier to a vendor device.

8. The method of claim 2, further comprising transmitting an approved message to a vendor device if the payment authenticator device approves the payment request.

9. The method of claim 2, further comprising transmitting a declined message to a vendor device if the payment authenticator device declines the payment request.

WAVELYNX EX. 1003, Pg. 854
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

10. The method of claim 2, wherein generating a payment request includes generating a payment request including an identity of a buyer device and buyer credentials.

11. The method of claim 2, further comprising transmitting a first receipt to a buyer device.

12. The method of claim 2, further comprising transmitting a fourth mutating identifier to a buyer device.

13. The method of claim 2, further comprising transmitting an approved message to a buyer device if the payment authenticator device approves the payment request.

14. The method of claim 2, further comprising transmitting a declined message to a buyer device if the payment authenticator device declines the payment request.

15. The method of claim 2, further comprising transmitting a third receipt to the payment authenticator device.

16. The method of claim 2, further comprising transmitting a sixth mutating identifier to the payment authenticator device.

17. The method of claim 2, further comprising transferring funds between a first account and a second account if the payment authenticator device approves the payment request.

18. The method of claim 2, further comprising providing an escrow service if the payment authenticator device approves the payment request.

WAVELYNX EX. 1003, Pg. 855
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

19. A method for establishing communication between a first device and a second device, the method comprising:

generating a request for a transaction key;

encrypting the request for a transaction key with a first mutating identifier of the first device;

transmitting the encrypted request for a transaction key to an authenticator device;

generating a transaction key;

generating a first message including the transaction key; and

encrypting the first message with the first mutating identifier of the first device.

20. The method of claim 19, further comprising generating a third mutating identifier for the first device, wherein the first message includes the third mutating identifier.

21. The method of claim 20, further comprising transmitting the first message to the first device.

22. The method of claim 19, further comprising generating a second message including the transaction key and encrypting the second message with a second mutating identifier of the second device.

23. The method of claim 22, further comprising generating a fourth mutating identifier for the second device, wherein the second message includes the fourth mutating identifier.

24. The method of claim 23, further comprising transmitting the second message to the second device.

25. The method of claim 19, wherein generating a request for a transaction key includes generating a request for a transaction key wherein the request for a transaction key includes request-identifying data.

WAVELYNX EX. 1003, Pg. 856
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

26. The method of claim 25, wherein generating a first message including the transaction key includes generating a first message including the transaction key and the request-identifying data.

WAVELYNX EX. 1003, Pg. 857
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

27. A method of performing electronic commerce using mutating identifiers, the method comprising:

encrypting buyer transaction data with a first mutating identifier;

transmitting the buyer transaction data to an authenticator device;

encrypting buyer credentials with a first transaction key;

transmitting the buyer credentials to a payment authenticator device;

encrypting vendor transaction data with a second mutating identifier;

transmitting the vendor transaction data to an authenticator device;

encrypting vendor credentials with a second transaction key;

transmitting the vendor credentials to a payment authenticator device;

decrypting the buyer transaction data;

decrypting the vendor transaction data;

generating a payment request for a payment authenticator device;

encrypting the payment request with a third mutating identifier;

transmitting the payment request to the payment authenticator device;

decrypting the payment request;

decrypting the buyer credentials;

decrypting the vendor credentials;

generating a first response based on the buyer credentials, the vendor credentials, and the payment request;

generating a second response based on the buyer credentials, the vendor credentials, and the payment request;

WAVELYNX EX. 1003, Pg. 858
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

transmitting the first response to a buyer device; and

transmitting the second response to a vendor device.

28. The method of claim 27, wherein encrypting vendor transaction data includes encrypting a bill of sale and a price.

29. The method of claim 27, wherein generating a payment request includes generating a payment request including an identity of the vendor device and an identity of the buyer device.

30. The method of claim 27, further comprising verifying the buyer credentials and verifying the vendor credentials.

31. The method of claim 27, wherein generating a payment request includes generating a payment request including a first receipt for the buyer device.

32. The method of claim 31, wherein generating a first response includes generating a first response including the first receipt.

33. The method of claim 27, wherein generating a payment request includes generating a payment request including a fourth mutating identifier for the buyer device.

34. The method of claim 33, wherein generating a first response includes generating a first response including the fourth mutating identifier.

35. The method of claim 27, wherein generating a first response includes generating a first response including at least one of the approved message and the declined message for the buyer device.

36. The method of claim 27, wherein encrypting buyer transaction data includes encrypting a bill of sale and a price.

37. The method of claim 27, wherein encrypting buyer transaction data includes encrypting an identity of the payment authenticator device.

38. The method of claim 27, wherein generating a payment request includes generating a payment request including a second receipt for the vendor device.

WAVELYNX EX. 1003, Pg. 859
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

39. The method of claim 38, wherein generating a second response includes generating a second response including the second receipt.

40. The method of claim 27, wherein generating a payment request includes generating a payment request including a fifth mutating identifier for the vendor device.

41. The method of claim 40, wherein generating a second response includes generating a second response including the fifth mutating identifier.

42. The method of claim 27, wherein generating a second response includes generating a second response including at least one of an approved message and a declined message for the vendor device.

43. The method of claim 27, wherein generating a payment request includes generating a payment request including a third receipt for the payment authenticator device.

44. The method of claim 27, wherein generating a payment request includes generating a payment request including a sixth mutating identifier for the payment authenticator device.

45. The method of claim 27, further comprising transferring funds from a first account to a second account if the payment authenticator approves the payment request.

46. The method of claim 27, further comprising providing an escrow service if the payment authenticator approves the payment request.

WAVELYNX EX. 1003, Pg. 860
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

47. An electronic commerce system comprising:

a vendor device;

a buyer device configured to encrypt buyer transaction data with a first mutating identifier and to transmit the buyer transaction data to an authenticator device; and

a payment authenticator device configured to approve or decline a payment request, to generate a first response for the buyer device, to generate a second response for the vendor device, to transmit the first response to the buyer device, and to transmit the second response to the vendor device; wherein

the authenticator device is configured to decrypt the buyer transaction data, to generate a payment request for the payment authenticator device, to encrypt the payment request with a third mutating identifier of the payment authenticator device, and to transmit the payment request to the payment authenticator device.

48. The system of claim 47, wherein the vendor device is further configured to encrypt vendor transaction data with a second mutating identifier and to transmit the vendor transaction data to the authenticator device.

49. The system of claim 48, wherein the authenticator device is further configured to decrypt the vendor transaction data.

50. The system of claim 47, wherein the vendor device is further configured to encrypt vendor credentials with a second transaction key and to transmit the vendor credentials to the payment authenticator device.

51. The system of claim 50, wherein the payment authenticator device is further configured to decrypt the vendor credentials.

52. The system of claim 50, wherein the payment authenticator device is further configured to encrypt the second response with the second transaction key.

53. The system of claim 47, wherein the authenticator device is further configured to include vendor credentials in the payment request.

WAVELYNX EX. 1003, Pg. 861
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

54. The system of claim 53, wherein the payment authenticator device is further configured to verify the vendor credentials.

55. The system of claim 47, wherein the authenticator device is further configured to include a fifth mutating identifier for the vendor device in the payment request.

56. The system of claim 55, wherein the payment authenticator device is further configured to transmit the fifth mutating identifier to the vendor device.

57. The system of claim 47, wherein the payment authenticator device is further configured to include at least one of an approved message and a declined message in the second response.

58. The system of claim 47, wherein the buyer device is further configured to encrypt buyer credentials with a first transaction key and to transmit the buyer credentials to the payment authenticator device.

59. The system of claim 58, wherein the payment authenticator is further configured to decrypt the buyer credentials.

60. The system of claim 58, wherein the payment authenticator device is further configured to encrypt the first response with the first transaction key.

61. The system of claim 47, wherein the authenticator device is further configured to include buyer credentials in the payment request.

62. The system of claim 61, wherein the payment authenticator device is further configured to verify the buyer credentials.

63. The system of claim 47, wherein the authenticator device is further configured to include a fourth mutating identifier for the buyer device in the payment request.

64. The system of claim 63, wherein the payment authenticator device is further configured to transmit the fourth mutating identifier to the buyer device.

WAVELYNX EX. 1003, Pg. 862
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

65. The system of claim 47, wherein the payment authenticator device is further configured to include at least one of an approved message and a declined message in the first response.

66. The system of claim 47, wherein the authenticator device is further configured to include a third receipt for the payment authenticator device in the payment request.

67. The system of claim 47, wherein the authenticator device is further configured to include a sixth mutating identifier for the payment authenticator device in the payment request.

68. The system of claim 47, wherein the payment authenticator device is further configured to transfer funds between a first account and a second account upon approving the payment request.

69. The system of claim 47, wherein the payment authenticator device is further configured to provide an escrow service upon approving the payment request.

WAVELYNX EX. 1003, Pg. 863
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

70. A method of performing electronic commerce using mutating identifiers, the method comprising:

encrypting transaction data with a first mutating identifier;

transmitting the transaction data to an authenticator device; and

receiving a response from a payment authenticator device.

71. A method of performing electronic commerce using mutating identifiers, the method comprising:

obtaining a payment request from an authenticator device including transaction data, the payment request encrypted with a first mutating identifier;

decrypting the payment request;

verifying the transaction data; and

transmitting a response.

72. A method of performing electronic commerce using mutating identifiers, the method comprising:

obtaining transaction data encrypting with a first mutating identifier;

decrypting the transaction data;

generating a payment request including at least a portion of the transaction data; and

transmitting the payment request to a payment authenticator device.

73. A method of establishing communication between a first device and a second device, the method comprising:

generating a request for a transaction key;

encrypting the request for a transaction key with a first mutating identifier of the first device;

WAVELYNX EX. 1003, Pg. 864
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

transmitting the encrypted request for a transaction key to an authenticator device; and

receiving a message including a transaction key encrypting with the first mutating identifier.

74. A method of establishing communication between a first device and a second device, the method comprising:

obtaining a request for a transaction key from the first device, the request for a transaction key encrypted in a first mutating identifier;

decrypting the request for a transaction key;

generating a first message including the transaction key;

encrypting the first message with the first mutating identifier of the first device; and

transmitting the first message to the first device.

75. A method of conducting electronic commerce, the method comprising:

encrypting buyer transaction data with a first buyer key;

transmitting the buyer transaction data to an authenticator device;

encrypting buyer credentials with a first transaction key; and

transmitting the buyer credentials to a payment authenticator device.

76. A method as claimed in claim 75, further comprising:

encrypting vendor transaction data with a vendor key;

transmitting the vendor transaction data to an authenticator device;

encrypting vendor credentials with a second transaction key; and

transmitting the vendor credentials to a payment authenticator device.

77. A method as claimed in claim 76, further comprising:

WAVELYNX EX. 1003, Pg. 865
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

decrypting the buyer transaction data;

decrypting the vendor transaction data;

generating a payment request for a payment authenticator device;

encrypting the payment request with a payment request key; and

transmitting the payment request to the payment authenticator device.

78. A method as claimed in claim 77, further comprising:

decrypting the payment request;

decrypting the buyer credentials;

decrypting the vendor credentials;

generating a first response based on the buyer credentials, the vendor credentials, and the payment request;

generating a second response based on the buyer credentials, the vendor credentials, and the payment request;

transmitting the first response to the buyer device; and

transmitting the second response to the buyer device.

WAVELYNX EX. 1003, Pg. 866
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

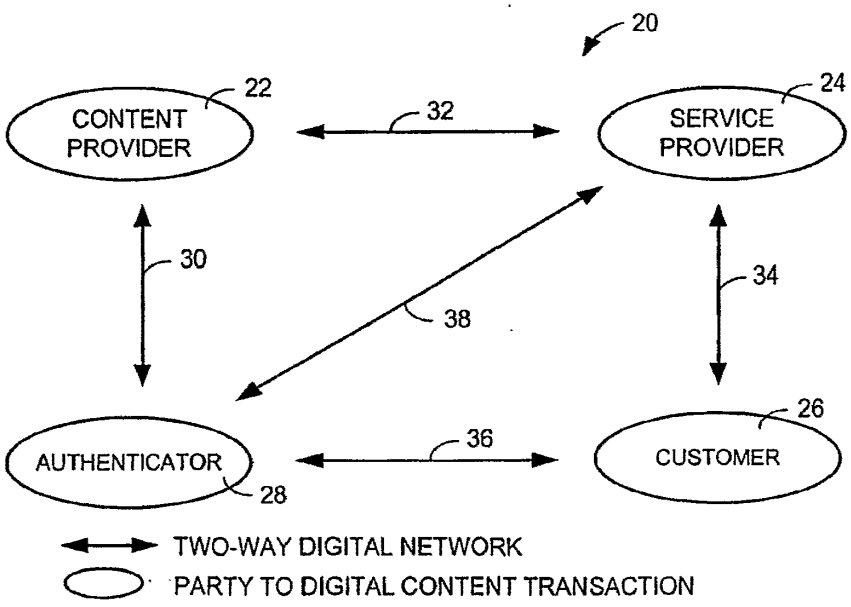

20
22
CONTENT PROVIDER
32
24
SERVICE PROVIDER
30
34
38
26
AUTHENTICATOR
36
CUSTOMER
28
TWO-WAY DIGITAL NETWORK
PARTY TO DIGITAL CONTENT TRANSACTION

FIG. 1

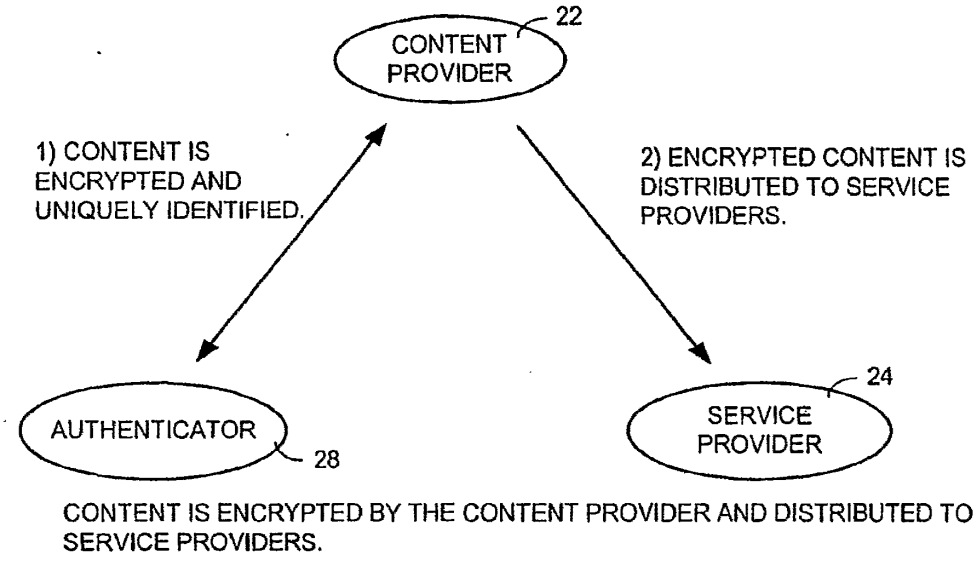

22
CONTENT PROVIDER
1) CONTENT IS ENCRYPTED AND UNIQUELY IDENTIFIED.
2) ENCRYPTED CONTENT IS DISTRIBUTED TO SERVICE PROVIDERS.
24
SERVICE PROVIDER
AUTHENTICATOR
28
CONTENT IS ENCRYPTED BY THE CONTENT PROVIDER AND DISTRIBUTED TO SERVICE PROVIDERS.

FIG. 2

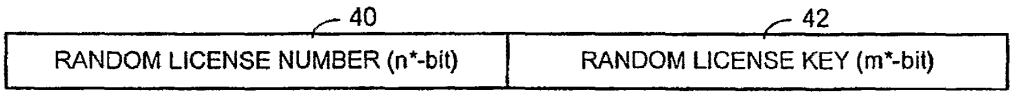
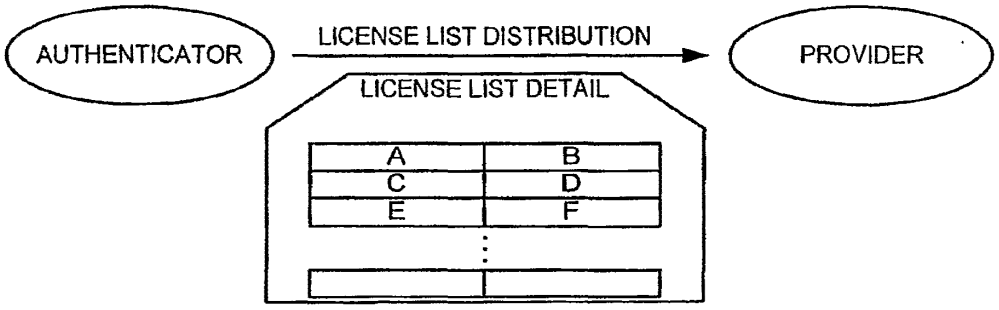

40
42
RANDOM LICENSE NUMBER (n*-bit)
RANDOM LICENSE KEY (m*-bit)
* IN THE REFERENCED IMPLEMENTATION n AND m ARE BOTH 256

FIG. 3A

THERE ARE TWO METHODS OF PACKAGING LICENSES. 1) FOR EACH PROVIDER, THE AUTHENTICATOR GENERATES A LIST OF LICENSE PAIRS.



THE LIST MAY BE DISTRIBUTED EITHER BY HAND, PUBLIC KEY ENCRYPTION, OR ENCODING WITH A LICENSE KEY FROM A PREVIOUSLY DISTRIBUTED LIST.

FIG. 3B

2) FOR EACH CONSUMER, A SINGLE MUTATING ID IS DISTRIBUTED.

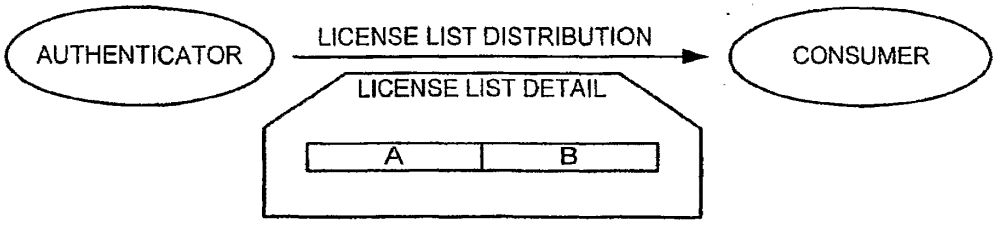


FIG. 3C

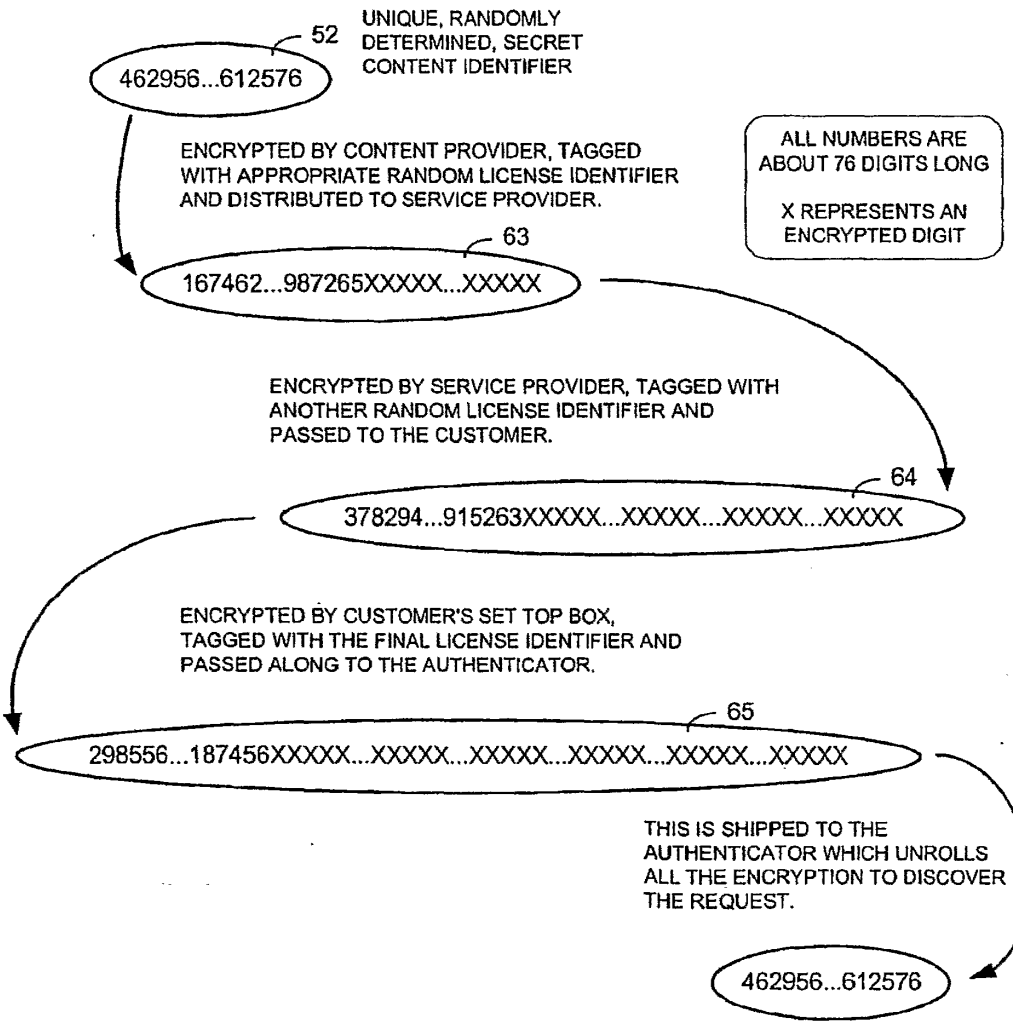
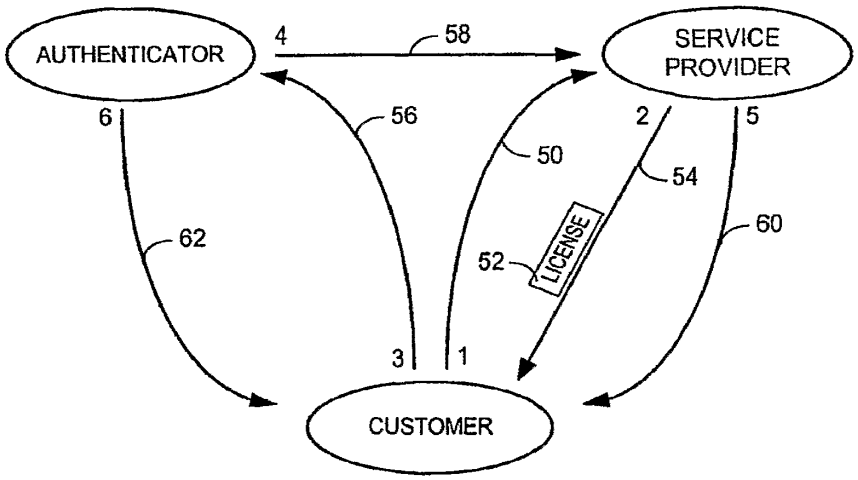

UNIQUE, RANDOMLY DETERMINED, SECRET CONTENT IDENTIFIER
52
462956...612576
ENCRYPTED BY CONTENT PROVIDER, TAGGED WITH APPROPRIATE RANDOM LICENSE IDENTIFIER AND DISTRIBUTED TO SERVICE PROVIDER.
ALL NUMBERS ARE ABOUT 76 DIGITS LONG
X REPRESENTS AN ENCRYPTED DIGIT
63
167462...987265XXXXX...XXXXX
ENCRYPTED BY SERVICE PROVIDER, TAGGED WITH ANOTHER RANDOM LICENSE IDENTIFIER AND PASSED TO THE CUSTOMER.
64
378294...915263XXXXX...XXXXX...XXXXX...XXXXX
ENCRYPTED BY CUSTOMER'S SET TOP BOX, TAGGED WITH THE FINAL LICENSE IDENTIFIER AND PASSED ALONG TO THE AUTHENTICATOR.
65
298556...187456XXXXX...XXXXX...XXXXX...XXXXX...XXXXX...XXXXX
THIS IS SHIPPED TO THE AUTHENTICATOR WHICH UNROLLS ALL THE ENCRYPTION TO DISCOVER THE REQUEST.
462956...612576

FIG. 4



1 CUSTOMER MAKES A REQUEST FROM THE SERVICE PROVIDER TO VIEW SOME CONTENT.

2 THE SERVICE PROVIDER SENDS TO THE CUSTOMER AN ENCRYPTED LICENSE RECEIVED FROM THE CONTENT PROVIDER AND ENCRYPTED BY THE SERVICE PROVIDER.

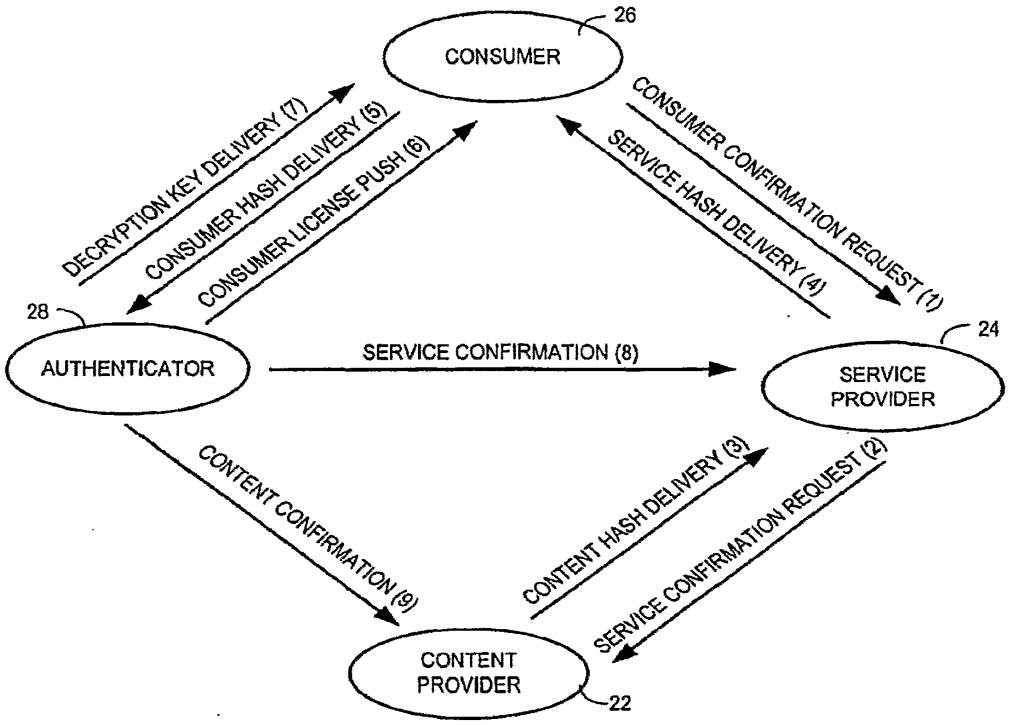
3 THE CUSTOMER'S SET TOP BOX OR DEVICE ENCRYPTS THE LICENSE WITH ITS OWN LICENSE AND SENDS IT TO THE AUTHENTICATOR.

4 THE AUTHENTICATOR CHECKS THE LICENSE FOR VALIDITY. IF VALID, THE AUTHENTICATOR TELLS THE SERVICE PROVIDER THAT THE CUSTOMER HAS AGREED TO PAY FOR THE CONTENT.

5 THE CONTENT PROVIDER THEN SENDS THE ENCRYPTED CONTENT TO THE CUSTOMER.

6 THE AUTHENTICATOR SENDS DECODING INFORMATION.

FIG. 5



(1) CONSUMER MAKES A REQUEST FOR A MOVIE OR CONTENT.
(2) THE DISTRIBUTOR OR SERVICE PROVIDER RELAYS REQUEST TO THE PRODUCER OR CONTENT PROVIDER.
(3) THE PRODUCER ENCRYPTS A HASH OF THE CONTENT'S TITLE (OR SOME OTHER IDENTIFYING INFORMATION) USING ONE OF THE LICENSES ACQUIRED EARLIER.
SENDS:

	ENCRYPTED BY PRODUCER KEY
PRODUCER LIC. #	IDENTIFYING INFORMATION FOR CONTENT

(4) DISTRIBUTOR ENCRYPTS MESSAGE FROM PRODUCER WITH ONE OF ITS LICENSES ACQUIRED EARLIER.
SENDS:

	ENCRYPTED BY DISTRIBUTOR KEY	ENCRYPTED BY PRODUCER KEY
DISTRIBUTOR LIC. #	PRODUCER LIC. #	IDENTIFYING INFORMATION FOR CONTENT

(5) CONSUMER ENCRYPTS MESSAGE FROM PRODUCER WITH HIS OWN LICENSE.
SENDS:

	ENCRYPTED BY CONSUMER KEY	ENCRYPTED BY DISTRIBUTOR KEY	ENCRYPTED BY PRODUCER KEY
CONSUMER LIC. #	DISTRIBUTOR LIC. #	PRODUCER LIC. #	IDENTIFYING INFORMATION FOR CONTENT

(6) NEW LICENSE IS SENT TO CONSUMER BY THE KEY MASTER OR AUTHENTICATOR.
(7) PRODUCER IS INFORMED OF THE USE OF HIS LICENSE IF HE SO DESIRES. ALTERNATIVELY, LICENSE USAGE MAY BE TRACKED BY THE AUTHENTICATOR.
(8) DECRYPTION KEY FOR VIDEO OR CONTENT IS DELIVERED TO THE CONSUMER.
(9) DISTRIBUTOR IS INFORMED OF THE USE OF HIS LICENSE (E.G., FOR BILLING PURPOSES) AND IS TOLD TO BEGIN SENDING ENCRYPTED MOVIE DATA OR CONTENT.

FIG. 6

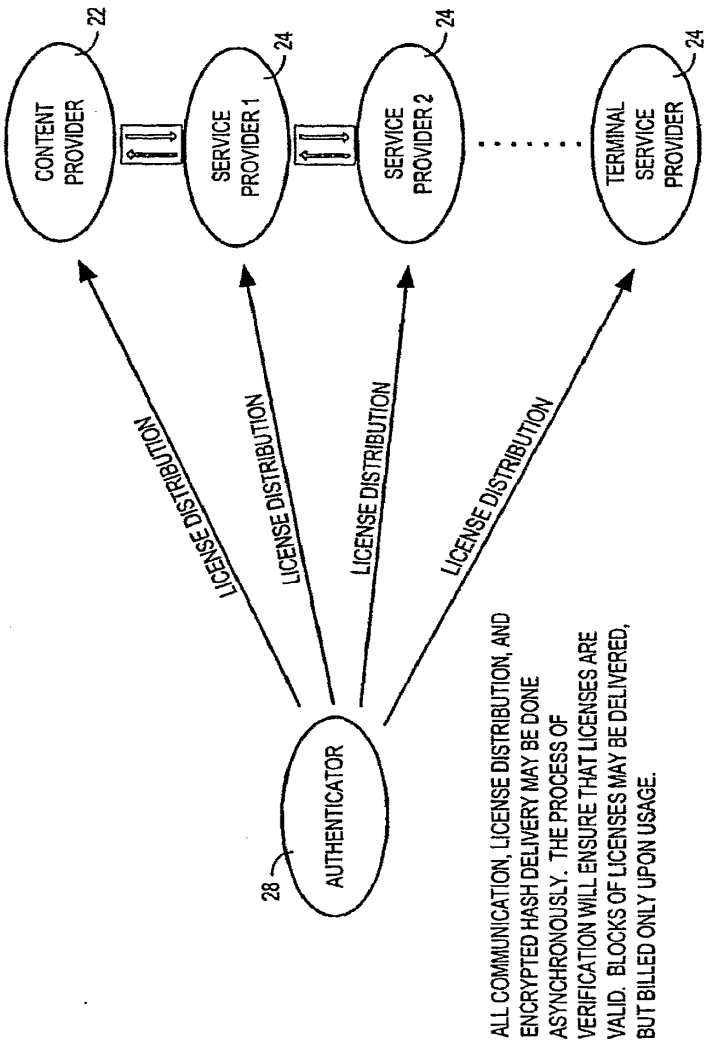

CONTENT PROVIDER
22
SERVICE PROVIDER 1
24
SERVICE PROVIDER 2
24
TERMINAL SERVICE PROVIDER
24
LICENSE DISTRIBUTION
LICENSE DISTRIBUTION
LICENSE DISTRIBUTION
LICENSE DISTRIBUTION
AUTHENTICATOR
28
ALL COMMUNICATION, LICENSE DISTRIBUTION, AND ENCRYPTED HASH DELIVERY MAY BE DONE ASYNCHRONOUSLY. THE PROCESS OF VERIFICATION WILL ENSURE THAT LICENSES ARE VALID. BLOCKS OF LICENSES MAY BE DELIVERED, BUT BILLED ONLY UPON USAGE.

FIG. 7

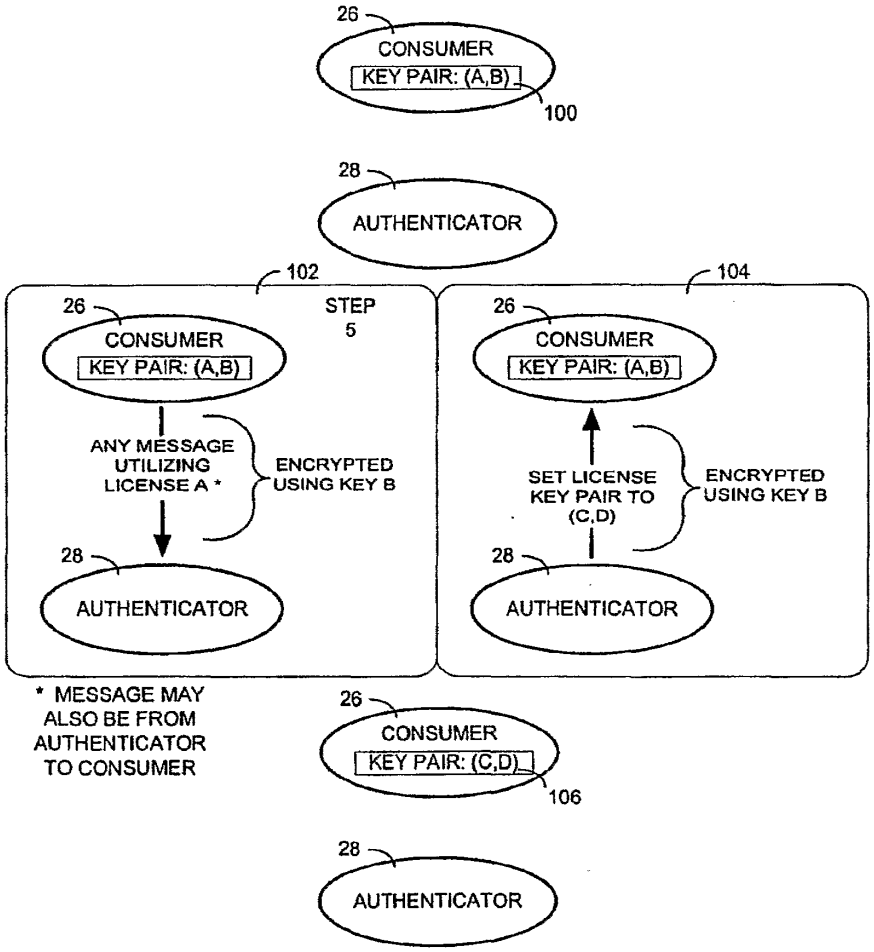

26
CONSUMER
KEY PAIR: (A,B)
100
28
AUTHENTICATOR
102
104
26
CONSUMER
KEY PAIR: (A,B)
STEP
5
ANY MESSAGE
UTILIZING
LICENSE A *
ENCRYPTED
USING KEY B
28
AUTHENTICATOR
26
CONSUMER
KEY PAIR: (A,B)
SET LICENSE
KEY PAIR TO
(C,D)
ENCRYPTED
USING KEY B
28
AUTHENTICATOR
* MESSAGE MAY
ALSO BE FROM
AUTHENTICATOR
TO CONSUMER
26
CONSUMER
KEY PAIR: (C,D)
106
28
AUTHENTICATOR

FIG. 8

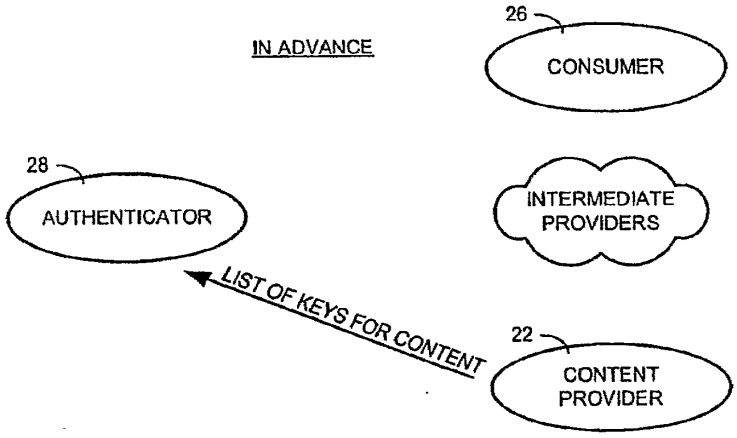

IN ADVANCE
26
CONSUMER
28
AUTHENTICATOR
INTERMEDIATE PROVIDERS
LIST OF KEYS FOR CONTENT
22
CONTENT PROVIDER

FIG. 9A

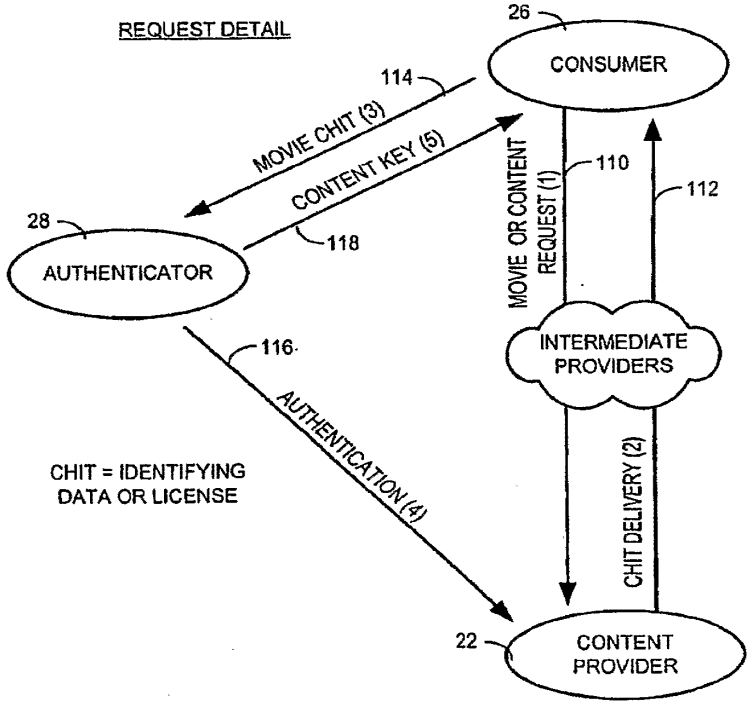

REQUEST DETAIL
26
CONSUMER
114
MOVIE CHIT (3)
CONTENT KEY (5)
118
28
AUTHENTICATOR
MOVIE OR CONTENT REQUEST (1)
110
112
INTERMEDIATE PROVIDERS
116
AUTHENTICATION (4)
CHIT = IDENTIFYING DATA OR LICENSE
CHIT DELIVERY (2)
22
CONTENT PROVIDER

FIG. 9B

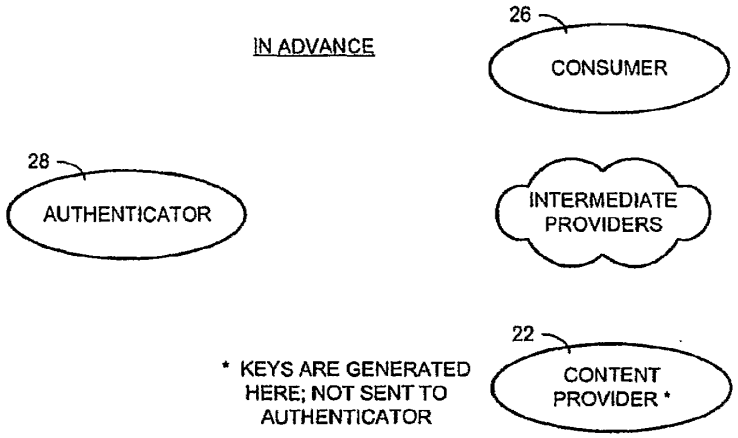


FIG. 10A

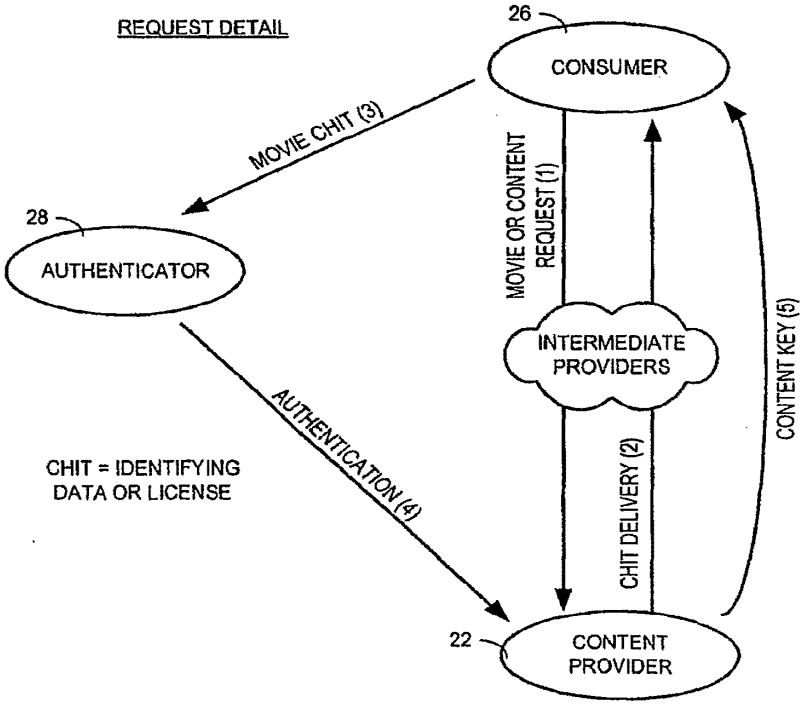


FIG. 10B

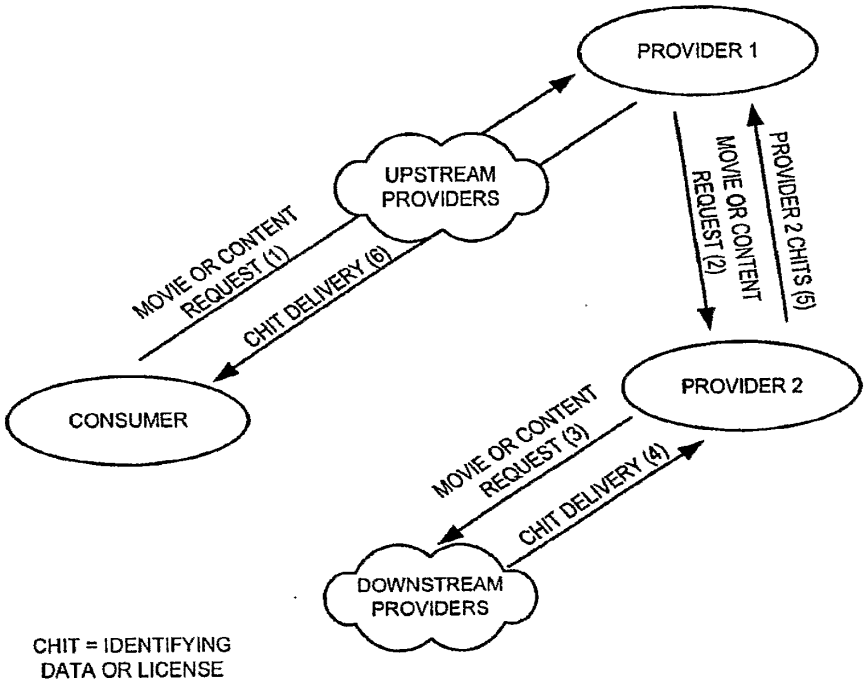

PROVIDER 1
UPSTREAM PROVIDERS
MOVIE OR CONTENT REQUEST (1)
CHIT DELIVERY (6)
MOVIE OR CONTENT REQUEST (2)
PROVIDER 2 CHITS (5)
CONSUMER
PROVIDER 2
MOVIE OR CONTENT REQUEST (3)
CHIT DELIVERY (4)
DOWNSTREAM PROVIDERS
CHIT = IDENTIFYING DATA OR LICENSE

FIG. 11

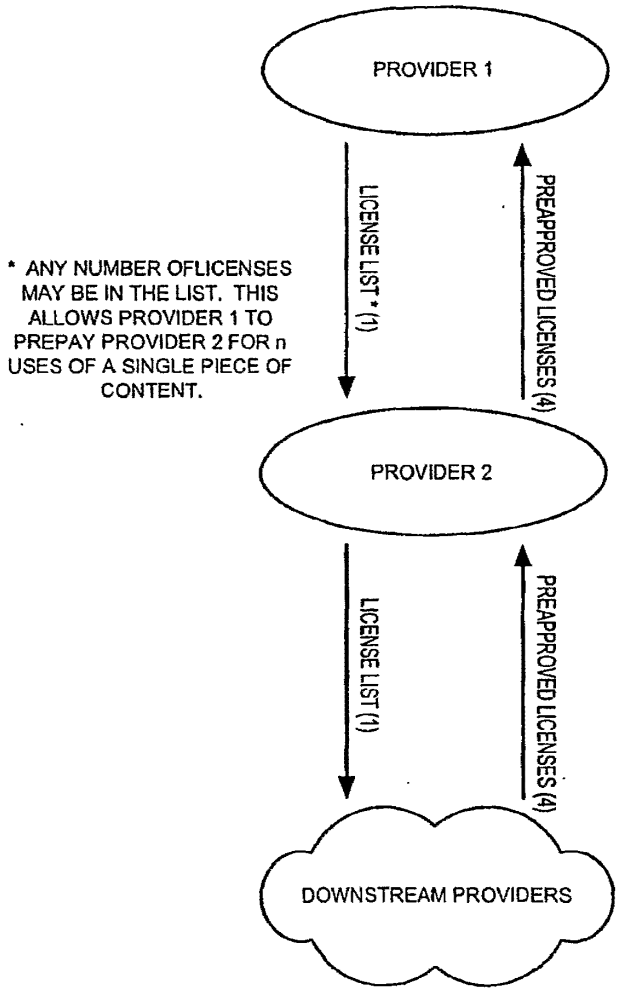

PROVIDER 1
LICENSE LIST * (1)
PREAPPROVED LICENSES (4)
* ANY NUMBER OFLICENSES MAY BE IN THE LIST. THIS ALLOWS PROVIDER 1 TO PREPAY PROVIDER 2 FOR n USES OF A SINGLE PIECE OF CONTENT.
PROVIDER 2
LICENSE LIST (1)
PREAPPROVED LICENSES (4)
DOWNSTREAM PROVIDERS

FIG. 12

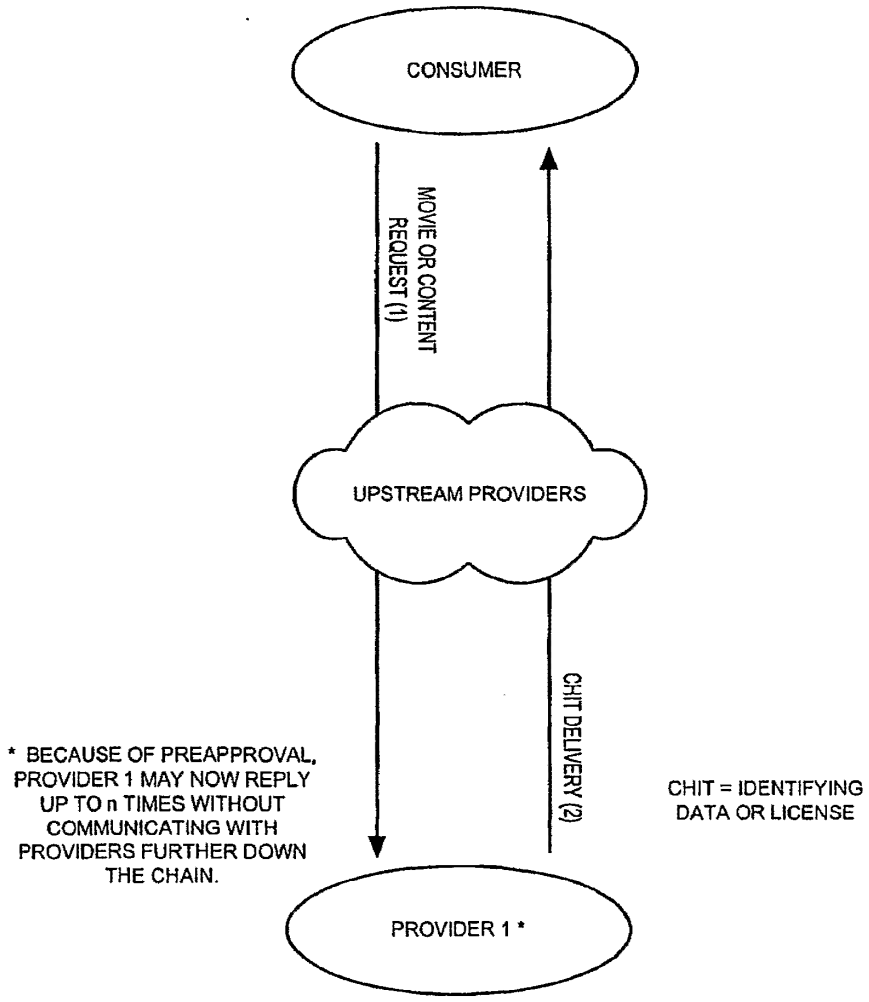

CONSUMER
MOVIE OR CONTENT REQUEST (1)
UPSTREAM PROVIDERS
CHIT DELIVERY (2)
* BECAUSE OF PREAPPROVAL, PROVIDER 1 MAY NOW REPLY UP TO n TIMES WITHOUT COMMUNICATING WITH PROVIDERS FURTHER DOWN THE CHAIN.
CHIT = IDENTIFYING DATA OR LICENSE
PROVIDER 1 *

FIG. 13

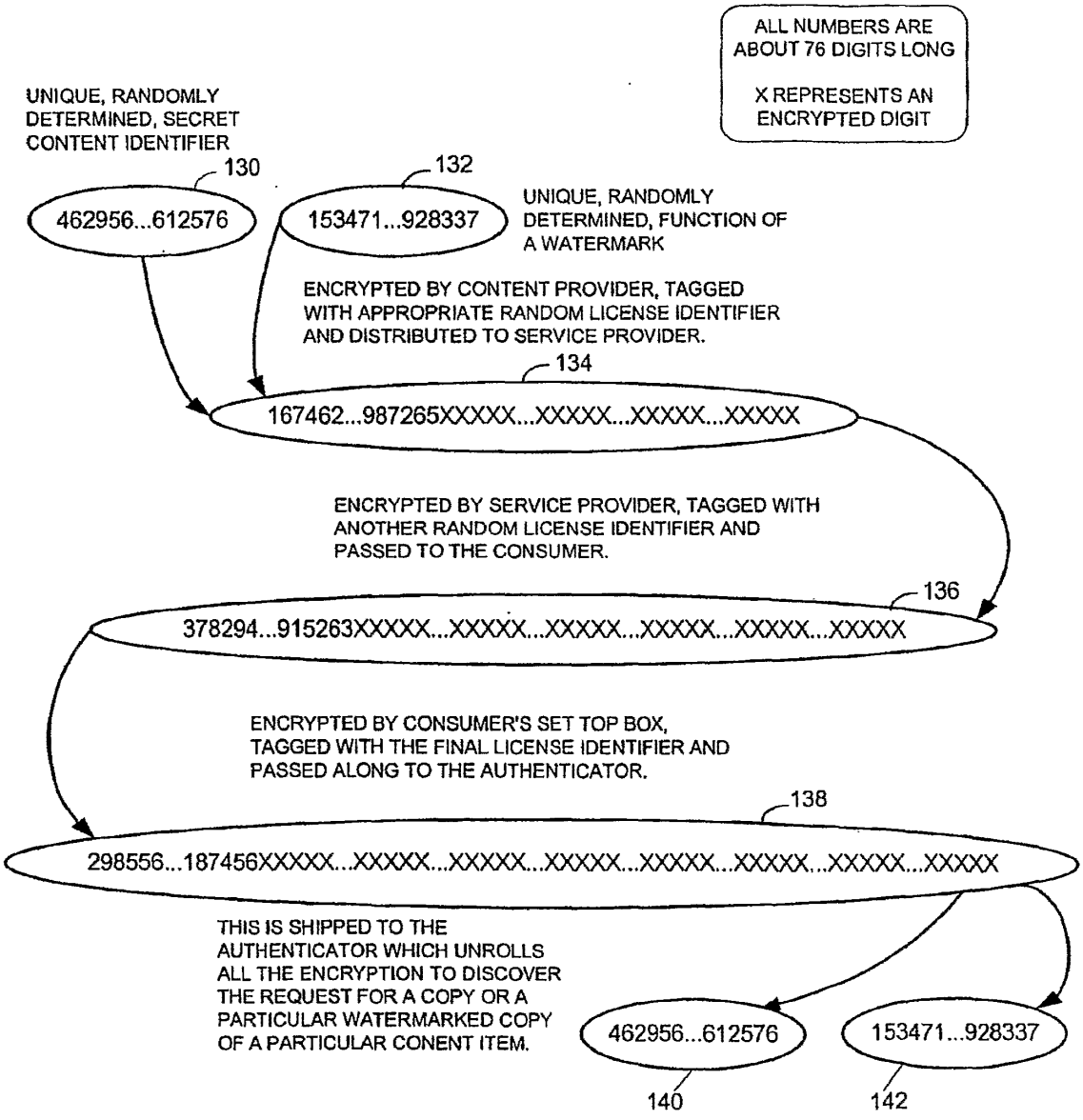
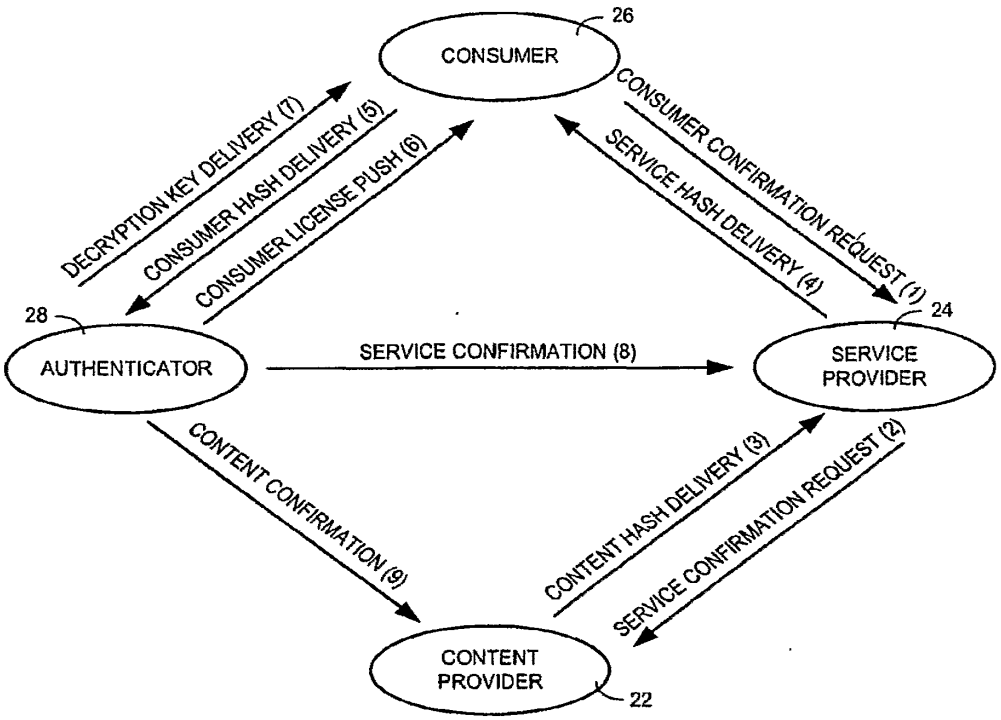

ALL NUMBERS ARE ABOUT 76 DIGITS LONG
X REPRESENTS AN ENCRYPTED DIGIT
UNIQUE, RANDOMLY DETERMINED, SECRET CONTENT IDENTIFIER
130
462956...612576
132
153471...928337
UNIQUE, RANDOMLY DETERMINED, FUNCTION OF A WATERMARK
ENCRYPTED BY CONTENT PROVIDER, TAGGED WITH APPROPRIATE RANDOM LICENSE IDENTIFIER AND DISTRIBUTED TO SERVICE PROVIDER.
134
167462...987265XXXXX...XXXXX...XXXXX...XXXXX
ENCRYPTED BY SERVICE PROVIDER, TAGGED WITH ANOTHER RANDOM LICENSE IDENTIFIER AND PASSED TO THE CONSUMER.
136
378294...915263XXXXX...XXXXX...XXXXX...XXXXX...XXXXX...XXXXX
ENCRYPTED BY CONSUMER'S SET TOP BOX, TAGGED WITH THE FINAL LICENSE IDENTIFIER AND PASSED ALONG TO THE AUTHENTICATOR.
138
298556...187456XXXXX...XXXXX...XXXXX...XXXXX...XXXXX...XXXXX...XXXXX...XXXXX
THIS IS SHIPPED TO THE AUTHENTICATOR WHICH UNROLLS ALL THE ENCRYPTION TO DISCOVER THE REQUEST FOR A COPY OR A PARTICULAR WATERMARKED COPY OF A PARTICULAR CONENT ITEM.
462956...612576
153471...928337
140
142

FIG. 14



(1) CONSUMER MAKES A REQUEST FOR A MOVIE OR CONTENT.
(2) THE DISTRIBUTOR OR SERVICE PROVIDER RELAYS REQUEST TO THE PRODUCER OR CONTENT PROVIDER.
(3) THE PRODUCER OR CONTENT PROVIDER ENCRYPTS A FUNCTION OF THE MOVIE'S TITLE (OR SOME OTHER IDENTIFYING INFORMATION) AND A FUNCTION OF A UNIQUE WATERMARK USING ONE OF THE LICENSES ACQUIRED EARLIER.
SENDS:

ENCRYPTED BY PRODUCER KEY

PRODUCER LIC. #	IDENTIFYING MOVIE INFORMATION	WATERMARK

(4) DISTRIBUTOR ENCRYPTS MESSAGE FROM PRODUCER WITH ONE OF ITS LICENSES ACQUIRED EARLIER.
SENDS:

ENCRYPTED BY DISTRIBUTOR KEY
ENCRYPTED BY PRODUCER KEY

DISTRIBUTOR LIC. #	PRODUCER LIC. #	IDENTIFYING MOVIE INFORMATION	WATERMARK

(5) CONSUMER ENCRYPTS MESSAGE FROM PRODUCER WITH HIS OWN LICENSE.
SENDS:

ENCRYPTED BY CONSUMER KEY
ENCRYPTED BY DISTRIBUTOR KEY
ENCRYPTED BY PRODUCER KEY

CONSUMER LIC. #	DISTRIBUTOR LIC. #	PRODUCER LIC. #	IDENTIFYING MOVIE INFORMATION	WATERMARK

(6) NEW LICENSE IS SENT TO CONSUMER BY THE KEY MASTER OR AUTHENTICATOR.
(7) PRODUCER IS INFORMED OF THE USE OF HIS LICENSE IF HE SO DESIRES, LINKING CUSTOMER TO WATERMARKED CONTENT ITEM. ALTERNATIVELY, LICENSE USAGE MAY BE TRACKED BY THE KEY MASTER.
(8) MATCHING DECRYPTION KEY FOR WATERMARKED MOVIE OR CONTENT IS DELIVERED TO THE CONSUMER.
(9) DISTRIBUTOR IS INFORMED OF THE USE OF HIS LICENSE (E.G., FOR BILLING PURPOSES) AND IS TOLD TO BEGIN SENDING MATCHING ENCRYPTED WATERMARKED MOVIE OR CONTENT.

FIG. 15

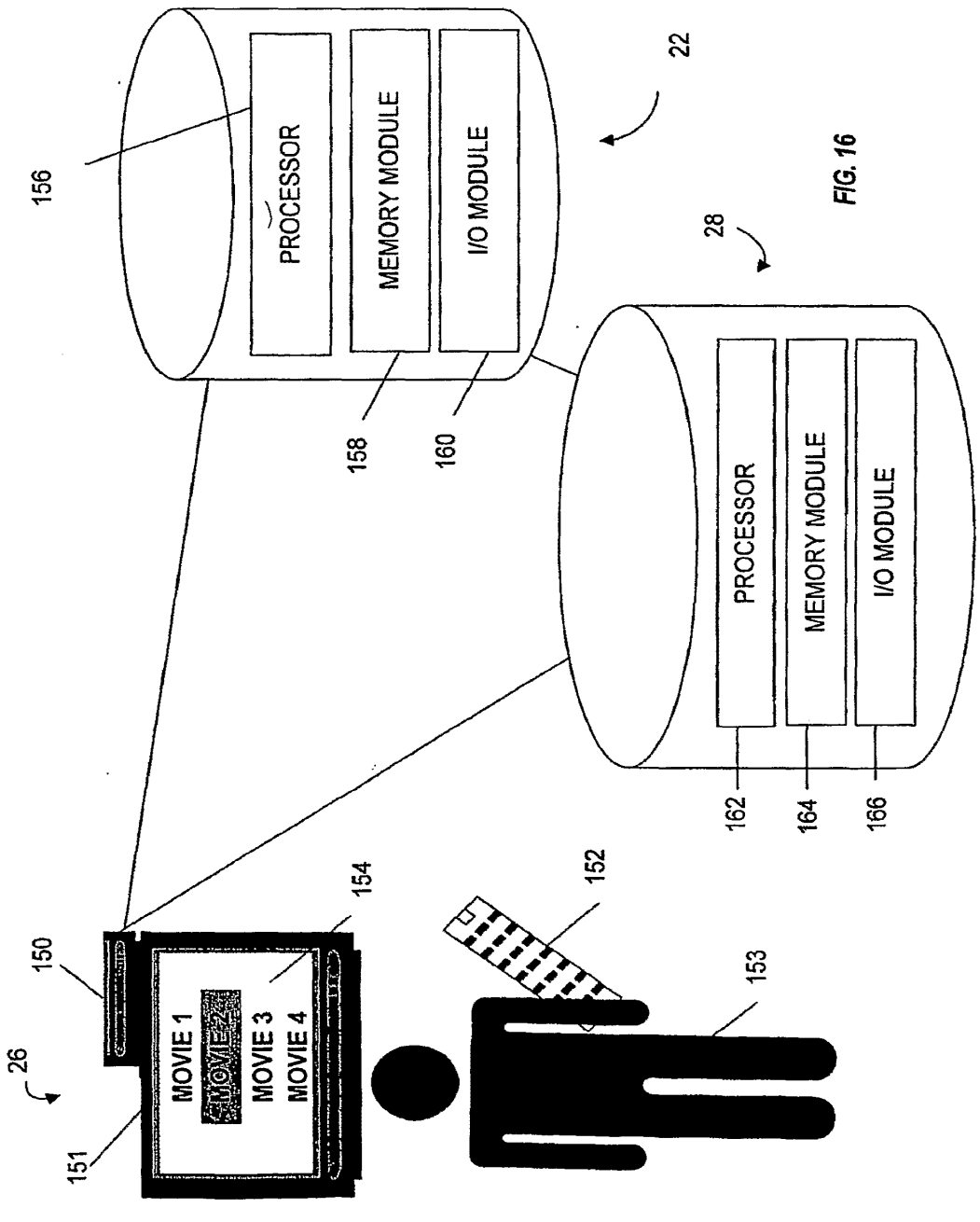


FIG. 16

WAVELYNX EX. 1003, Pg. 882
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

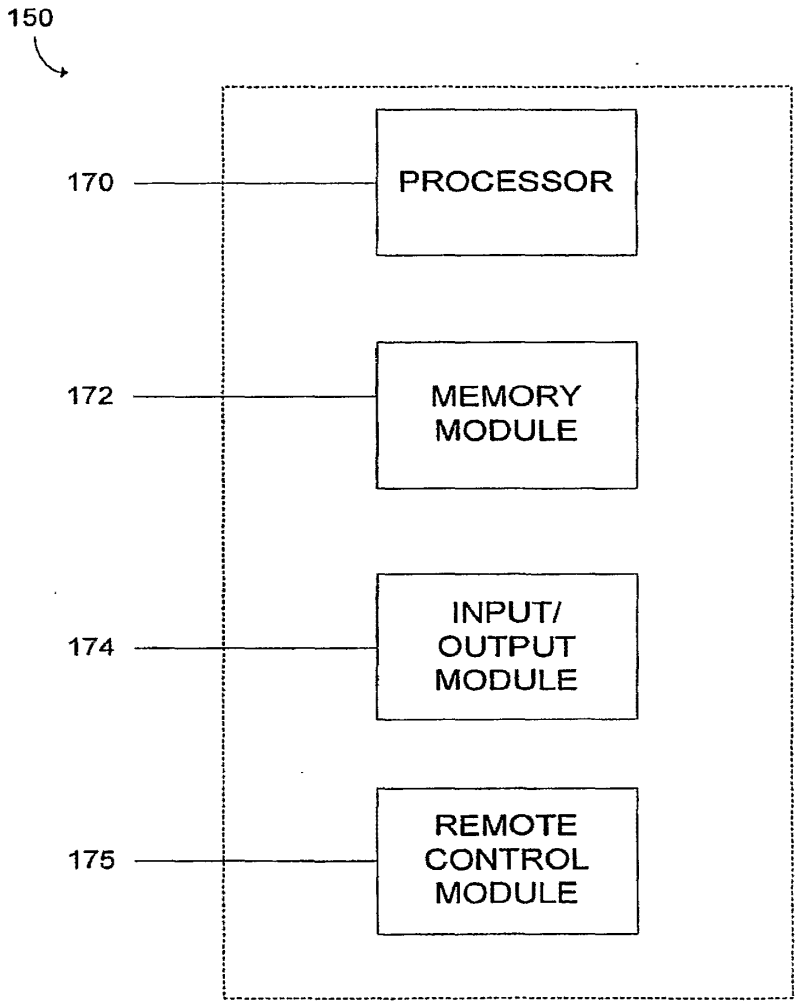

150
170
PROCESSOR
172
MEMORY MODULE
174
INPUT/ OUTPUT MODULE
175
REMOTE CONTROL MODULE

FIG. 17

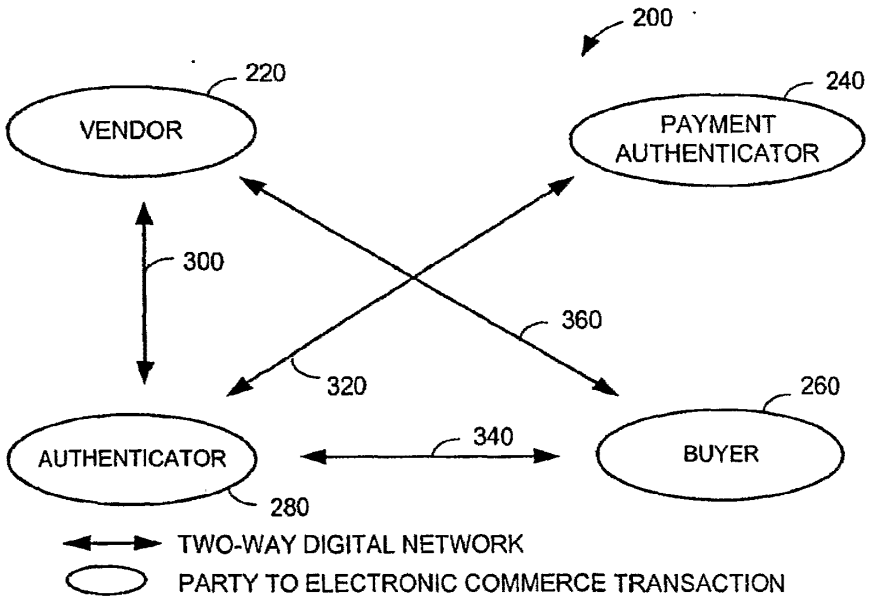

200
220
VENDOR
240
PAYMENT AUTHENTICATOR
300
360
320
260
AUTHENTICATOR
340
BUYER
280
TWO-WAY DIGITAL NETWORK
PARTY TO ELECTRONIC COMMERCE TRANSACTION

FIG. 18

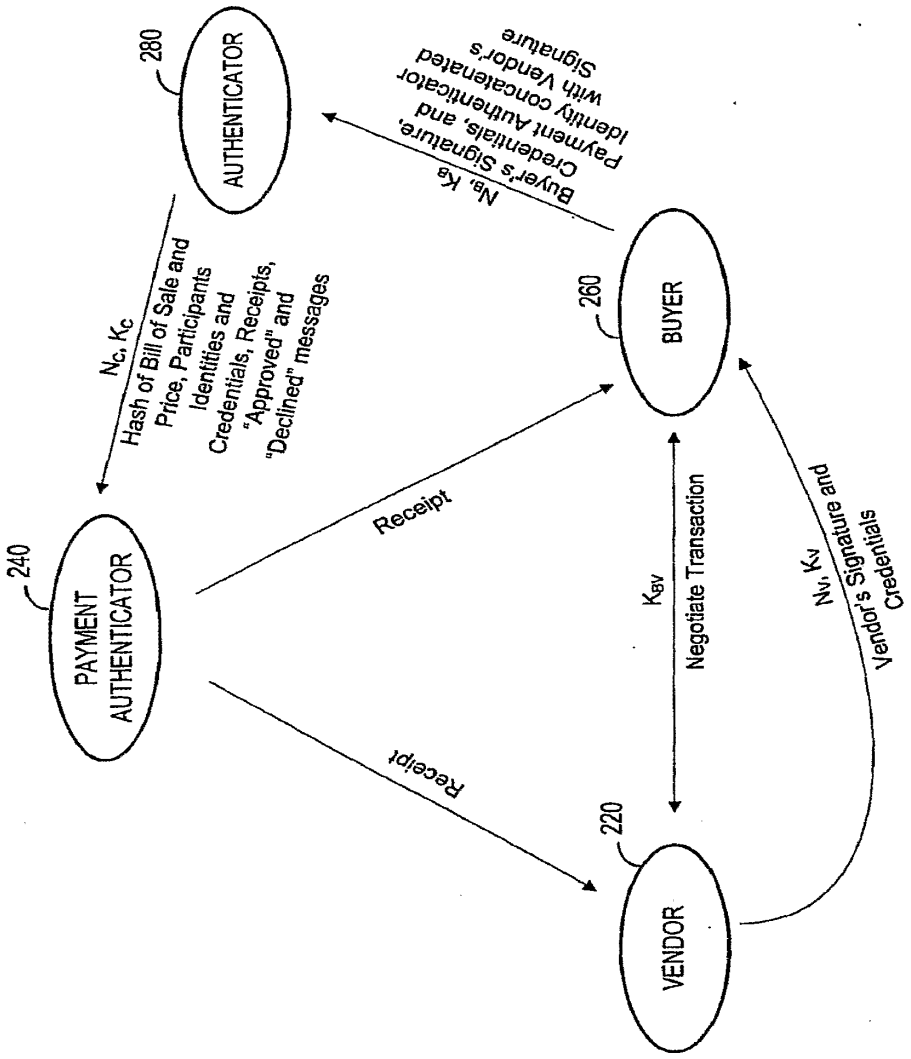

AUTHENTICATOR
280
BUYER
260
PAYMENT AUTHENTICATOR
240
VENDOR
220
Buyer's Signature, Credentials, Payment Authentication, and Identity concatenated with Vendor's Signature
Na, Ka
Nc, Kc
Hash of Bill of Sale and Price, Participants Identities and Credentials, Receipts, "Approved" and "Declined" messages
Receipt
Receipt
KBV
Negotiate Transaction
Nv, Kv
Vendor's Signature and Credentials

FIG. 19

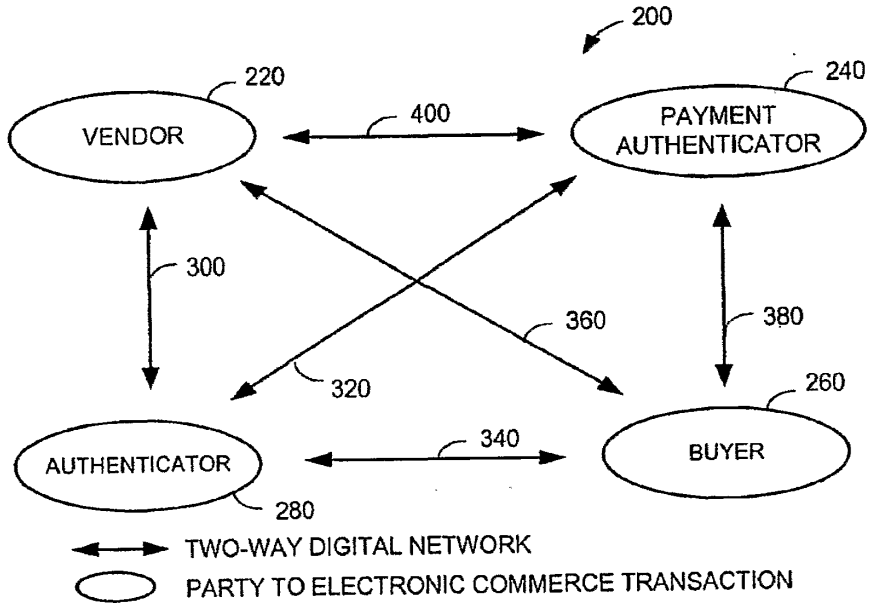

200
220
VENDOR
400
240
PAYMENT AUTHENTICATOR
300
360
380
320
260
AUTHENTICATOR
340
BUYER
280
TWO-WAY DIGITAL NETWORK
PARTY TO ELECTRONIC COMMERCE TRANSACTION

FIG. 20

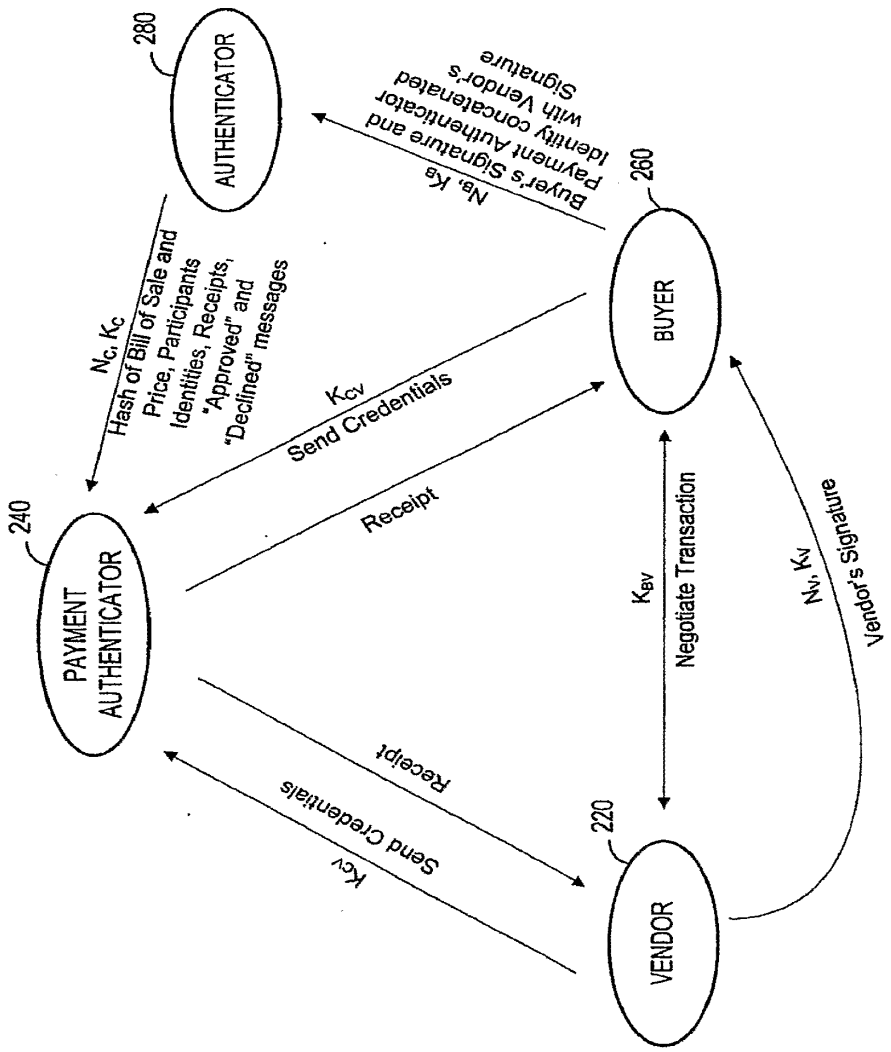

280
AUTHENTICATOR
260
BUYER
240
PAYMENT AUTHENTICATOR
220
VENDOR
Buyer's Signature and Payment Authenticator identity concatenated with Vendor's Signature
Na, Ka
Nc, Kc
Hash of Bill of Sale and Price, Participants Identities, Receipts, "Approved" and "Declined" messages
Kcv
Send Credentials
Receipt
Kav
Negotiate Transaction
Nv, Kv
Vendor's Signature
Receipt
Send Credentials
Kcv

FIG. 21

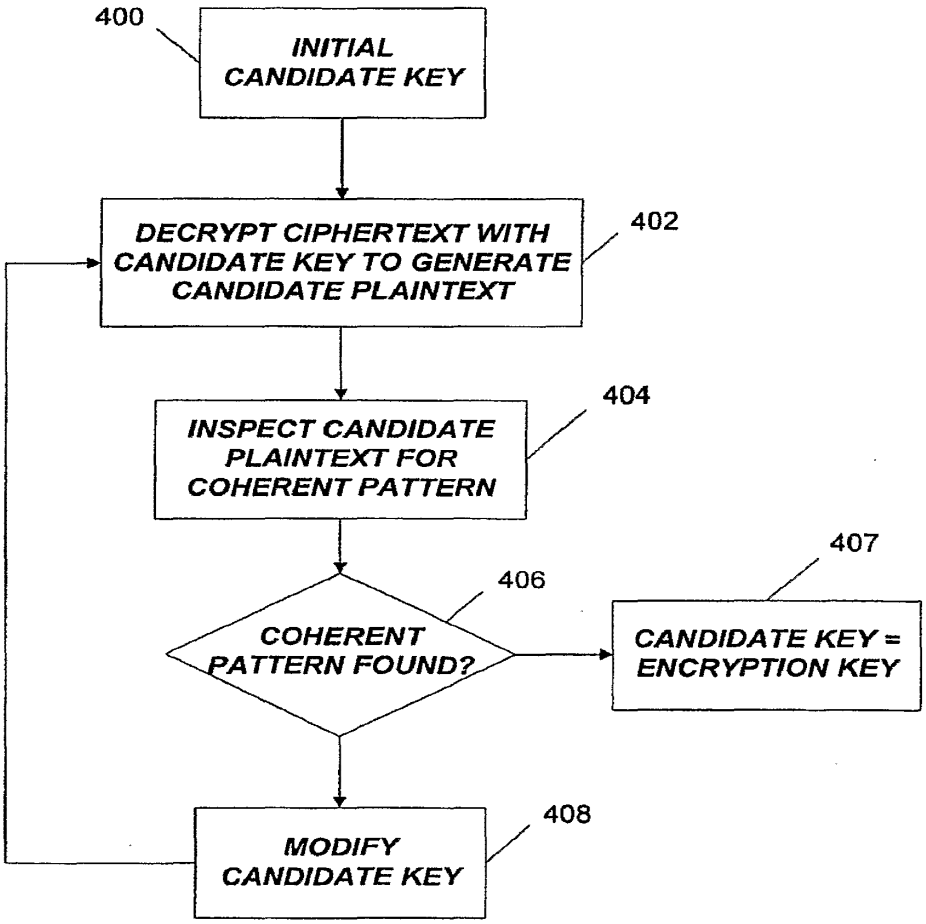

400
INITIAL CANDIDATE KEY
DECRYPT CIPHERTEXT WITH CANDIDATE KEY TO GENERATE CANDIDATE PLAINTEXT
402
INSPECT CANDIDATE PLAINTEXT FOR COHERENT PATTERN
404
406
COHERENT PATTERN FOUND?
407
CANDIDATE KEY = ENCRYPTION KEY
MODIFY CANDIDATE KEY
408

FIG. 22

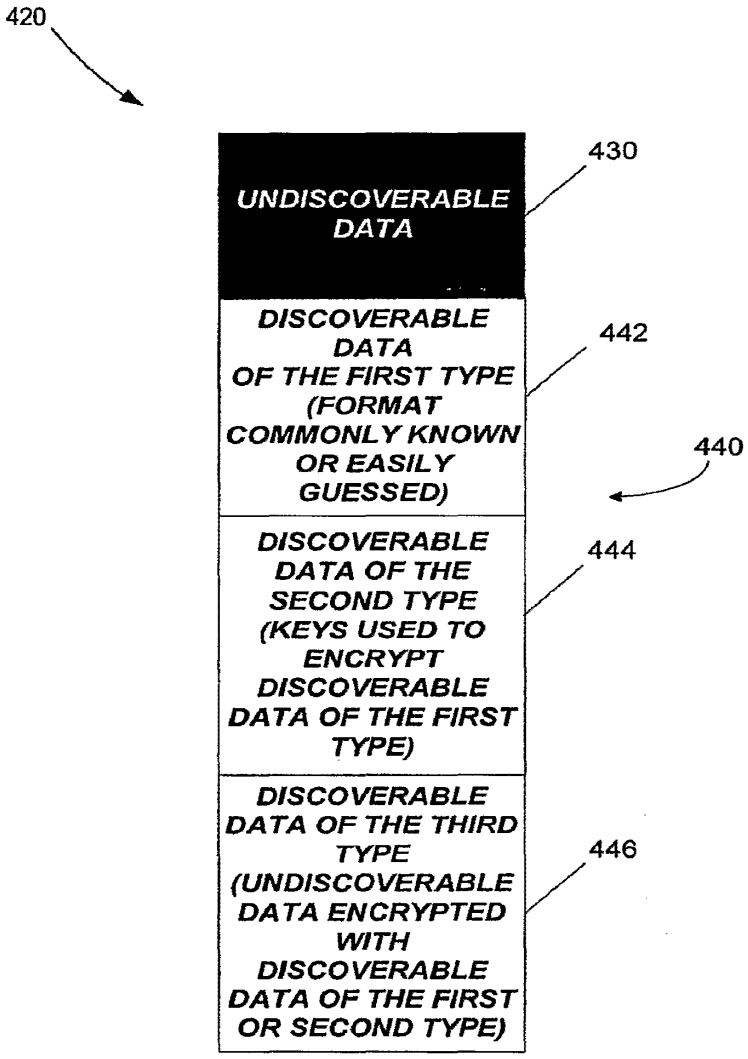

420
UNDISCOVERABLE DATA
430
DISCOVERABLE DATA OF THE FIRST TYPE (FORMAT COMMONLY KNOWN OR EASILY GUESSED)
442
440
DISCOVERABLE DATA OF THE SECOND TYPE (KEYS USED TO ENCRYPT DISCOVERABLE DATA OF THE FIRST TYPE)
444
DISCOVERABLE DATA OF THE THIRD TYPE (UNDISCOVERABLE DATA ENCRYPTED WITH DISCOVERABLE DATA OF THE FIRST OR SECOND TYPE)
446

FIG. 23

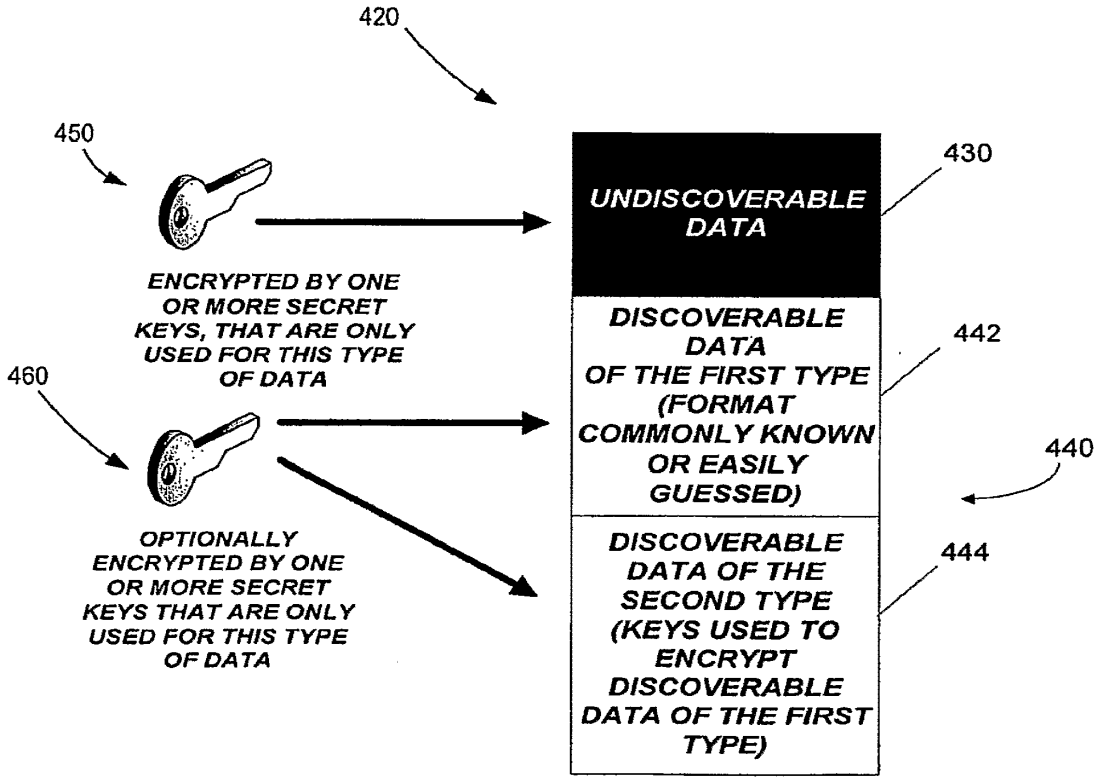

420
450
430
UNDISCOVERABLE DATA
ENCRYPTED BY ONE OR MORE SECRET KEYS, THAT ARE ONLY USED FOR THIS TYPE OF DATA
DISCOVERABLE DATA OF THE FIRST TYPE (FORMAT COMMONLY KNOWN OR EASILY GUESSED)
442
460
440
OPTIONALLY ENCRYPTED BY ONE OR MORE SECRET KEYS THAT ARE ONLY USED FOR THIS TYPE OF DATA
DISCOVERABLE DATA OF THE SECOND TYPE (KEYS USED TO ENCRYPT DISCOVERABLE DATA OF THE FIRST TYPE)
444

FIG. 24

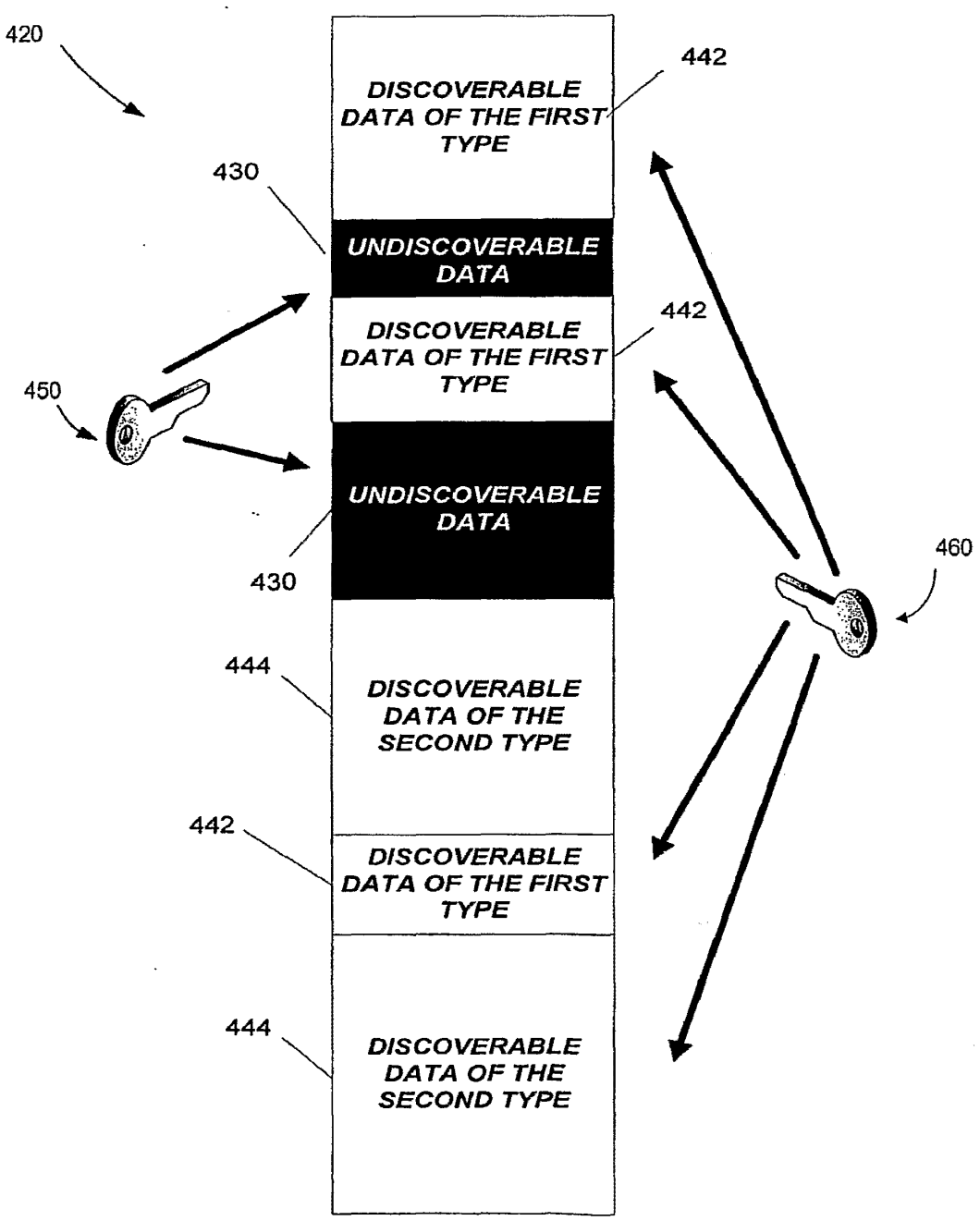

420
442
DISCOVERABLE DATA OF THE FIRST TYPE
430
UNDISCOVERABLE DATA
442
DISCOVERABLE DATA OF THE FIRST TYPE
450
UNDISCOVERABLE DATA
430
460
444
DISCOVERABLE DATA OF THE SECOND TYPE
442
DISCOVERABLE DATA OF THE FIRST TYPE
444
DISCOVERABLE DATA OF THE SECOND TYPE

FIG. 25

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2008 年1 月3 日 (03.01.2008)

PCT

(10) 国際公開番号
WO 2008/001918 A1

(51) 国際特許分類:
G09C 1/00 (2006.01) *H04L 9/26* (2006.01)
G06F 7/58 (2006.01)

(21) 国際出願番号: PCT/JP2007/063182

(22) 国際出願日: 2007 年6 月30 日 (30.06.2007)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2006-181941 2006 年6 月30 日 (30.06.2006) JP

(71) 出願人 および
(72) 発明者: 由井 清人 (YUI, Kiyoto) [JP/JP]; 〒1080074 東京都港区高輪１丁目２番１６号 Tokyo (JP).

(72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 住澤 大介 (SUMISAWA, Daisuke) [JP/JP]; 〒1080074 東京都港区高輪１丁目２番１６号 Tokyo (JP). 川崎 昭 (KAWASAKI, Akira) [JP/JP]; 〒1080074 東京都港区高輪１丁目２番１６号 Tokyo (JP). 佐々木 建之 (SASAKI, Takeshi) [JP/JP]; 〒1080074 東京都港区高輪１丁目２番１６号 Tokyo (JP).

(74) 代理人: 塩野谷英城 (SHIONOYA, Hideki); 〒1140023 東京都北区滝野川６－８３－３－３０１ 塩野谷国際特許事務所 Tokyo (JP).

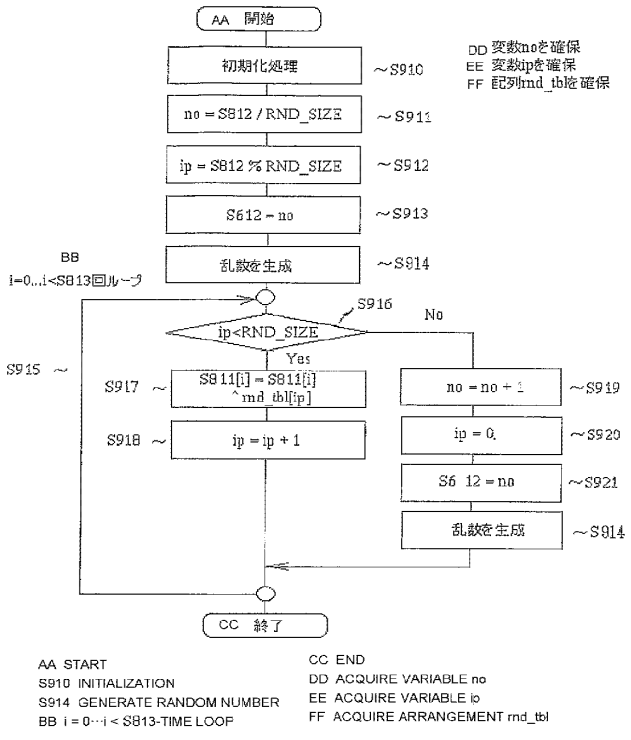
(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM,

/続葉有/

WO 2008/001918 A1

(54) Title: RANDOM NUMBER GENERATION DEVICE, ENCRYPTION/DECRYPTION DEVICE USING THE SAME, ITS PROGRAM, AND PROGRAM RECORDING MEDIUM

(54) 発明の名称: 乱数表発生装置及びこれを利用した暗号化／復号化装置並びにそのプログラム及びプログラム記録媒体



(57) Abstract: [PROBLEMS] The necessity of sophisticated encryption and decryption which can easily be used for a memory disc or the like is increasing. The present invention generates an effective pseudo-random number which can be used in a small-scale computer system and a mobile terminal. A pseudo-random number generation device has n encryption keys and uses a combination of the encryption keys so as to extract the maximum performance of the pseudo-random number and execute encryption and decryption for a rugged memory disc. [MEANS FOR SOLVING PROBLEMS] Since a random number sequence generated by a reproducible pseudo-random number has an N-1 cycle, a mechanism has been developed for using the generated random number sequence in this cycle without entering the next cycle. Moreover, the random number generation device includes a mechanism for extracting a random number sequence without overlap in the random number sequence N-1 generated by the pseudo-random number so as to maximize the disorder of the random number and extract the maximum performance of the random number generation device. The random number generation device is embedded to encrypt/decrypt data so as to protect data in a memory disc or the like.

/続葉有/

WAVELYNX EX. 1003, Pg. 892
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(57) 要約: 【課題】 メモリディスク等に簡単に利用できる高性能の暗号化復号化の必要が増大している。本発明は、小規模コンピュータシステム、携帯端末にて使用できる効率の良い疑似乱数を生成するものである。疑似乱数発生装置に対して、ｎ個の暗号鍵を備え、この暗号鍵の組み合わせで疑似乱数の性能を最大限引き出せるようにし、堅牢なメモリディスク用の暗号化復号化を行うものである。【解決手段】 再現性のある疑似乱数の生成する乱数列はN-1の周期を備える事から、生成する乱数列をこの周期の中で使用するようにし、次の周期にまたがらない機構を開発した。また、疑似乱数が生成する乱数列N-1の中で重複無く乱数列を取り出す機構を備え、乱数の無秩序性が最大になるようにして、乱数発生装置の性能を最大に引き出すようにしたものである。および、この乱数発生装置を組み込みデータの暗号化復号化を行い、メモリディスク等のデータを保護するようにした。

明 細 書

乱数表発生装置及びこれを利用した暗号化／復号化装置並びにそのプログラム及びプログラム記録媒体

技術分野

[0001] 本発明は、乱数表発生装置及びこれを利用した暗号化／復号化装置に関するものである。特に、少リソースな、組込機器、家電品、携帯機器、小型情報端末、小型ゲーム端末機器に好適な乱数表発生装置及び暗号化／復号化装置に関するものである。

背景技術

[0002] 一般的な乱数生成の技術として、タイマーを備え、電源起動からの時間情報と、前回の最終情報とに基づいてシードを生成し、このシードによって乱数生成を行うことで、外部からの解析にも耐える手段が開示されている(特許文献1参照)。

[0003] また、直近の乱数のやり取りを覚えておき、使った乱数値を使わないようにすることで、認証の安全性を高めようとする方法が開示されている(特許文献2参照)。

[0004] また、少ないリソースで乱数生成を実現しようとするもので、自装置内で乱数生成する手段が開示されている(特許文献3参照)。

[0005] また、非特許文献1には、メルセンヌ・ツイスタが紹介されている。メルセンヌ素数を利用したメルセンヌ・ツイスタ(Mersenne twister)は、1997年(論文発表は1998年1月)に松本眞と西村拓士によって開発された擬似乱数生成器である。周期が2の19937乗−1つまり623ワード(19937ビット、2492バイト)の表現力で、623次元超立方体の中に 均等に分布することが証明されている。

[0006] 32ビット版のメルセンヌ・ツイスタは、内部構造として1ワード増やし、624ワード(19968ビット、2496バイト)の内部変数を持つ。この内部変数の組み合わせで624ワード(19968ビット、2496バイト)の乱数表を生成する。この乱数表は、内部の624ワード(19968ビット、 2496バイト)の組み合わせだけ異なる乱数表が存在する。

[0007] 以上より、均等に分布することを証明された最低でも624ワード(19968ビット、 2496バイト)の乱数表が2の19937乗−1の個数が存在すると説明できる(非特許文献1参照

WAVELYNX EX. 1003, Pg. 894
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

)。

[0008] また、非特許文献2は、外務省が公開している暗号の紹介ページである。このページでは、有限乱数式および無限乱数式暗号を判りやすく紹介している。ここでは無限乱数式暗号は論理的に解読できないという事実を述べている(非特許文献2参照)。

[0009] データの暗号化においては、有限乱数式と無限乱数式が存在する。有限乱数式とは同じ乱数表を繰り返し暗号化に使用する方式である。効率が良い反面一度乱数表を解読されると後は簡単に解読ができる。無限乱数式は一度使用した乱数表は2度と使わないという方式である。無限乱数式は同じ乱数を2度使わず1度限りの使用においては絶対に解読できないことが情報理論により証明されている暗号である。

特許文献1:特開2000－242470号公報

特許文献2:特開2001－177519号公報

特許文献3:特開平10－247140号公報

非特許文献1:M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator", ACM Trans. on Modeling and Computer Simulation Vol. 8, No. 1, January pp.3-30 (1998)http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/mt.html

非特許文献2:外務省大臣官房情報通信課 非常勤講師 奥村定夫「外交と暗号」http: //www.mofa.go.jp/mofaj/annai/shocho/e_seifu/toukou2004.html

発明の開示

発明が解決しようとする課題

[0010] 当発明は、特定個人、特定のコンピュータシステム、特定の機材の接続等の条件を満たした時だけデータを正しく復号化できるようにして、データ盗難・データ改ざん・ウイルスやスパイウェア・ピアツーピアによるアプリケーションからなる被害を最小にすることを課題とする。

課題を解決するための手段

[0011] 当発明は、データに対してn個の暗号鍵を設定し、特定個人、特定のコンピュータシステム、特定の機材の接続等の条件を満たした時に情報を復号化できるようにするものである。また、本発明は、疑似乱数の発生装置をきめ細かく制御し重複する乱数

の発生を防ぎ、毎回異なる乱数を生成するようにして無限乱数式の強度を取り入れたものである。

[0012] 本発明の乱数表発生装置は、コンピュータ処理による乱数表発生装置であり、以下の要素を備える。即ち、使用制限数n（nは正数）以上の要素数からなる乱数表を生成する手段と、生成した乱数表を記憶する領域と、使用制限数nを記憶する領域と、乱数表の要素を使用制限数nまで使用したとき当該乱数表のn＋1以降の要素を使用せずに別の乱数表を生成させる手段とを備える。これは、後述する実施例のうち図9のS916に対応する思想である。

[0013] また、本発明の乱数表発生装置は、起源の異なる複数の暗号鍵を取得する手段と、前記起源の異なる複数の暗号鍵を記憶する領域と、前記起源の異なる複数の暗号鍵をシードとして一つの乱数表を生成する手段と、前記乱数表を記憶する領域とを備える。

[0014] 上記乱数表発生装置は、メルセンヌ素数を使用した疑似乱数発生方式又はメルセンヌ・ツイスタによる疑似乱数発生方式により前記乱数表を生成することができる。

[0015] また、上記暗号鍵の一つとして時間情報を含むことができる。この場合、乱数表と時間情報とを演算した結果を乱数表とする手段を備えることができる。

[0016] また、暗号鍵の一つとして、乱数表発生装置としてのコンピュータシステムに接続している機器の固有情報を含むことができる。

[0017] また、暗号鍵の一つとして、乱数表発生装置としてのコンピュータシステム自体の固有情報を含むことができる。この場合、コンピュータシステムの固有情報として、当該コンピュータシステムが内蔵する記憶装置のシリアル番号を取得することができる。

[0018] 本発明において、コンピュータシステムには、組込機器、家電品、携帯機器、小型情報端末又は小型ゲーム端末機器等も含まれる。

[0019] また、暗号鍵の一つとして、ユーザの生体情報を含むことができる。

[0020] また、暗号鍵の一つとして、入力装置から入力された又は自動的に設定したキーワードを含むことができる。

[0021] また、暗号鍵の一つとして、連続番号等のユニークコードを、生成する乱数表毎に与える手段を備えることができる。

[0022] また、乱数表を複数生成し当該複数の乱数表の要素を組み合わせることにより新たな乱数系列を生成する手段を備えることができる。

[0023] 次に、本発明の暗号化／復号化装置は、上述した乱数表発生装置を備えると共に、以下の要素を備える。即ち、暗号化又は復号化すべき情報を記憶する領域と、乱数表を用いて暗号化又は復号化すべき情報を暗号化又は復号化する手段と、暗号化又は復号化した情報を記憶する領域とを備える。このうち、暗号化又は復号化する手段は、暗号化又は復号化すべき情報毎に、乱数表発生装置から異なるシードによる乱数表を個別に生成させ、最近に生成させた乱数表を用いて暗号化又は復号化を行う。

[0024] この際、暗号鍵に設定する時間情報は、暗号化又は復号化すべき情報に関連する時間情報とすることができる。

[0025] 例えば、暗号化した情報をコンピュータの基本制御ソフトが持つファイル管理システムによりファイルとして記憶装置に保存させる手段と、当該暗号化に用いた時間情報を当該ファイルの生成時間もしくは更新時間として当該ファイルに保存させる手段とを備えることができる。

[0026] また、暗号鍵に設定するユニークコードとしてセクタ及びブロック等の記憶装置の記録単位を指標する値を与えることができる。この場合、ある記録単位に記録されている情報を暗号化又は複合化する際に用いる乱数表は、その記録単位を指標する値をユニークコードに与えて生成する手段を備えることができる。

発明を実施するための最良の形態

[0027] 以下、本発明の実施例について図面を用いて説明する。

[0028] ［実施例1］

[0029] 本実施例は、暗号化と復号を行うプログラムおよび装置に関する。図18に本発明の第1実施例のコンピュータシステムの電子機器としての全体構成図を示す。

[0030] 情報処理装置（ホスト機器）は、計算機の全体制御および各種プログラムを実行する中央処理装置S1811（以下CPU）と、各種プログラムやデータがロードされるメモリ装 置S1813と、ROMBIOS－S1812と、アダプタS1815，S1816とを備えている。これらは、システムバスS1814に接続されている。

[0031] このうち、アダプタS1815には、HDD－S1830のホストインタフェースS1820が接続されている。また、アダプタS1816には、ストレージデバイスを装着可能なリーダライタS1819が接続されている。本実施例では、ストレージデバイス（記憶装置）として、USB装置S1840を装着できるようになっている。USB装置S1840は、ストレージデバイスとしての機能を備える。本実施例では、USB装置S1840が、メモリディスク装置であるとして説明するが、本実施例の効果を奏する範囲内であれば任意の装置でよい。

[0032] ROMBIOS－S1812には、HDD－S1830あるいはストレージデバイスなどのシステムブート可能デバイスから一つのデバイスを選択する手段と、当該デバイスからシステムブートローダをメモリ装置S1813に読出す手段と、読み出したシステムブートローダに分岐する手段（いわゆるシステムブート機能）とを備える。システムブートにおいて、HDD－S1830あるいは任意のストレージデバイスから読み出されたOSプログラムが、メモリ装置S1813に格納され、CPU－S1811により実行される。

[0033] 次に、上記コンピュータシステムにおいて発生させる乱数の管理方式を説明する。

[0034] 本実施例では疑似乱数発生装置としてメルセンヌ・ツイスタを使用する。疑似乱数は、その欠点として周期を持つことが挙げられる。メルセンヌ・ツイスタは、一定周期で同じ乱数を生成する。メルセンヌ・ツイスタは、2の19937乗−1。十進法の場合は約10の6000乗の周期を持つ。つまり、約10の6000乗の個数の乱数を繰り返し生成する。図1の上図は、その一周期を直線的に表したものである。横軸はデータ長を示し、区切られた短冊は、その個々が1個の乱数を表す単位データ長である。

[0035] いま、図1の下図のように、2の19937乗−1を要素Nとする巨大な乱数表が存在する。S112は、この要素数nの重なり合わない事を特徴とする小テーブルである。本実施例の暗号化は、この小テーブルS112を使用して行う。

[0036] 小テーブルS112は、一度使ったならば2度と使わない。これにより疑似乱数の周期の中で無限乱数式の暗号化と復号化を行う。ここで、小テーブルS112は、隙間無く連続して並んでいることが効率上このましいが、空隙があってもよい。

[0037] メルセンヌ・ツイスタは、その内部仕様として内部で2496バイト（624ワード）のテーブルを備え、一つのシードを与えた時発生させる乱数を2496バイト以下としたとき、当該テーブルを小テーブルS112として使用できる。この場合、一つのシードにおいて生成

する乱数を2496バイト以下とし、同じシードを二度と使わないという条件において重複の起きない乱数を生成し管理できる。

[0038] 即ち、下記簡素計算式を1テーブルと見なし、上記理論を確立するものとする。

(a ≒ s + a' = a'') × n = ∞

a = 10の6000乗

a'= 2492バイト

a'' = 623ワード数

s = 1つのシード

n = テーブル数

∞ = 無限数値

[0039] 続いて、本実施例の暗号化の方法を、図2に基づいて説明する。S210は、暗号化したい平文である。また、S211は、乱数表である。乱数表S211は、一つ一つ異なるシードで生成した小テーブルS112の集合体である。そして、平文S210の各要素と、小テーブルS211の各要素とを1対1、つまりバイト単位で排他的論理和をとった結果がS212である。

[0040] このように、S210とS211の間で演算を行うことにより暗号化ができる。演算の種類としては排他的論理和の他に加算ないし減算をしてもよいが、処理が簡単と言う理由で排他的論理和を本実施例では採用する。排他的論理和を使用した場合、復号も同一の排他的論理和で足り、コンピュータ処理に向いているからである。即ち、平文S210を復号したい場合、暗号化されたデータS212と、暗号化した時に使用した乱数表との排他的論理和を取ることによって、平文S210を復号することが出来る。

[0041] 続いて、図3は、本実施例のシステムの物理的全体像を示す。S310は、図18に示したコンピュータのUSBコネクタである。また、S311は、指紋認証装置を内蔵したUSBメモリ装置(S1840)である。本実施例は、指紋認証装置を内蔵したUSBメモリ装置により試作を行ったが、本発明は、この指紋認証装置を必須とするものではない。指紋認証装置は、実施例の一つであることを明示する。

[0042] 続いて、図4に、本実施例で採用したUSBメモリ装置の使い方を説明する。なお、USBの基本制御ソフトには、米マイクロソフト社のものを使用している。

[0043] 本USBメモリ装置をUSBコネクタS310に接続すると、コンピュータシステムは、表示装置の画面上にS410の画面を表示する。画面上には、CD-ROM装置S411と、リムーバブルディスクS412とが出現する。

[0044] 続いて、コンピュータシステムは、S416の指紋認証を促す画面を表示装置に表示する。ユーザが入力装置から認証を指示すると、コンピュータシステムは、S417の指紋認証画面を表示装置に表示する。ユーザが、USBメモリ装置に内蔵された指紋センサーに指をスライドさせると、コンピュータシステムは、ユーザの指紋情報を読み取り、指紋認証を実行する。指紋認証が通ると、コンピュータシステムは、S410の画面をS413の画面に切り替える。

[0045] 画面の切替後に表示されるS414とS415は、指紋認証した時に始めて出現する保護されたディスク領域である。なお、この一連の処理を行うためには、USBメモリ装置付属のツールでユーザの指紋を予めコンピュータシステムに登録しておく必要がある。

[0046] ここで、この指紋認証およびUSBメモリ装置の使い方を述べるのは、本発明の理解を促進するためであり、本発明とは直接関係ないことを明示しておく。

[0047] 図5は、上述したコンピュータシステムにおいて乱数を生成する際のシードとなるn個の暗号鍵を示す。まず、本実施例では、静的に定まる暗号鍵を持つ。これらは、本装置が作動する上で必須の情報である。

[0048] 図5において、S510は、暗号鍵生成用のメモリ領域である。本実施例では624ワード(2496バイト)のサイズを確保する。このサイズは、本実施例が採用しているメルセンヌ・ツイスタ公開プログラム(2002年版(mt19937ar.c))に設定できるシードの最大桁数に合わせている。ただし、624ワードのサイズを確保しているが全てを使う必要はない。

[0049] ここで、メモリ領域S510には、誤動作を防ぐため、このメモリ領域を確保した時に一度全てゼロを格納するものとする。その上で内部の各要素を格納するものとする。また、メモリ領域S510に格納する暗号鍵のうち静的に定まる各要素S511～S514は、全て使用するか否かは任意とする。使用する項目には有値を格納し、使わない項目の値はゼロとする。ここで、静的に定まる各要素とは、後述する時間情報および連続番号(動的暗号鍵S610)以外の要素のことである。

[0050] 静的に定まる各要素のうち、キーワードS511は、本実施例が作動するコンピュータシステムに記憶する値であり、任意の文字列である。本実施例では、128バイト以内のアスキー文字列とし、この 文字列はヌルコードで終端するものとする。このキーワードS511は、S515に示すような入力画面を介して初期値を設定し、コンピュータが不揮発性の記憶装置に保存する。

[0051] また、コンピュータ固有情報S512は、本実施例が動作しているコンピュータシステムもしくは端末のユニークコードを読み取り格納するものである。本実施例において、コンピュータ固有情報を格納するメモリ領域は、4バイトの領域とする。例えば、米マイクロソフト社の基本制御ソフトでは、ディスク装置を使用するにあたり、ディスク装置を初期化する毎に、ボリュームシリアル番号と呼称する4バイトのユニークコードを発行する。本実施例では、この4バイトの値を固有情報S512としてメモリ領域S511に書き込むものとする。

[0052] 図5において、S516の画面には、この4バイトのユニークコードが示されている。このS516の画面は、ディスクの一覧情報を採取したものである。このうち、S517の16進コードがこの4バイトコードである。

[0053] 一方、本実施例において、コンピュータシステムのユニークコードのチェックをしたくない場合は、常にこのコンピュータ固有情報S512の領域にゼロを格納しておくものとする。この場合、どのコンピュータシステムにUSBメモリ装置S311を接続してもゼロとなり暗号鍵のチェックが緩くなる。

[0054] 続いて、S513には、USBメモリ装置S311の製造会社番号、製品番号およびシリアル番号を登録するものとする。USB装置は、本来の仕様では製造会社番号、製品番号、シリアル番号を備える。この3つのコードの組み合わせを格納するものとする。通常は、このコードは合計で24バイトである。余裕を持ち64バイトの大きさをS513に割り振る。

[0055] ここで、S513に登録するUSBの製造会社番号、製品番号は、USBメモリ装置に製造時に書き込まれている情報である。これは異なる会社のUSBを接続した時にその製造会社番号、製品番号が格納されないようにするための安全措置である。

[0056] ここでは、USBメモリ装置S311を接続した時にUSBメモリ装置より読み取りS513に格

納するものとする。

[0057] 続いて、S514には、4バイトの生体認証キーを登録する。これは、生体認証の結果出現する図4のS415のボリュームシリアル番号を読み取るものとする。本来生体認証キーを直接読み取るのが好ましいが、指紋認証の場合は管理者を含め複数の人間が指紋を登録する場合が考えられる。この場合、指紋登録を行った人物毎に異なる生体認証データが出現する。しかし、複数の人が一台のコンピュータを利用しファイルを共有する場合、暗号化のキーは共有する必要がある。従い、生体認証キーをそのまま暗号に使用すると共有が崩れ不都合が発生する。ゆえに、このようにS415をキーにする必要がある。

[0058] これより生体認証の結果出現するコードを生体認証コードとして登録する。

[0059] 続いて、S610は、動的暗号鍵領域を示している。この動的暗号鍵領域S610に格納するn個の動的暗号鍵の生成について、図6を参照しつつ説明する。図6は、動的暗号鍵領域S610のメモリマップを示している。

[0060] この図6において、S611は、時間情報領域であって、8バイトのメモリ領域である。ここにはコンピュータシステムの時間情報を格納する。時間情報の仕様は、コンピュータに搭載された基本制御ソフトによるが、通常は4バイトないし8バイトで示すシステム秒数を備え、現在の年月日時分秒を一意に決定できる。よくあるシステム秒数は、1秒単位で西暦1970年1月1日の00:00:00から現在に至るまでの秒数になっている。4バイトの表現力で約130年の幅を確保しているので、2100年までの日時を表現でき都合がよい。

[0061] 時間情報領域S611には、暗号化を行う時には、その時の現在時間を格納する。また、この時間情報は、復号化の際に必要となるため、暗号化したデータを記憶装置に保存する時に、必ず暗号鍵に使用した時間情報も関連付けて保存するものとする。この時間情報の保存処理は、本変換プログラム(暗号化／復号化プログラム)を呼び出す上位プロセスの役割であり、本変換プログラムに属さないことを明記する。また上述のとおり、この時間情報がないと暗号化したデータを復号できない。ゆえに暗号化と復号化をセットで行う上で、上位のプログラムないし装置は必ず時間情報を保存する必要があることを明記する。

[0062] 時間情報領域S611には、復号化を行う時には、暗号化が行われた時の時間を格納する。時間情報はかならず上位プログラムにより暗号化したデータ一つずつに存在している。その時間情報をここに登録する。

[0063] 続いて、連続番号領域S612には、ゼロから始る連続番号を格納する。図6の下図において、S613は、暗号化または復号化したいデータを示している。横軸はデータ長であり、N個の要素を備えている。また、S614は、RND_SIZEバイト単位の区分を示している。本実施例では、RND_SIZEバイト単位で連続番号を振る。S613の最初のRND_SIZEの時ゼロ、次のRND_SIZEの時1、以後一つずつ数字が上がっていく。即ち、S613という暗号化もしくは復号化したいデータを与えた時、S614のようにして得られる数字を連続番号領域S612に格納する。本実施例において、連続番号領域S612は、4バイトの領域を持つものとする。

[0064] また、本実施例において、RND_SIZEは2048バイトを示すものとする。RND_SIZEは、任意の大きさでよいのであるが、使用する疑似乱数が保障する重複の無い乱数表の幅に納まる事が好ましい。本実施例ではメルセンヌ・ツイスタを使用しており、これの安全な幅が2496バイトなので、16進法で切りのよい2048バイトとしている。この幅は使用する疑似乱数発生装置の特性に合わせて最良の値を選ぶべきものである。

[0065] 続いて、S620は空き領域である。図5に示す全体領域S510は624ワード(2496バイト)を確保しており、静的決定暗号鍵S511～S514並びに動的暗号鍵である時間情報領域S611および連続番号領域S612を割り振った残りの領域が、空き領域S620である。

[0066] 本実施例では、この空き領域S620に、時間情報S611と連続番号S612とを繰り返し保存する。ただし、空き領域S620が、ゼロでクリアされていても本発明は問題なく動作する。本実施例で、空き領域S620に上述のように値を設定するのは、メモリ領域S510の取る値をできるだけ、かく乱し、より品質の良い乱数を生成するための工夫である。

[0067] 続いて、図7に実際に行う暗号化復号化の考え方を示す。S710は、暗号化ないし復号化を行いたいデータを示す。このデータはNバイトの長さを持つものとする(図1、図2において、単位データ長を1バイトとした場合に相当する)。また、S711は、図5に示したメモリ領域S510のデータをシードとして生成したRND_SIZEの大きさを持つ乱数

表である(図1、図2の小テーブルS112に相当する)。各乱数表S711は、先頭からの位置に応じて図6のS612に格納する連続番号が変化しているので、それぞれ異なる乱数表として生成される。

[0068] そして、S712は、暗号化ないし復号化を行いたいデータS710と、各乱数表S711との排他的論理和を取った結果を示している。S710が平文の時は暗号化した結果がS712になる。S710が暗号文の時は復号化し解読した結果がS712となる。

[0069] 続いて、図8に本実施例の暗号復号化装置を実現するプログラムの入出力を示す。本実施例では暗号化と復号化は同一の処理を行うので以後、単に「変換」と記載する。

[0070] まず、アドレス参照変数を格納するメモリ領域を定義しておく。アドレス参照変数を格納するメモリ領域は、コンピュータシステムおよび携帯端末において実装するメモリ領域を参照する位置を格納するものであり、そのコンピュータシステムにより変化する。

[0071] 本実施例は、米マイクロソフト社の32ビットの基本制御ソフトで試作したので、アドレス参照変数を格納するメモリ領域は4バイトである。ここで、アドレス参照変数は、ポインタ型変数、ポインタ変数またはアドレス変数と同義である。

[0072] S810は、アドレス参照変数を格納するメモリ領域である。その内容として図5に示したメモリ領域S510の存在するメモリアドレスを格納する。

[0073] S811は、アドレス参照変数を格納するメモリ領域である。その内容は、暗号化もしくは復号化したいデータを格納したS814の存在するメモリアドレスを格納する。S814は暗号化もしくは復号化するデータを格納したテーブルである。

[0074] ここで、本プログラムは暗号復号化を行う部品であり、実際にはプログラムを利用する上位のプログラムが別途存在する。上位プログラムは変換を行うときS817により表すデータの全体像を持つ。そして、S814はS817の任意の写像であるものとする。上位プログラムは、S817の変換したい場所をメモリ上のテーブルの形で確保格納しこのメモリアドレスをS811に格納するものとする。

[0075] S815は、S814に格納したデータがS817のどこから始るデータかを表す位置情報である。先頭であればゼロである。100バイト目であれば100である。

WAVELYNX EX. 1003, Pg. 904
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0076] S816は、S814に格納したデータが何バイトであるかを表す情報である。

[0077] S812は、8バイトのメモリ領域である。内容としてS815を格納する。S812は、32ビットの基本制御ソフトにおいて通常は4バイトで十分であるが、本プログラムでは余裕を持たせて8バイトとする。もしS817が単純にファイルもしくはメモリ上の小さなデータであれば4バイトで十分である。一方で、ハードディスクを丸ごと変換する場合もある。この場合変換するデータ全体は数十Gバイトになる。そのような場合も考えて8バイトとしている。

[0078] S813は、4バイトのメモリ領域である。内容としてはS816を格納する。

[0079] 続いて、図9に本実施例の制御の流れを示す。図8により入出力を定めたプログラムの実際の流れを示すものである。

[0080] 本出願の流れ図の特殊記号はコンピュータ言語であるC言語に準拠する。

%・・・剰余計算を示す記号である。

^ ・・・排他的論理輪を示す記号である。

++ ・・・インクリメント、値を一つ増加を意味する記号である。

[n] ・・・配列を意味する記号である。nは配列の添字である。

* ・・・アドレス型変数の前に記す場合は、変数の示すアドレス上の値を示す。

[0081] S910は、初期化処理を行なっている。ここでは、4バイトのサイズを持つ変数no、4バイトのサイズを持つ変数ipおよびRND_SIZEの大きさを持つ配列変数rnd_tblを確保している。no,ip,rnd_tblの内容は初期化する必要はない。

[0082] S911では、S812が格納する値をRND_SIZEで割り算した結果をnoに格納している。これにより、変換に必要な乱数テーブルの連続番号を算出している。

[0083] S912は、S812の剰余を求め、ipに代入している。

[0084] S913では、連続番号領域S612にnoを代入している。つまり、ここでS612に連続番号を格納している。連続番号領域S612のアドレスは、S810が格納しているS510のメモリアドレスからその位置を算出できる。

[0085] S914では、メルセンヌ・ツイスタにより乱数表を発生させている。具体的には、後述の図10、図11の処理を実行する。後述するS1010にS810の値を格納すると共に、S1011にはrnd_tblのメモリ上の開始アドレスを格納する。そして、図10、図11の処理を

行う。

[0086] S915はループである。初期値として変数i=ゼロとし、一回りするごとにiを1加算する。そして、iがRND_SIZE回ループする。つまりi= 0…i=RND_SIZE−1の値の間でループを繰り返す。

[0087] S916は制限子である。ipは、現在使用している乱数表rnd_tblの参照位置を示す。ipがRND_SIZE以下の値を取る時、正しく乱数表を参照している。一方、ipがRND_SIZEをオーバーした時は、乱数表を使い切ったときである。

[0088] S917は、実際の変換処理を行っている。S811の示すメモリアドレスの上からi番目の要素とrnd_tblの上からip番目の要素の排他的論理和をとり、代入している。代入先はS811の示すメモリアドレスの上からi番目の要素である。

[0089] S918ではipを1つ増加している。これにより、利用する乱数の要素を次にずらしている。

[0090] S919では連続番号を1つ増加している。この処理はいままで使っていた乱数を使い切った時の処理である。次の乱数表を生成する準備を行っている。

[0091] S920はipにゼロを代入している。新しい乱数を生成するにあたり使用開始位置を示すipをクリアしている。

[0092] S921では、S612にnoを代入している。つまり、ここでS612に連続番号を格納している。

[0093] 続いて、図10に乱数生成プログラムの入出力を示す。

[0094] S1010は、アドレス参照変数を格納するメモリ領域である。その内容としてS510の存在するメモリアドレスを格納する。

[0095] S1011は、アドレス参照変数を格納するメモリ領域である。その内容は、乱数表の存在するメモリアドレスを格納する。この乱数表は、図9に記載のrnt_tblのことである。S910で確保したRND_SIZEの大きさを持つメモリ領域である。

[0096] 続いて、図11に乱数生成プログラムの処理を示す。

[0097] S1110では、初期化処理を行う。ここでは、内部で使用する変数を確保している。noは4バイトサイズの変数である。また、ptrはアドレス型変数であり、4バイト単位のメモリを参照する目的で確保する。iは4バイトサイズの変数である。

[0098] また、メルセンヌ・ツイスタが使用する内部作業メモリもここで初期化する。これは、内部作業メモリの初期値が不定のためにメルセンヌ・ツイスタの生成する乱数が不定になることを避けるためである。

[0099] S1111では、乱数のシードを与えている。メルセンヌ・ツイスタが公開している標準関数init_by_array()を実行している。この関数には、S510をシードとして与えている。同関数は、シード発生の鍵となるデータを格納しているメモリ領域のアドレスと、その長さとを指定して呼び出すものである。ここでは、S510の開始アドレスとその領域の長さを与えている。

[0100] S1112では、変数noにRND_SIZEを4で割った値を代入している。メルセンヌ・ツイスタは、1ワードつまり4バイト単位で乱数を発生させる。従い、ここでRND_SIZEを充填するのに必要な乱数の発生回数を算出している。

[0101] S1113は、変数ptrにS1011の値を代入している。

[0102] S1114は、no回ループ処理をしている。

[0103] S1115は、乱数を発生させている。メルセンヌ・ツイスタの公開関数genrand_int32()を実行している。この関数の帰り値は4バイトの乱数である。この4バイトの乱数をptrの示すアドレス領域に代入している。一度に4バイトの乱数を格納している。

[0104] S1116は、任意のブラックボックス処理である。この処理はなくても良い。メルセンヌ・ツイスタが生成する乱数をそのまま採用すると、アルゴリズムが公開されているので暗号用乱数としては好ましくない。これを是正するために、任意の変形をかけるのが好ましい。その変形処理をS1116で行っている。S1116がなくても問題なく動作する。しかし入れることにより信頼性を上げることができる。

[0105] S1117は、変数ptrを増加している。ここでは、ptrの示すメモリアドレスの値を4バイト分加算している。変数ptrは4バイトを参照するアドレス型変数である。ゆえに、指示する値を一つ増やすと言うことは次の4バイトの領域を示すこととなり、実際にはメモリアドレスが4つ程増加する。表記上は1加算であるが、バイト換算では4加算であることをここに付記する。

[0106] 続いて、図12に固有機器のユニークコードを取得するプログラムの流れを記述する。本実施例は暗号化復号化を行う装置であり、固有機器のユニークコードは、上位プ

ログラムないし上位装置にて取得を行い、図5のS513に格納するものである。ここでは、試作にあたり実装した上位プログラムがユニークコードを取得するプログラムを記述する。ここにおいて、ユニークコードの取得が可能であることを証明し、その機構を表す。

[0107] 本実施例では、固有機器として指紋認証機能内蔵のUSBメモリ装置の認識とその固有コードの識別を行う。

[0108] S1220は、USB機器の持つ、製造会社コード、製品番号、シリアル番号を表にして表したものである。16進4バイトの製造会社を識別するコードを持つ。同様に16進4バイトの製品番号を持つ。そして、16バイトのシリアル番号を持つ。本試作では、シリアル番号の桁数は余裕を持たせ54バイトとし、合計で64 バイトとしている。

[0109] S1221は、この会社番号、製造番号、製品番号を米マイクロソフト社の32ビットの基本共通ソフトのレジストリに登録している文字列を参考に表している。基本共通ソフトにより表現方法は異なるが、このように登録して管理している。各コードをアスキー文字列に変換して一つの文字列に連結して管理している。基本共通ソフトは、このようにして現在接続しているUSB装置のユニークコードを取得できるようにしている(レジストリとは、基本共通ソフトが管理用に内蔵しているデータベースである)。

[0110] S1210は、初期化を行っている。当プログラムが終了し、呼び出しもとの上位プログラムに渡す値をゼロクリアするものとする。

[0111] S1211は、永久ループである。

[0112] S1212は、基本共通ソフトから現在接続しているUSB機器のユニークコードを取得する。本試作では初回は最初の接続機器を取得。ループして2回目には2つめの接続機器を取得。これを繰り返し接続機器の取得を全て行うと取得失敗の動作をする。

[0113] S1213では、接続機器S311の有無を判定する。なしの場合は、プログラムを終了する。この時、S1210で設定しているゼロクリアしたメモリ領域を上位プログラムに返り値として格納している。

[0114] 接続機器S311が存在する場合、S1214では、接続機器S311から取得したS1221のごとき値の会社番号を判定する。この際、事前に本プログラムに書き込んでいる製造会社コードとの一致を判定する。プログラムに組み込む製造会社コードは本プログラム

WAVELYNX EX. 1003, Pg. 908
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

を製造販売する時に純正指定もしくは対応する会社のコードを埋め込むものとする。

[0115] 製造会社コードが一致した場合、S1215では、取得したS1221のごとき値の製品番号を判定する。この際、事前に本プログラムに書き込んでいる製品番号コードとの一致を判定する。プログラムに組み込む製品番号コードは、本プログラムを製造販売する時に純正指定もしくは対応する会社のコードを埋め込むものとする。

[0116] 製品番号コードが一致した場合、S1216において、製造会社コード、製品番号コード、シリアル番号を帰り値に格納する。

[0117] 本プログラムの機構を利用することにより、図5の接続機器情報領域S513に、接続機器S311のユニークコードを登録し、本実施例の接続機器S311を利用した暗号鍵の使用を行う。

[0118] 本実施例は、以上説明した技術を用いることにより、(1)図5に示すメモリ領域S510に静的暗号鍵と動的暗号鍵からなる暗号鍵を格納し、(2)この暗号鍵に基づいて生成したシードをメルセンヌ・ツイスタに入力して乱数表を生成する。当該乱数表は、暗号鍵に連続番号S612が含まれることによって、それぞれ異なる乱数表0～nとして図7のごとく生成される。(3)そして、図7に示すように、暗号化または復号化するデータS710と、各乱数表S711との排他的論理和を取ることによって、暗号化結果または復号結果S712を得るものである。(4)そのコンピュータ処理は、図18および図3に示すハードウェアによって行われ、その処理内容は、図8から図12までに説明したメモリ領域の利用とプログラム処理によって、行われる。

[0119] 本実施例において、すべてのコンピュータ処理は、コンピュータがプログラムを実行することによって実現される。また、コンピュータ処理に用いるデータは、コンピュータによって、必要に応じて記憶装置から読み出され、また、必要に応じて記憶装置に格納される。ユーザからコンピュータへの指示は入力装置の操作によって行われ、また、コンピュータからコンピュータへの情報出力は、表示装置などの出力装置によって行われる。後述の実施例においても同様である。

[0120] [実施例2]

[0121] 実施例2では、実施例1よりも高速に乱数を生成する装置およびプログラムを紹介する。まず、図13に基づき、本実施例の変換(暗号化／復号化)の概要を説明する。

[0122] S1310,S1311,S1312は、RND_SIZEの乱数表である。この乱数表は、大きさはそれぞれ任意でよいが、管理しやすいという理由でここではRND_SIZEとしている。

[0123] また3つ用意しているのも、本来はn個となり任意でよいが、RND_SIZEの大きさから現実的な数としと3を割り出している。今、S1310,S1311,S1312には、それぞれ異なる乱数表を格納しているものとする。

[0124] S1313は、変換を行いたい任意のデータである。また、S1319は、変換を行った結果のデータである。

[0125] S1314は、S1310,S1311,S1312の排他的論理和を取り生成したRND_SIZEの大きさを持つ乱数表である。S1314の最初の1バイトは、S1310,S1311,S1312それぞれの最初の1バイトについて排他的論理和を取った結果である。S1314の各要素は、S1310,S1311,S1312の同じ位置の要素について排他的論理和を取った結果である。

[0126] S1315は、S1314と同一の工程で生成した乱数表である。ただし、生成する前に、S1311を1バイト回転させた上で生成したものである。この回転は、時計方向、反時計方向のどちらでもよい。ここでは、反時計回りに1バイト回転させたものとする。

[0127] S1316、S1317は、同様にしてS1311を各1バイトずつ回転させてから生成している。このようにして乱数を生成していくと、RND_SIZE 回繰り返すとS1311は循環して同じ乱数を生成するようになる。この時、S1312を1バイト反時計回りで回転させる。このようにして、n個の乱数表の組み合わせで長周期の乱数を生成できる。ここでは、RND_SIZEの乱数表が3個あるので、2048×2048×2048=8,589,934,592つまり約8Gバイトで循環して元に戻る周期を持つ乱数表を作ることが出来る(つまり、3つの乱数表S1310,S1311,S1312の値の組み合わせをずらしながら乱数表0～nを生成する)。一つの変換したいデータに対して個別に生成するのであれば8Gバイトは十分な大きさである。

[0128] 乱数表S1310,S1311,S1312は、S1319一つごとにそれぞれ乱数表を生成できると最も好ましい。しかし、S1311,S1312を事前に生成しておき、S1310だけS1319一つごとに生成しても品質のよい乱数を生成できる。

[0129] 本実施例は、この方式を採用した暗号化復号化を行うプログラムと装置を提供するものである。

[0130] 図14に本実施例の暗号復号化装置を実現するプログラムの入出力を表す。本実

施例では暗号化と復号化は同一の処理を行うので以後、単に変換と記載する。

[0131] まず、アドレス参照変数を格納するメモリ領域を定義しておく。アドレス参照変数を格納するメモリ領域は、コンピュータシステムおよび携帯端末において実装するメモリ領域を参照する位置を格納する物であり、そのコンピュータシステムにより変化する。本実施例は、米マイクロソフト社の32ビットの基本制御ソフトで試作したので、アドレス参照変数を格納するメモリ領域は4バイトである。

[0132] S1413は、アドレス参照変数を格納するメモリ領域である。この領域には、暗号化もしくは復号化したいデータS814の存在するメモリアドレスを格納する。図14において、暗号化もしくは復号化したいデータS817と、その部分であるS814と、データS817の中でS814が存在する先頭からの位置S815と、S814の長さS816とは、実施例1の図8に示したものと同一である。よって、同一符号を付して重複説明は省略する。

[0133] S1414は、8バイトのメモリ領域である。この領域には、S815を格納する。この領域は、32ビットの基本制御ソフトにおいて通常は4バイトで十分であるが、本プログラムでは余裕を持ち8バイトとする。S817が単純にファイルもしくはメモリ上の小さなデータであれば4バイトで十分である。しかし、ハードディスクを丸ごと変換する場合もあり、この場合変換するデータ全体は数十Gバイトになる。そのような場合も考慮して8バイトとしている。

[0134] S1415は、4バイトのメモリ領域である。この領域にはS816を格納する。

[0135] S1416は、RND_TBLのサイズを持つ乱数表である。図13のS1310に相当する。これは実施例1で説明した図10、図11のプログラムにより格納データを生成するものとする。

[0136] S1417は、RND_TBLのサイズを持つ乱数表である。図13のS1311に相当する。これは実施例1で説明した図10、図11のプログラムにより格納データを生成するものとする。

[0137] S1418は、RND_TBLのサイズを持つ乱数表である。図13のS1312に相当する。これは実施例1で説明した図10、図11のプログラムにより格納データを生成するものとする。

[0138] S1410は、アドレス参照変数を格納するメモリ領域である。その内容は、S1416が存

在するメモリ上のアドレスを格納する。

[0139] S1411は、アドレス参照変数を格納するメモリ領域である。その内容は、S1417が存在するメモリ上のアドレスを格納する。

[0140] S1412は、アドレス参照変数を格納するメモリ領域である。その内容は、S1418が存在するメモリ上のアドレスを格納する。

[0141] 続いて、図15に本実施例の制御の流れを示す。これは、図14により入出力を定めたプログラムの実際の流れである。

[0142] まず、S1510において、初期化処理を行う。ここでは、4バイト型の変数i,ip1,ip2,ip3,lwkを確保している。ip1は、S1416が示す乱数表を管理する。ip2は、S1417が示す乱数表を管理する。ip3は、S1418が示す乱数表を管理する。

[0143] S1511は、S1416の示す乱数表の最初に使用する位置を算出しip1に代入している。S1511の乱数は、RND_SIZEの大きさのバッファを巡回利用する。従い、RND_SIZEで剰余を取った結果が最初の使用開始点となる。

[0144] S1512は、S1417の示す乱数表の最初に使用する位置を算出しip2に代入している。S1512はS1416を一回循環することで1バイト循環する。従い、S1314をRND_SIZEで割り算出している。

[0145] S1513は、S1418の示す乱数表の最初に使用する位置を算出しip3に代入している。S1513はS1417が一回循環することで1バイト循環する。従い、ip2をRND_SIZEで割り算出している。

[0146] S1514では、ip2をRND_SIZEで剰余を取り、ip2に代入している。S1512でのip2への代入計算は、S1417が1回転以上循環した時の処置をしていない。ここで、剰余を取り循環した結果を算出している。

[0147] S1515で、ip3をRND_SIZEで剰余を取り、ip3に代入している。S1513でのip3への代入計算は、S1418が1回転以上循環した時の処置をしていない。ここで、剰余を取り循環した結果を算出している。S1418が1回転する事態とは、同じ乱数を発生して再利用することを意味する。従い、本来発生してはいけない事態であるが安全装置として存在している。この安全装置が意味をなす時とは、8Gバイトを越す変換したいデータが発生した時である。通常はまず考えられない。

WAVELYNX EX. 1003, Pg. 912
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0148] S1516では、変数lwkにS1414つまり変換したいデータの位置情報を格納している。

[0149] S1517はループ処理をする。S1415に格納されている値だけループする。変数iをループカウンタとし、i=0に始まりループごとにiを一つずつ増加させながらループする。

[0150] S1518とS1519では、3つの乱数表より変換情報を生成している。具体的にはS1518で、S1416の上でip1の示す値を要素番号とする値と、S1417の上でip2の示す値を要素番号とする値との排他的論理輪をwkに格納する。S1519では、wkとS1418の上でip3の示す値を要素番号とする値との排他的論理輪を取りwkに格納している。

[0151] S1520において、変換したいデータを格納しているS1413の先頭からi番目のデータにwkの排他的論理和を代入している。これにより、S1413 の変換を1バイト行う。

[0152] S1521では、変数ip1の格納値を一つ増加している。乱数表S1416の次に使用する値を設定している。

[0153] S1522では、変数ip2の格納値を一つ増加している。乱数表S1417の次に使用する値を設定している。

[0154] S1522Aでは、変数ip3の格納値を一つ増加している。乱数表S1418の次に使用する値を設定している。

[0155] S1523では、変数lwkの格納値を一つ増加している。lwkは変換したいデータの全体像の中の最初からの位置を示すものである。

[0156] S1524では、ip1とRND_SIZEとの大小比較を行う。

[0157] S1525は、ip1の格納値がRND_SIZEと同じか大きい時に行う処理である。この判定は乱数表S1416の要素数を超えた時の処理である。ip1にゼロを設定し、乱数表S1416の使用を先頭に戻している。

[0158] S1526では、変数ip2の格納値を一つ増加している。乱数表S1417の次に使用する値を設定している。乱数表S1416が一回転したので、S1417を1バイトずらす処置をここでしている。

[0159] S1527では、lwkをRND_SIZE2にて割った際の剰余を求める。結果がゼロの時は、lwkの値はRND_SIZE2の整数倍の時である。この時は、乱数表S1416,S1417の全組み合わせがなされた時である。有値の時は、乱数表S1416,S1417の全組み合わせが一巡していないときである。

WAVELYNX EX. 1003, Pg. 913
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0160] S1528では、乱数表S1416とS1417との組み合わせが全てなされたとみなし、乱数表S1418を1バイトずらす処理をしている。変数ip3の格納値を一つ増加している。

[0161] S1529では、ip2とRND_SIZEとの大小比較を行う。

[0162] S1530では、変数ip2にゼロを代入している。乱数表S1417の使用領域が最後に至ったので初期化して先頭に戻す処理をしている。

[0163] S1531では、ip3とRND_SIZEとの大小比較を行う。

[0164] S1532では、変数ip3にゼロを代入している。乱数表S1418の使用領域が最後に至ったので初期化して先頭に戻す処理をしている。

[0165] 以上説明した本実施例では、実施例1と同様に、図8～図12の処理を経て、3つの乱数表S1310,S1311,S1312を生成しておき、これら3つの乱数表を利用して、図14及び図15の処理を行うことにより、データの変換(暗号化または復号)を行っている。実施例1と重複する部分の説明は省略する。

[0166] さて、高速に乱数を生成する事は開発者の課題となっているが、乱数性を高めようとすると処理が増加し遅くなる傾向がある。一方、本実施例によれば、事前に生成した3つの乱数表を複合させ暗号化に使用する乱数表を生成している。このため、簡単なメモリ操作で乱数表を生成できる。結果、乱数性の高い乱数表を高速に得ることができる。

[0167] [実施例3]

[0168] 続いて説明する実施例3は、暗号化／復号化を行う簡単なアプリケーションプログラムである。暗号化／復号化の処理自体は、実施例1で説明した処理を利用する。

[0169] 図16に本実施例の処理の流れを記述する。本実施例は、基本制御ソフトより実行し、変換(暗号化または復号化)をかけたいファイルを指定する。変換したいファイルを読み込んで変換を実行し、変換結果をファイルとして記憶装置に保存する。本実施例の示すアプリケーションに一回かけると暗号化を行う。生成結果を再度本アプリケーションにかけると復号を行うものである。

[0170] S1610は初期化を行う。4バイト変数のipとnoとiとを確保している。また、図5のS510として使用するメモリ領域を確保している。また、ファイルの読み書きの作業用メモリ領域としてdat_tblを確保している。dat_tblはRND_SIZEの大きさである。

[0171] S1611は、変数ipにゼロを代入している。

[0172] S1612では、S512の記憶領域(コンピュータ固有情報領域)をゼロクリアしている。S512の領域において、メモリの初期値の不定により、変換結果が不定になることをここで回避している。

[0173] S1613では、変換元のファイル名を入力している。これは、ディスク上のファイル名パスを対話形式で当アプリケーションの実行者に質問するものである。ここで指定したファイルを変換の対象とする。

[0174] S1614では、変換元のファイル名をバイナリのリードモードでオープン処理する。

[0175] S1615では、変換先のファイル名をバイナリのライトモードでオープン処理する。オープンする時のファイル名は本実施例では、S1613で取得したファイル名の最後に".enc"という文字列を追加したもので、S1613の変換元ファイルと同じパスに保存するものとする。

[0176] S1617では、端末情報を取得している。具体的には本実施例が動くコンピュータシステムの固有情報を取得するものとする。ここでは、図5に示したコンピュータ固有情報領域S512に、HDDのボリュームシリアル番号S517を保存するものとする。

[0177] S1618では、コンピュータシステムに接続している機器情報を取得している。図12のプログラムを実行し、機器固有情報領域S513に機器情報を登録する。機器が接続していない時はここにゼロクリアされた値が入るものとする。

[0178] S1619では、S1613で取得した変換元ファイルのクリエートタイムを取得している。クリエートタイムとは、そのファイルを生成した時の時間情報である。この時間情報を図6の時間情報領域S611に格納する。ここではクリエートタイムを使用しているが、ライトタイムつまり、最後に書き込みの発生した時間を使用してもよい。重要なことは暗号鍵として利用する時間情報を統一することである。本実施例ではクリエートタイムで統一している。

[0179] 本実施形態において、S510を構成する上記以外の暗号鍵は、ここでは使用しない。未使用の領域にはゼロを値として持つものとする。

[0180] S1620は、iをカウンタとして構成する永久ループである。iの初期値をゼロとし、1回ループするごとにiの値を一つ加算する。

[0181] S1621では、変換元のファイルをRND_SIZEの大きさだけシーケンシャルにてバイナリモードで読み込みを行う。読み込みデータはdat_tbl に格納する。ファイルの残りのデータ量がRND_SIZEに満たない時には、存在する量だけ読み取るものである。

[0182] S1622では、1バイト以上の読み込みが成立したかを判定する。1バイトでも読めたときはYesとする。読み込みがゼロバイトの時は全ファイルデータを読み込み終了とみなしNoに分岐する。

[0183] S1623では、S1621で読み込みに成功したバイト数を変数noに代入する。

[0184] S1624では、連続番号領域S612に連続番号としてiを格納している。乱数のシードを構成するS510の中の動的暗号鍵をここで設定している。

[0185] S1625では図8、図9に記述した変換プログラムを実行している。図8において、S810にはシード情報を格納したS510の存在するメモリ開始アドレスを格納するものとする。S811には dat_tblのメモリ開始アドレスを格納するものとする。S812にはipの値を格納するものとする。S813にはnoの値を格納するものとする。

[0186] S1626では、dat_tblを変換先ファイルにライトする。シーケンシャルでバイナリモードで書き込みを行う。

[0187] S1627では、次のループに備え、ipの値にnoの値を加算している。S1628では、変換先ファイルに時間情報をセットする。この時間情報はS1619で取得した時間情報である。変換先ファイルに対してクリエートタイムとしてこの時間情報を書き込む。
この時間情報をセットすることにより、次回本実施例でこの変換先ファイルを呼んだときに復号ができるのである。

[0188] S1629では変換元ファイルをクローズ処理する。S1630では変換先ファイルをクローズ処理する。

[0189] 本実施例によれば、ファイル処理により簡単に暗号化または復号化を行うことができる。

[0190] [実施例4]

[0191] 続いて紹介する本実施例も、暗号化または復号化を行う簡単なアプリケーションプログラムである。暗号化／復号化の処理自体は、実施例2のプログラムを実行する暗号化装置、復号化装置を使用する。

[0192] 図17に本実施例の処理の流れを記述する。本実施例は基本制御ソフトより実行し、変換をかけたいファイルを指定する。変換したいファイルを読み込んで変換をし、変換結果をファイルとして記憶装置に保存する。本実施例の示すアプリケーションに一回かけると暗号化を行う。生成結果を再度本アプリケーションにかけると復号を行うものである。

[0193] S1710は初期化を行う。4バイト変数のipとnoとiとを確保している。同様にS510として使用するメモリ領域を確保している。同様にファイルの読み書きの作業用メモリ領域としてdat_tblを確保している。

[0194] dat_tblはRND_SIZEの大きさである。同様に乱数表格納用の配列変数として rnd_tbl1,rnd_tbl2,rnd_tbl3を確保する。大きさは3つともRND_SIZEである。

[0195] S1711は変数ipにゼロを代入している。

[0196] S1712では、S512の記憶領域をゼロクリアしている。S512の領域において、メモリの初期値の不定により、変換結果が不定になることをここ回避している。

[0197] S1713では、変換元のファイル名を入力している。これは、ディスク上のパスおよびファイル名を対話形式で当アプリケーションの実行者に質問するものである。ここで指定したファイルを変換の対象とする。また、ここで指定したファイルのクリエートタイム(時間情報)を取得し記憶する。

[0198] S1714では、変換元のファイル名をバイナリのリードモードでオープン処理する。

[0199] S1715では、変換先のファイル名をバイナリのライトモードでオープン処理する。オープンする時のファイル名は本実施例では、S1713で取得したファイル名の最後に".enc"という文字列を追加したもので、S1713で指定した変換元ファイルと同じパスに保存するものとする。

[0200] S1716では、端末情報を取得している。具体的には本実施例が動くコンピュータシステムの固有情報を取得するものとする。ここでは、コンピュータ固有情報領域S512にHDDのボリュームシリアル番号S517を保存するものとする。

[0201] S1717では、rnd_tbl1に乱数表を格納している。図10、図11のプログラムを実行して乱数を生成しrnd_tbl1に格納している。即ち、図10のS1010にS510の存在するメモリアドレスを格納する。また、S1011にはrnd_tbl1の存在するメモリアドレスを格納する。

そして図11のプログラムを実行してrnd_tbl1に乱数表を得る。

[0202] S1718では、コンピュータシステムに接続している機器情報を取得している。図12のプログラムを実行し、機器固有情報領域S513に機器情報を登録する。機器が接続されていない時はこの領域をゼロクリアした後、先頭の4バイトに16進コードの FFFFの値が入るものとする。これはゼロであった場合S510の全体の値がS1717で設定したS510と同じとなり、つまり同一のシードを与えることになり、結果、rnd_tbl1とrnd_tbl2の乱数表の内容が同一となり好ましくないからである。従い、機器情報を取得できない時にはゼロ以外になるようにFFFFを与えている。

[0203] S1719では、rnd_tbl2に乱数表を格納している。図10、図11のプログラムを実行して乱数を生成rnd_tbl2に格納している。即ち、図10のS1010にS510の存在するメモリアドレスを格納する。また、S1011にrnd_tbl2の存在するメモリアドレスを格納する。そして図11のプログラムを実行してrnd_tbl2に乱数表を得る。

[0204] S1720では、S1713で取得したファイルのクリエートタイムを取得している。クリエートタイムとは、そのファイルを生成した時の時間情報である。この時間情報を時間情報領域S611に格納する。ここではクリエートタイムを使用しているがライトタイムつまり、最後に書き込みの発生した時間を使用してもよい。重要なことは暗号鍵として利用する時間情報を統一することである。本実施例ではクリエートタイムで統一している。

[0205] 本実施例において、S510を構成する未設定の暗号鍵はここでは使用しない。未使用の領域にはゼロを値として持つものとする。

[0206] S1721では、rnd_tbl3に乱数表を格納している。図10、図11のプログラムを実行して乱数を生成しrnd_tbl3に格納している。即ち、図10のS1010にS510の存在するメモリアドレスを格納する。S1011にrnd_tbl3の存在するメモリアドレスを格納する。そして図11のプログラムを実行してrnd_tbl3に乱数表を得る。

[0207] S1722は、iをカウンタとして構成する永久ループである。iの初期値をゼロとし、1回ループするごとにiの値を一つ加算する。

[0208] S1723では、変換元のファイルをRND_SIZEの大きさだけシーケンシャルにてバイナリモードで読み込みを行う。読み込みデータはdat_tbl に格納する。ファイルの残りのデータ量がRND_SIZEに満たない時には、存在する量だけ読み取るものである。

[0209] S1724では、1バイトでも読み込みが成立したかを判定する。1バイトでも読めたときはYesとする。読み込みがゼロバイトの時は全ファイルデータ読み込み終了とみなしNoに分岐する。

[0210] S1725で、変数noに、S1723で読み込みに成功したバイト数を代入する。

[0211] S1726では、連続番号領域S612に連続番号としてiを格納している。乱数のシードを構成するS510の中の動的暗号鍵をここで設定している。

[0212] S1727では図14、図15に記述した変換プログラムを実行している。図14の各入出力への設定を次に記述する。

[0213] S1410にはrnd_tbl3の存在するメモリ領域開始アドレスを格納する。

[0214] S1411にはrnd_tbl2の存在するメモリ領域開始アドレスを格納する。

[0215] S1412にはrnd_tbl1の存在するメモリ領域開始アドレスを格納する。

[0216] S1413にはdat_tblのメモリ開始アドレスを格納するものとする。S1414にはipの値を格納するものとする。S1415にはnoの値を格納するものとする。

[0217] S1728では、dat_tblを変換先ファイルにライトする。シーケンシャルでバイナリモードで書き込みを行う。

[0218] S1729では次のループに備え、ipの値にnoの値を加算している。

[0219] S1730では変換先ファイルに時間情報をセットする。この時間情報はS1713で取得した時間情報である。さらに変換先ファイルに対してクリエートタイムとしてこの時間情報を書き込む。この時間情報をセットすることにより、次回本実施例でこの変換先ファイルを呼んだときに復号ができるのである。

[0220] S1731では変換元ファイルをクローズ処理する。S1732では変換先ファイルをクローズ処理する。

[0221] このようにしても、ファイル処理により簡単に暗号化または復号化を行うことができる。

[0222] [実施例5]

[0223] 続いて説明する実施例5は、実施例1において生成する乱数に、更に時間情報を影響させ、乱数表をかく乱し、生成する乱数表間における重複の可能性を排除するものである。

[0224] 実施例1が前提としている乱数発生装置であるメルセンヌ・ツイスタは内部に624ワード(1ワードは4バイト)の内部メモリを備え、非特許文献1によるならば、論理上はシードが異なればそのシードに属する624ワード(2496バイト)のうち最低でも1ワードは異なる乱数表を生成することを保障している。つまり、2496バイト単位で異なる乱数表が生成されることを保障している。逆に、2496以下の小さな表においては部分的に同一の乱数表が出現する可能性を示している。

[0225] 本実施例は、2496バイト単位以下の小さな乱数表の範囲で、時系列変化に従い重複の発生しないかく乱を行い、乱数表内の部分的な重複の可能性を排除する機構を組み込んだものである。

[0226] 図19は、この機構をプログラムにより実現したものである。図19は、図11に記載したS1116の処理(任意のブラックボックス処理)を抽出し示した物である。本実施例は、実施例1のS1116に、図19のプログラムを組み込んで実現されるものである。

[0227] S1910は、ループカウンタiの示す値の3の剰余を調べている。そして、剰余ゼロの場合はS1911へ分岐し、剰余1の場合はS1912に分岐し、剰余2の場合はS1913に分岐する。

[0228] S1911では、時間情報S611の上位4バイトと、変数ptrの示すアドレスのメモリの格納値4バイトとの排他的論理和をとり、ptrの示すアドレスに格納している。

[0229] S1912では、時間情報S611の下位4バイトと、変数ptrの示すアドレスの4バイトとの排他的論理和をとり、ptrの示すアドレスに格納している。

[0230] S1913では、S612つまり4バイトの連続番号と、変数ptrの示すアドレスの4バイトとの排他的論理和をとり、ptrの示すアドレスに格納している。

[0231] 以上により、12バイト(4バイト×3)単位でかく乱をかけている。この12バイト単位のかく乱は時系列変化の中で重複のないかく乱を行うことが出来る。結果、乱数発生装置が生成した乱数に対して、さらに3バイト単位の間隔で重複のない乱数を生成する事ができる。

図面の簡単な説明

[0232] [図1]擬似乱数の周期を示した説明図である。

[図2]データの暗号化を示した説明図である。

[図3]実施例の物理的全体像を示した図である。

[図4]指紋認証記憶装置の説明を示した図である。

[図5]n個の静的決定暗号鍵を説明した図である。

[図6]動的暗号鍵の説明した図である。

[図7]実際の暗号化復号化の考え方を説明した図である。

[図8]暗号化復号化装置および関数fの入出力を説明した図である。(実施例1)

[図9]変換プログラムの処理を説明した図である。(実施例1)

[図10]乱数表生成プログラムの入出力を表した説明図である。(実施例1)

[図11]乱数表の発生プログラムを表した説明図である。(実施例1)

[図12]固有機器のユニークコードを取得するプログラムの説明図である。

[図13]n個の乱数表による乱数表の合成を示した説明図である。

[図14]暗号化復号化プログラムの入出力を示した説明図である。(実施例2)

[図15]変換プログラムの流れを示した説明図である。(実施例2)

[図16]変換アプリケーションを示した説明図である。(実施例3)

[図17]変換アプリケーションを示した説明図である。(実施例4)

[図18]ホストとなる装置を示した説明図である。

[図19]S1116処理の実施例を示した説明図である。(実施例5)

符号の説明

[0233] S1811…中央処理装置(CPU)。

S1812…ROMBIOS。

S1813…メモリ装置。

S1814,S1817,S1818…システムバス。

S1815,S1816…アダプタ。

S1820…ホストI/F。

S1830…HDD。

S1819…リーダライタ。

S1840…USB装置。

WAVELYNX EX. 1003, Pg. 921
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

請求の範囲

[1] コンピュータ処理による乱数表発生装置であって、

使用制限数n（nは正数）以上の要素数からなる乱数表を生成する手段と、

生成した乱数表を記憶する領域と、

前記使用制限数nを記憶する領域と、

前記乱数表の要素を前記使用制限数nまで使用したとき当該乱数表のn＋1以降の要素を使用せずに別の乱数表を生成させる手段と、

を備えたことを特徴とする乱数表発生装置。

[2] コンピュータ処理による乱数表発生装置であって、

起源の異なる複数の暗号鍵を取得する手段と、

前記起源の異なる複数の暗号鍵を記憶する領域と、

前記起源の異なる複数の暗号鍵をシードとして一つの乱数表を生成する手段と、

前記乱数表を記憶する領域と、

を備えたことを特徴とする乱数表発生装置。

[3] 前記乱数表発生装置は、メルセンヌ素数を使用した疑似乱数発生方式又はメルセンヌ・ツイスタによる疑似乱数発生方式により前記乱数表を生成することを特徴とした請求項1又は2に記載の乱数表発生装置。

[4] 前記暗号鍵の一つとして時間情報を含むことを特徴とした請求項2に記載の乱数表発生装置

[5] 請求項4に記載の乱数表発生装置において、

前記乱数表と前記時間情報とを演算した結果を前記乱数表とする手段を備えたことを特徴とする乱数表発生装置。

[6] 前記暗号鍵の一つとして、前記乱数表発生装置としてのコンピュータシステムに接続している機器の固有情報を含むことを特徴とした請求項2記載の乱数表発生装置。

[7] 前記暗号鍵の一つとして、前記乱数表発生装置としてのコンピュータシステム自体の固有情報を含むことを特徴とした請求項2記載の乱数表発生装置。

[8] 前記コンピュータシステムの固有情報として、当該コンピュータシステムが内蔵する

WAVELYNX EX. 1003, Pg. 922
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

記憶装置のシリアル番号を取得することを特徴とした請求項11記載の乱数表発生装置。

[9] 前記暗号鍵の一つとして、ユーザの生体情報を含むことを特徴とした請求項2記載の乱数表発生装置。

[10] 前記暗号鍵の一つとして、入力装置から入力された又は自動的に設定したキーワードを含むことを特徴とした請求項2記載の乱数表発生装置。

[11] 前記暗号鍵の一つとして、連続番号等のユニークコードを、前記生成する乱数表毎に与える手段を備えたことを特徴とする請求項2記載の乱数表発生装置。

[12] 請求項1又は2に記載の乱数表発生装置において、

前記乱数表を複数生成し当該複数の乱数表の要素を組み合わせることにより新たな乱数系列を生成する手段を備えたことを特徴とする乱数表発生装置。

[13] 請求項1乃至12のいずれかの乱数表発生装置と、

暗号化又は復号化すべき情報を記憶する領域と、

前記乱数表を用いて前記暗号化又は復号化すべき情報を暗号化又は復号化する手段と、

前記暗号化又は復号化した情報を記憶する領域とを備え、

前記暗号化又は復号化する手段は、暗号化又は復号化すべき情報毎に、前記乱数表発生装置から異なるシードによる乱数表を個別に生成させ、最近に生成させた乱数表を用いて暗号化又は復号化を行うことを特徴とした暗号化／復号化装置。

[14] 請求項4又は5に記載の乱数表発生装置において、前記時間情報は、前記暗号化又は復号化すべき情報に関連する時間情報とすることを特徴とした請求項13に記載の暗号化／復号化装置。

[15] 請求項13に記載の暗号化／復号化装置において、

暗号化した情報をコンピュータの基本制御ソフトが持つファイル管理システムによりファイルとして記憶装置に保存させる手段と、

当該暗号化に用いた前記時間情報を当該ファイルの生成時間もしくは更新時間として当該ファイルに保存させる手段とを備えたことを特徴とする暗号化／復号化装置。

WAVELYNX EX. 1003, Pg. 923
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[16] 請求項11に記載の乱数表発生装置において、前記ユニークコードとしてセクタ及びブロック等の記憶装置の記録単位を指標する値を与えるものとし、

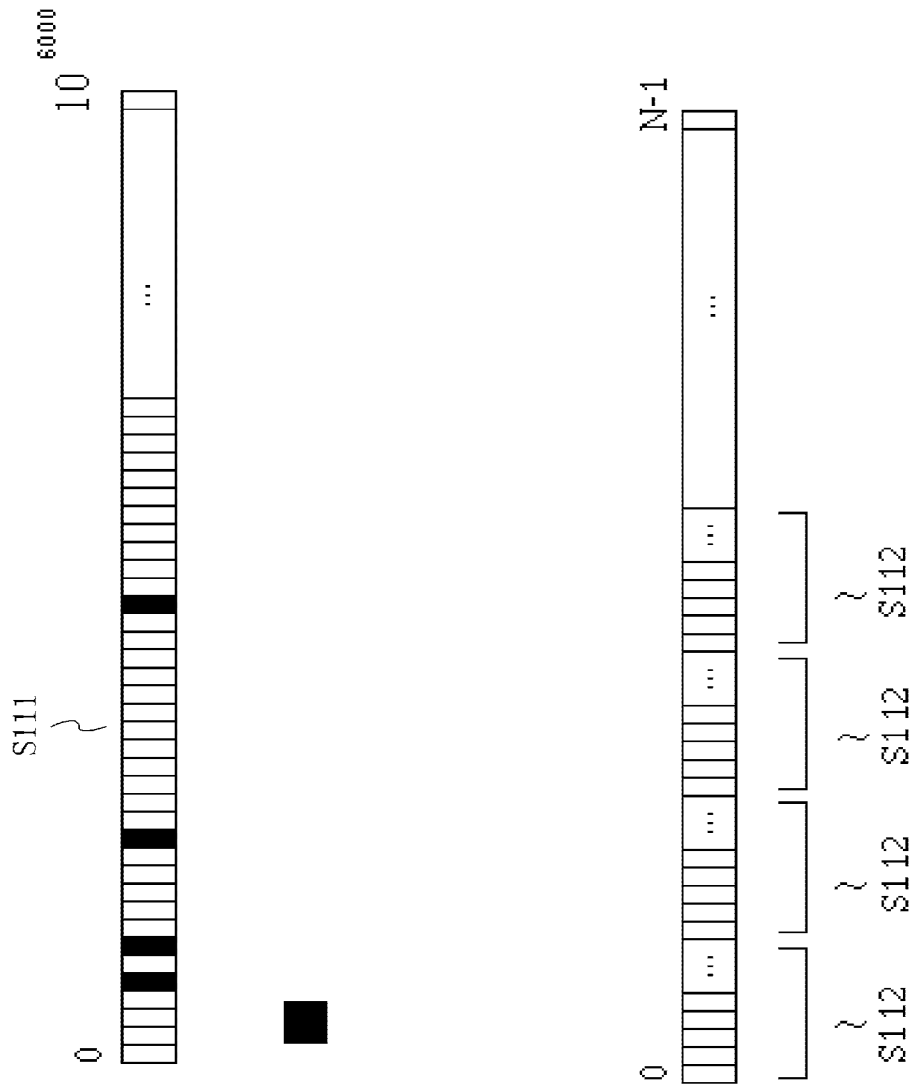
ある記録単位に記録されている情報を暗号化又は復号化する際に用いる乱数表は、その記録単位を指標する値を前記ユニークコードに与えて生成する手段を備えたことを特徴とする請求項13に記載の暗号化／復号化装置。

[17] 請求項1乃至16のいずれかの装置において、

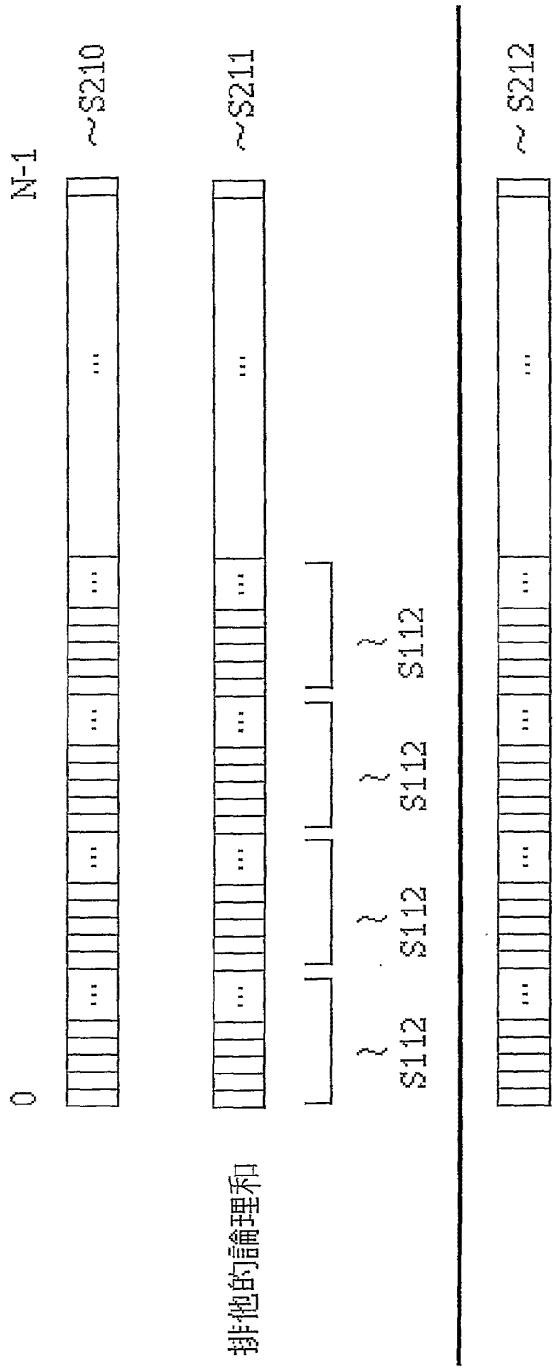
前記各手段の動作をコンピュータに実行させるためのプログラム。

[18] 請求項17に記載のプログラムを記録したコンピュータ読み取り可能な記録媒体。

[図1]

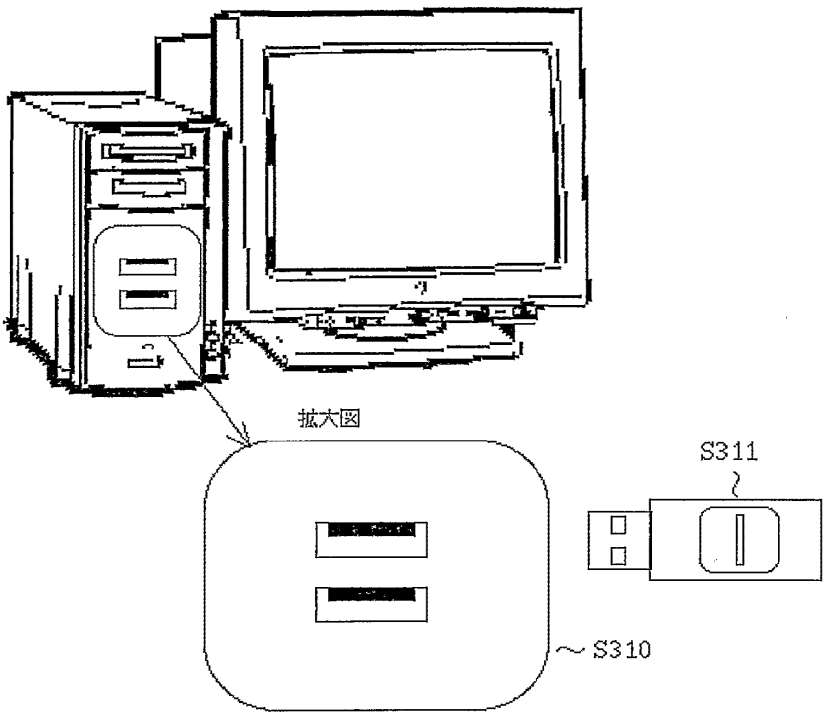


[図2]



WAVELYNX EX. 1003, Pg. 926
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

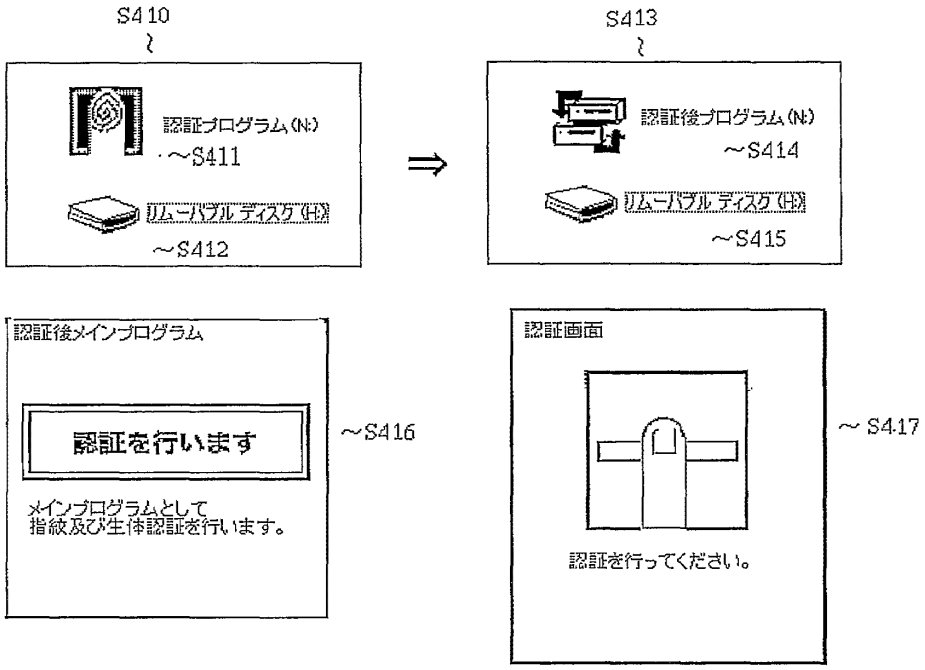
[図3]



差替え用紙(規則26)

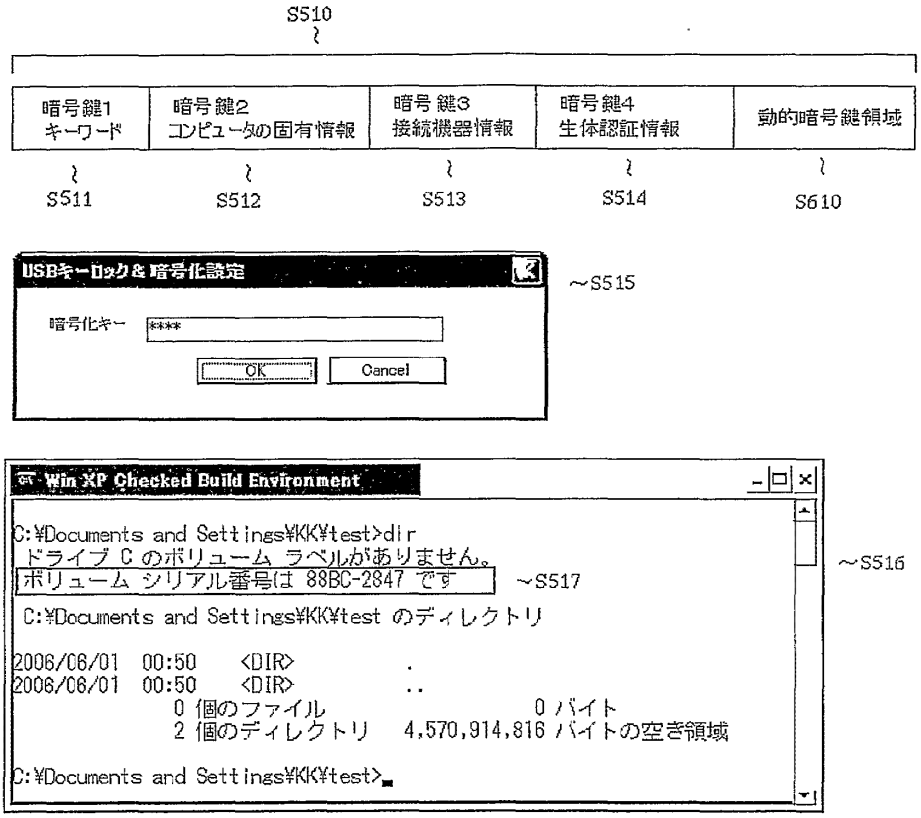
WAVELYNX EX. 1003, Pg. 927
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図4]



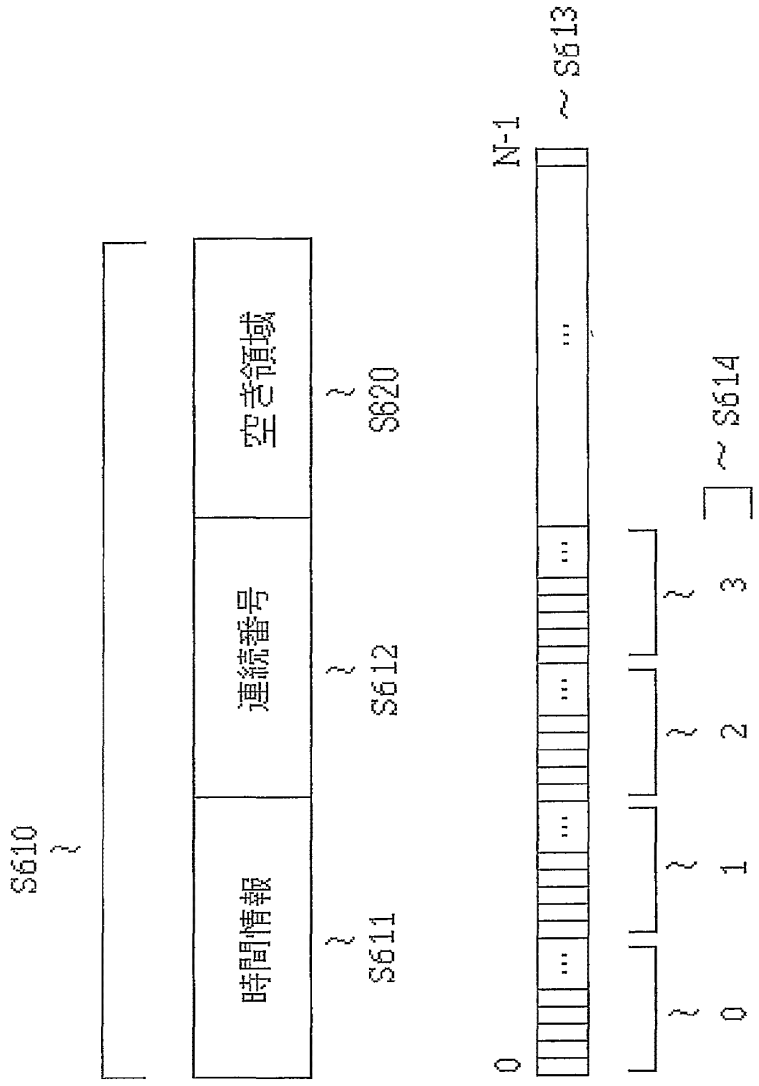
WAVELYNX EX. 1003, Pg. 928
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図5]



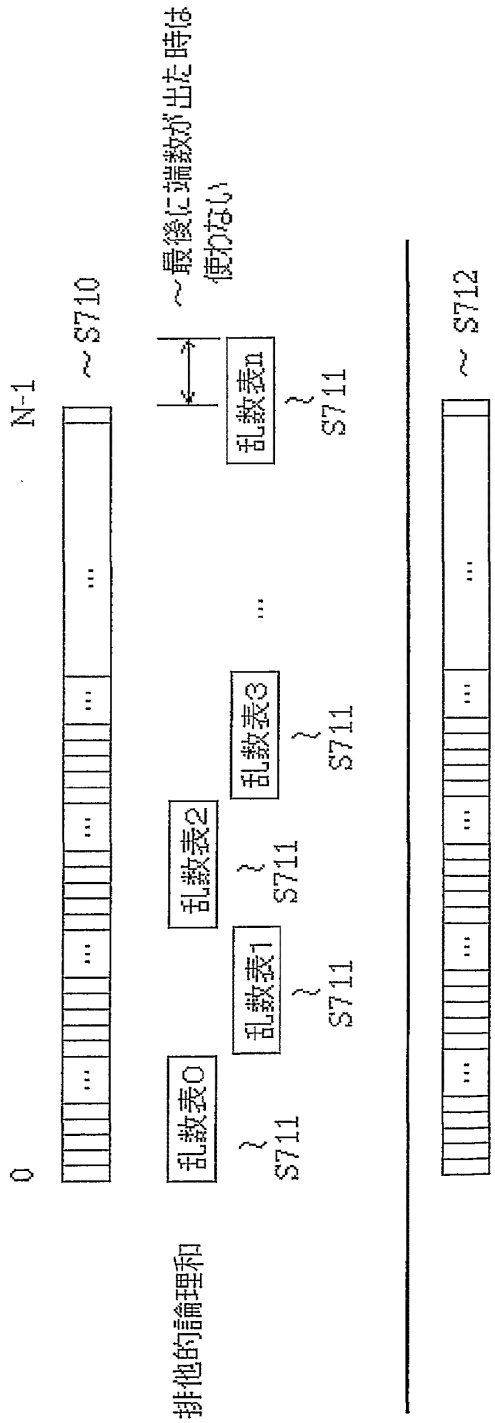
WAVELYNX EX. 1003, Pg. 929
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図6]



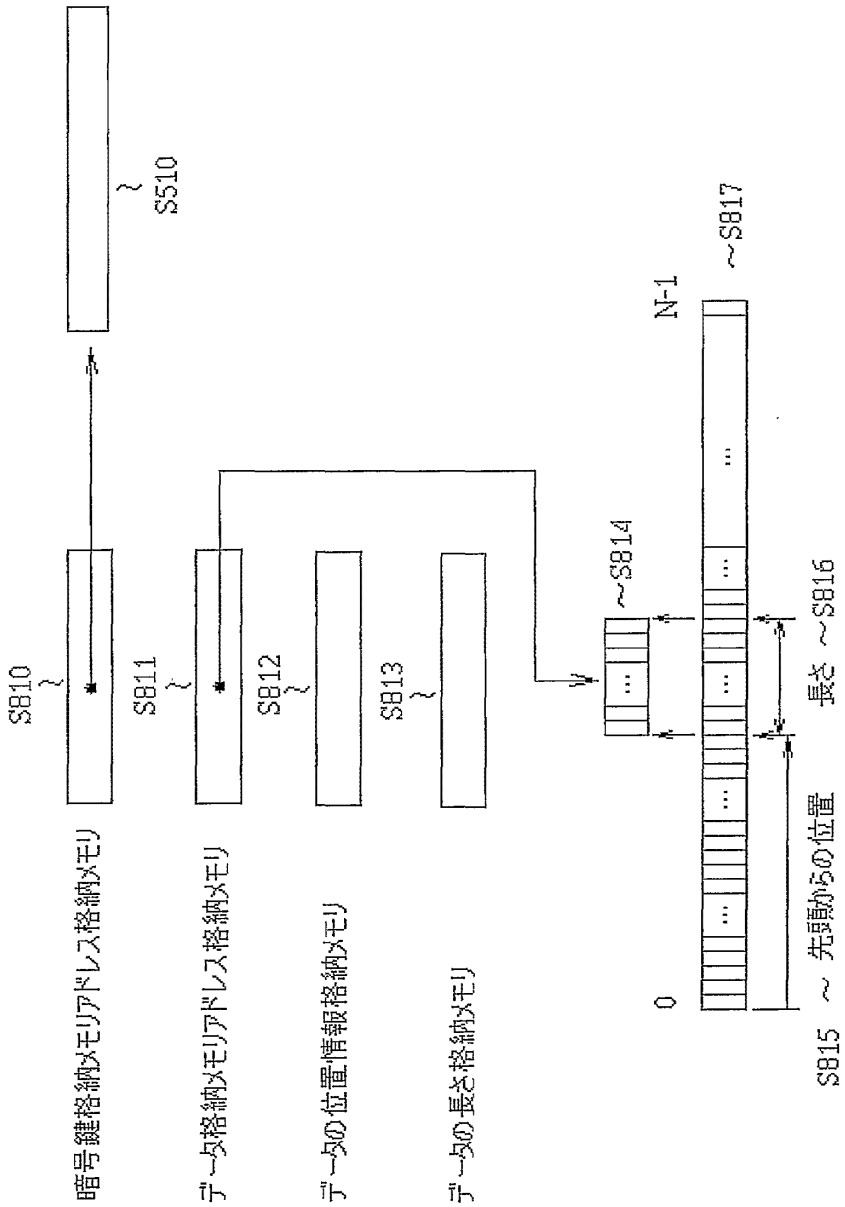
WAVELYNX EX. 1003, Pg. 930
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図7]



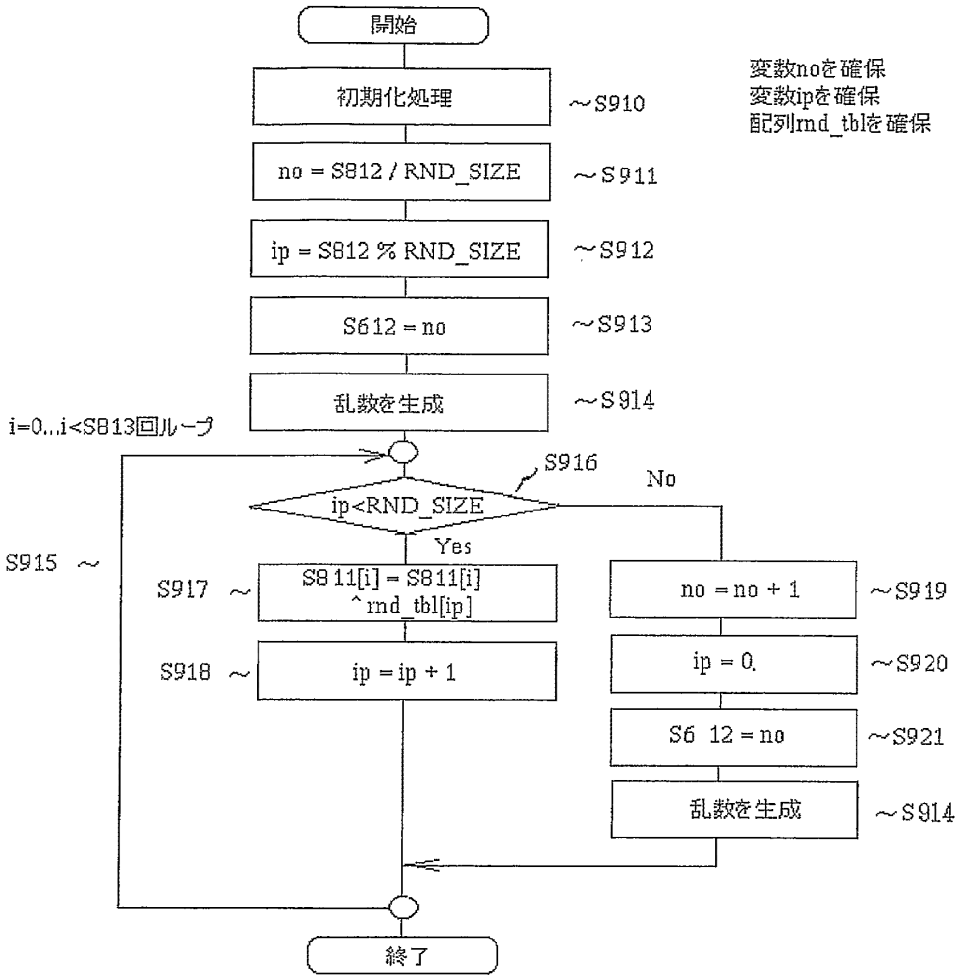
WAVELYNX EX. 1003, Pg. 931
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図8]



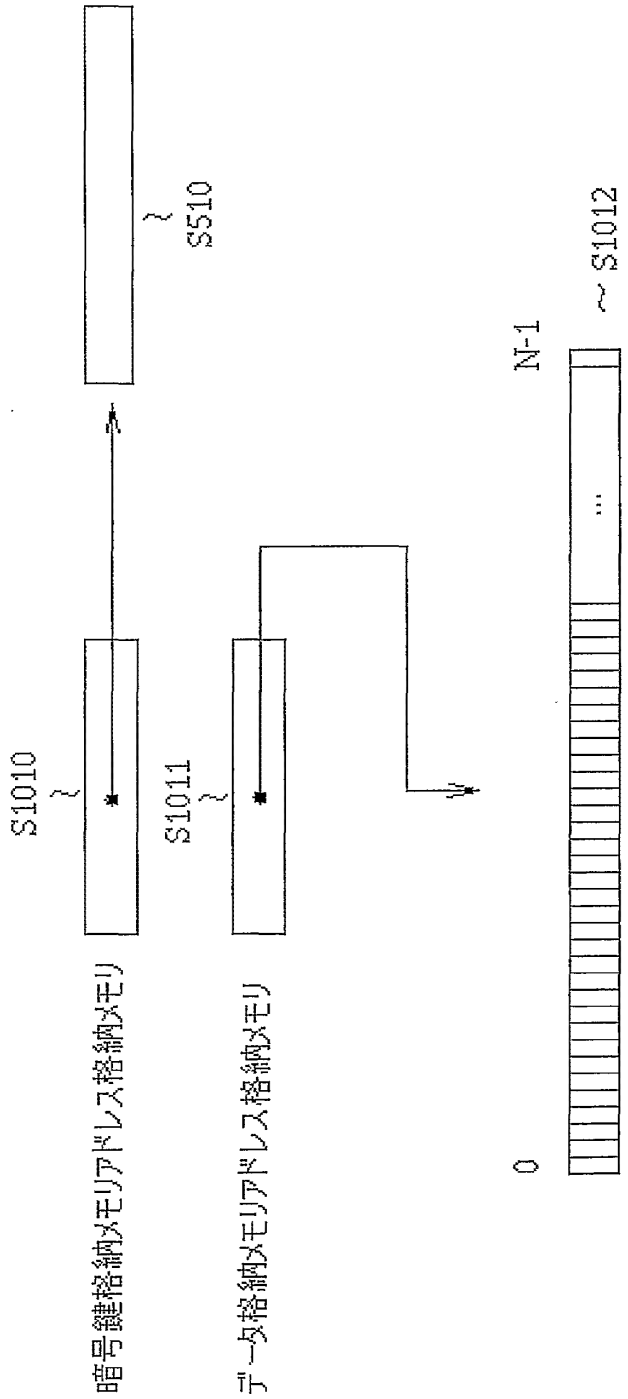
WAVELYNX EX. 1003, Pg. 932
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図9]



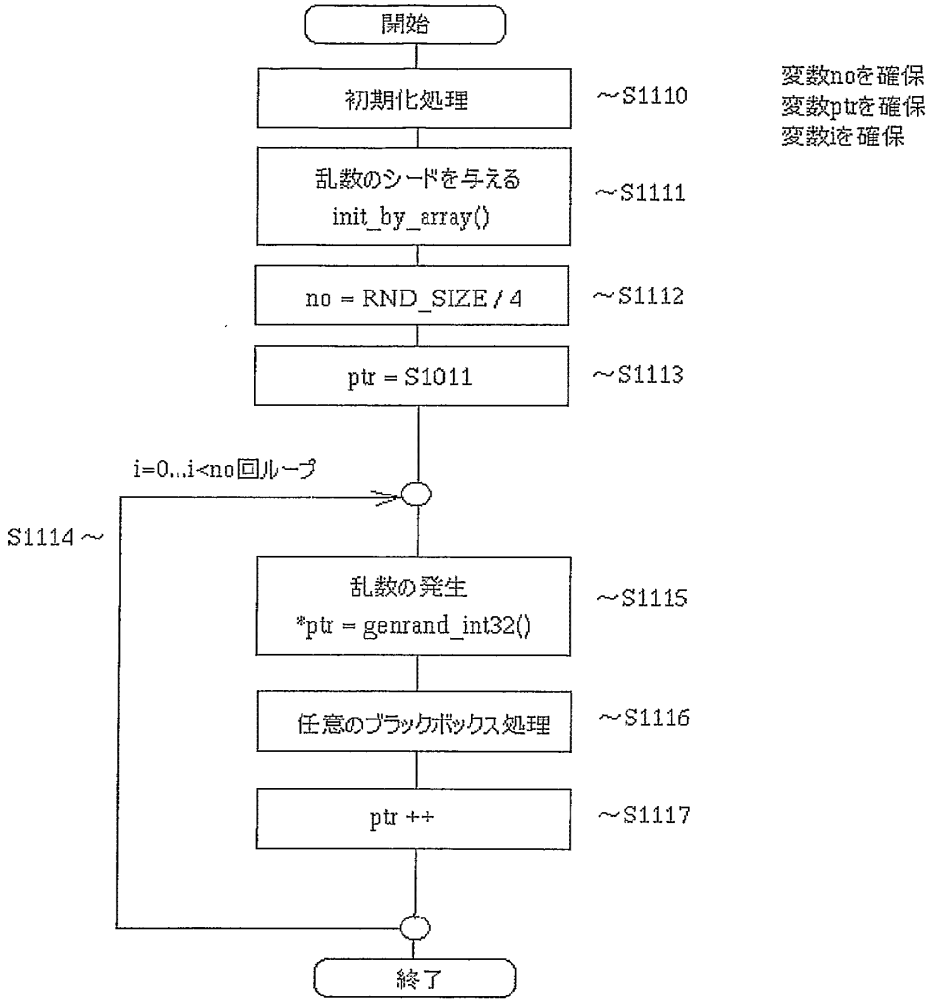
WAVELYNX EX. 1003, Pg. 933
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図10]



WAVELYNX EX. 1003, Pg. 934
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図11]



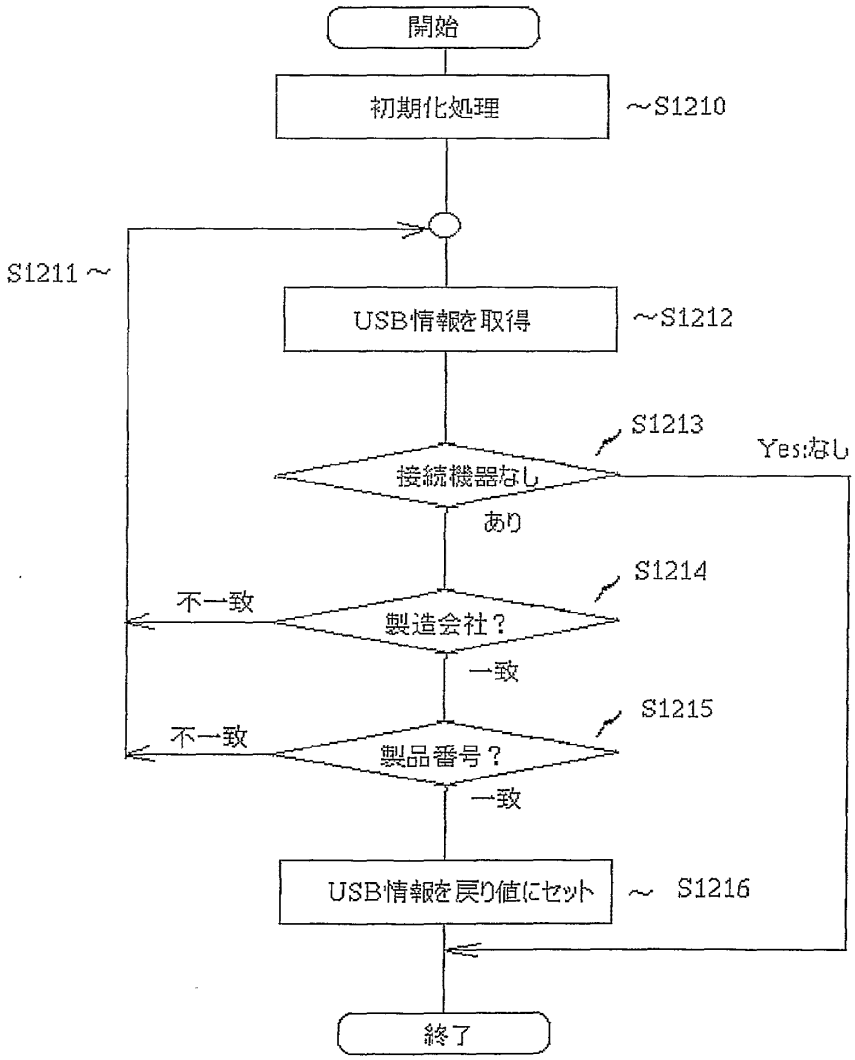
WAVELYNX EX. 1003, Pg. 935
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図12]

製造会社	製品番号	シリアルNo
0d7d	1424	A511075505900968

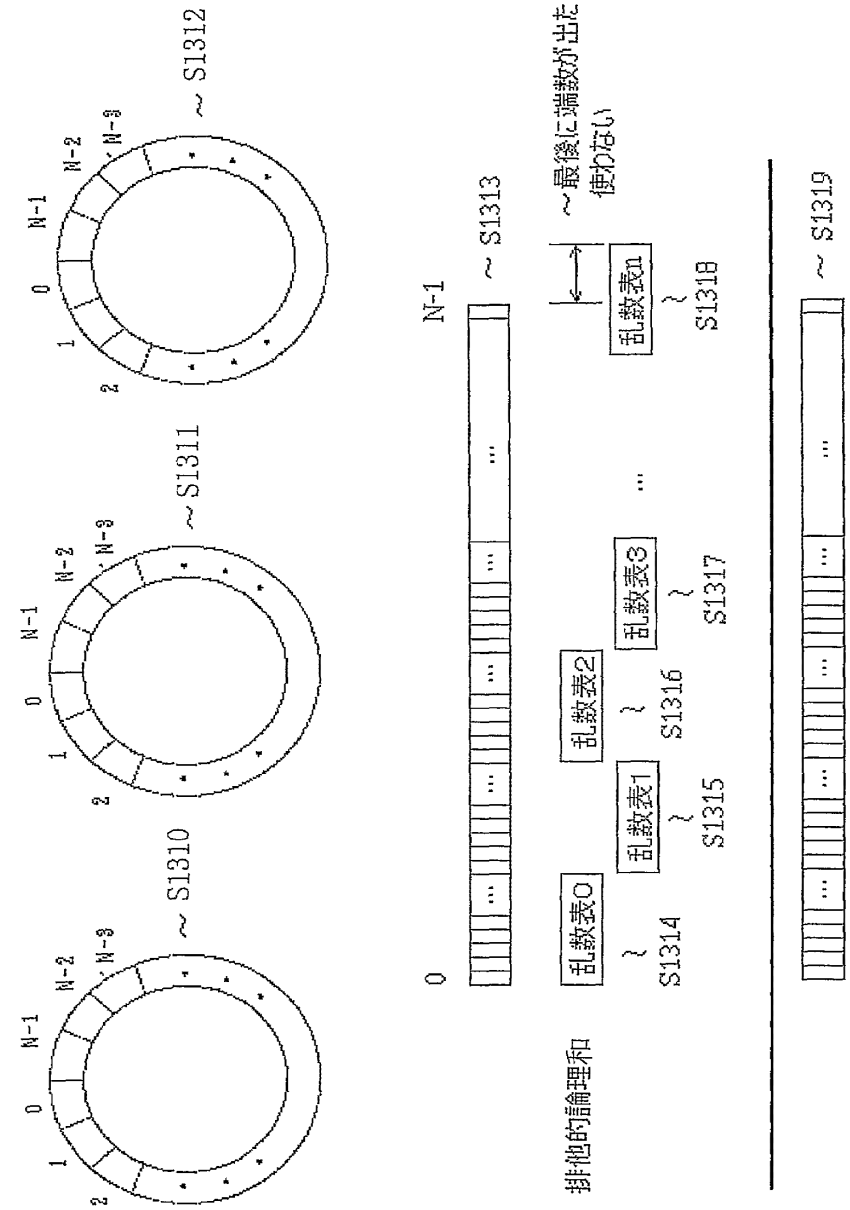
～S1220

Vid_0d7d&Pid_1424\A511075505900968 ～S1221



差替え用紙（規則26）

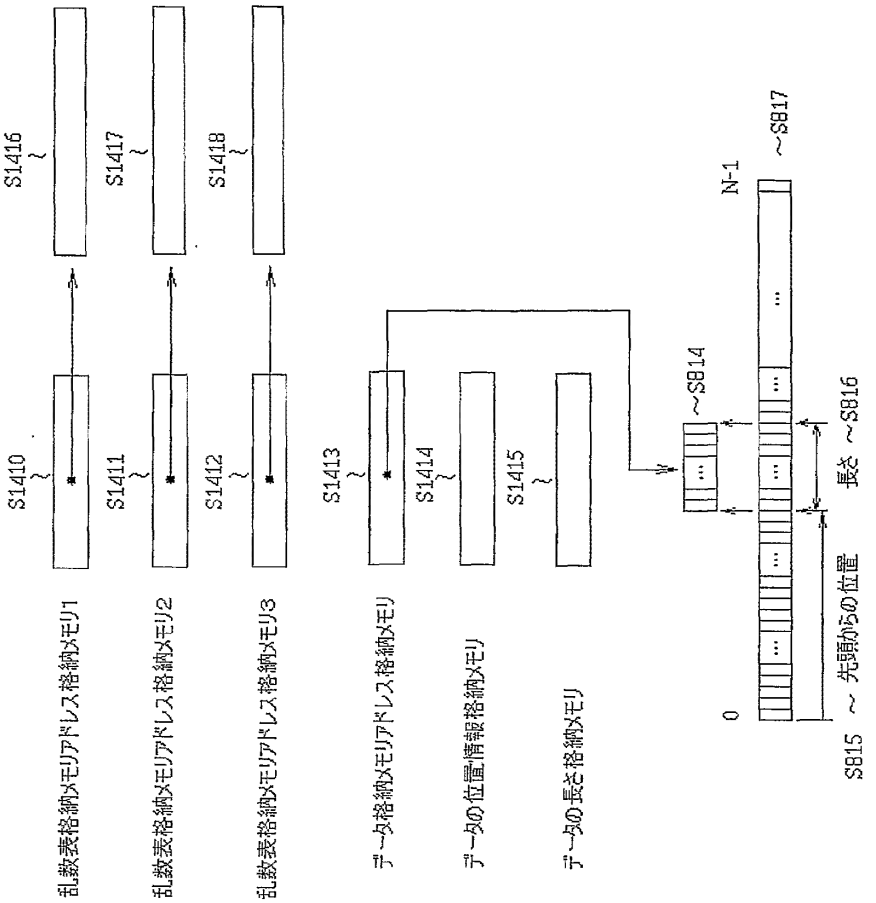
[図13]



差替え用紙(規則26)

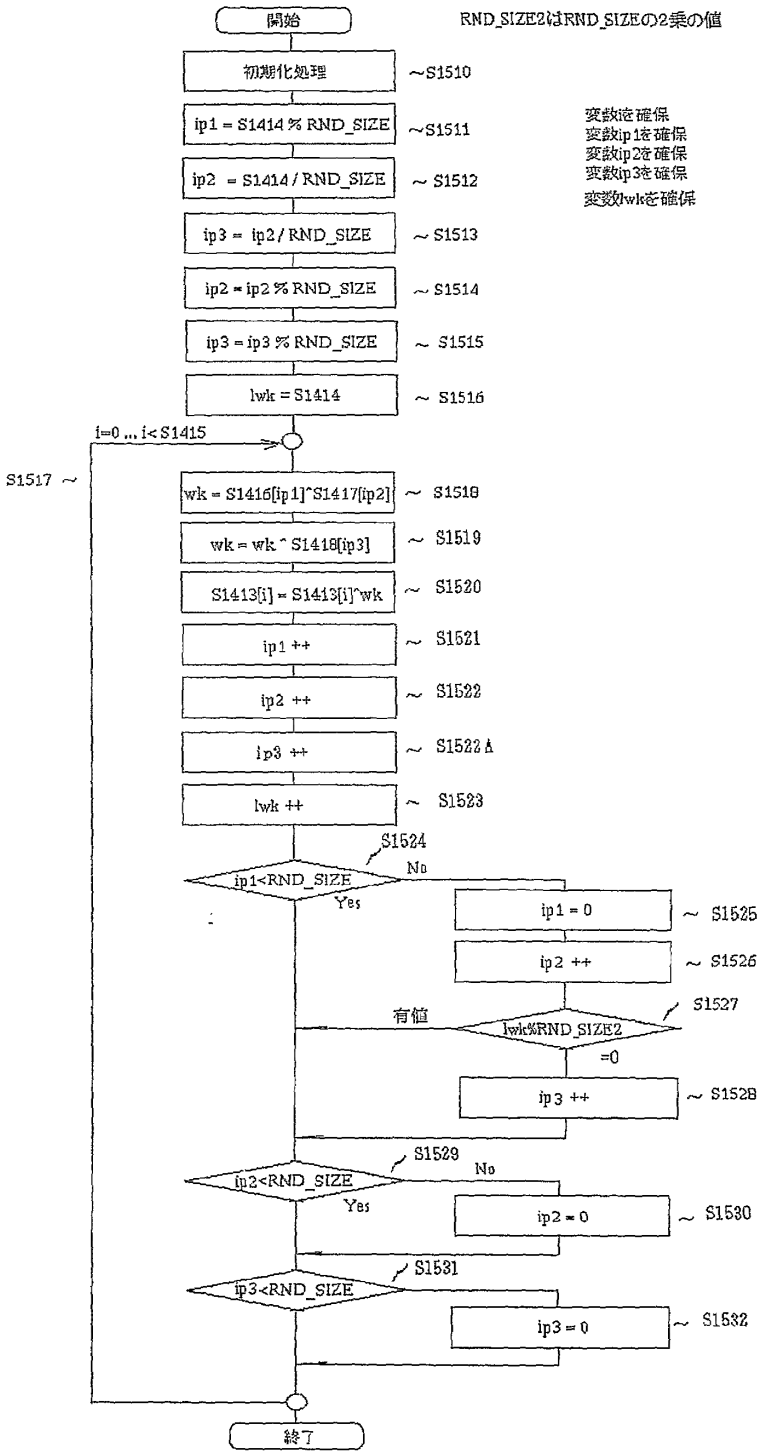
WAVELYNX EX. 1003, Pg. 937
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図14]



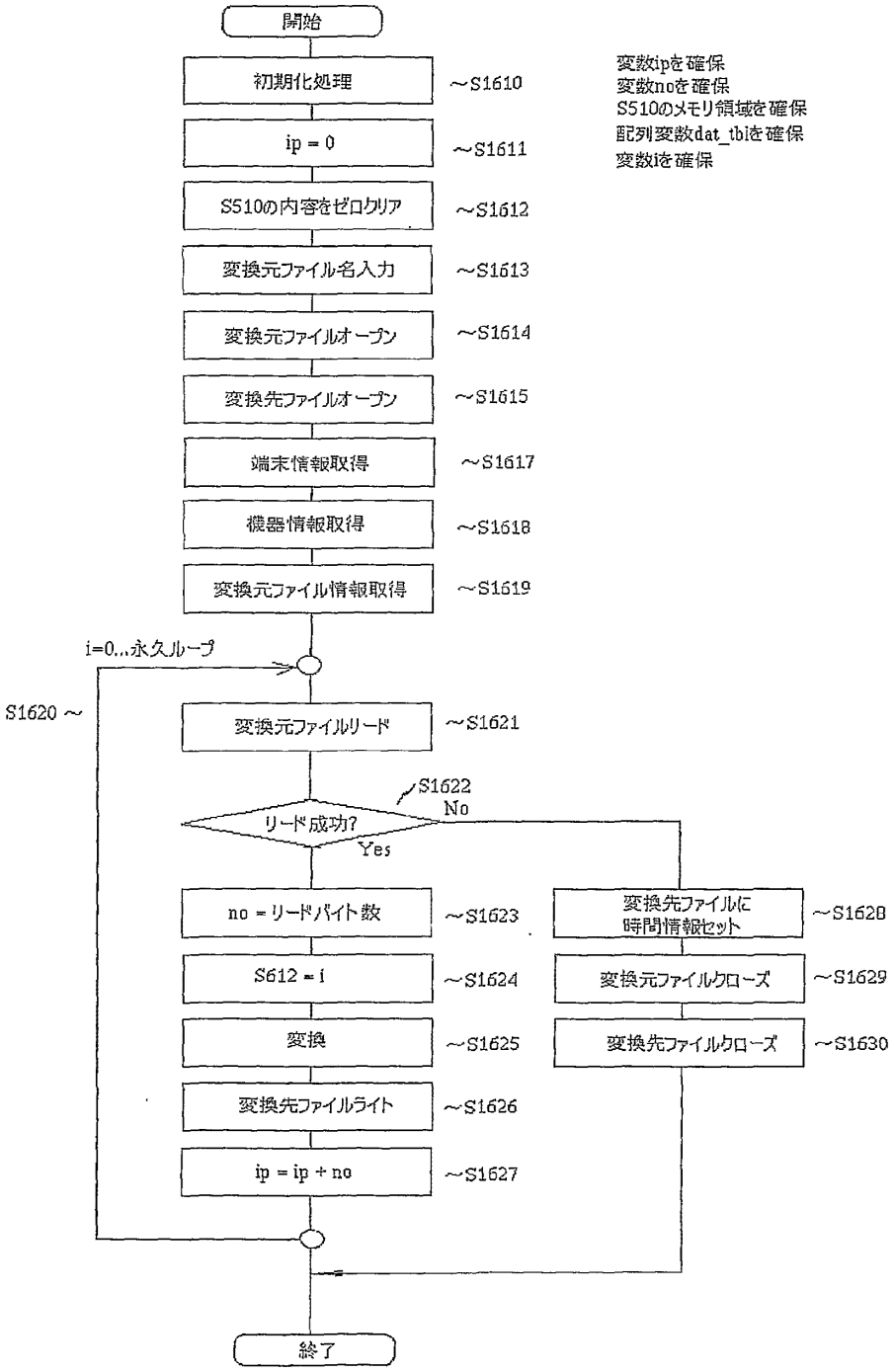
差 替 え 用 紙 (規則26)

[図15]



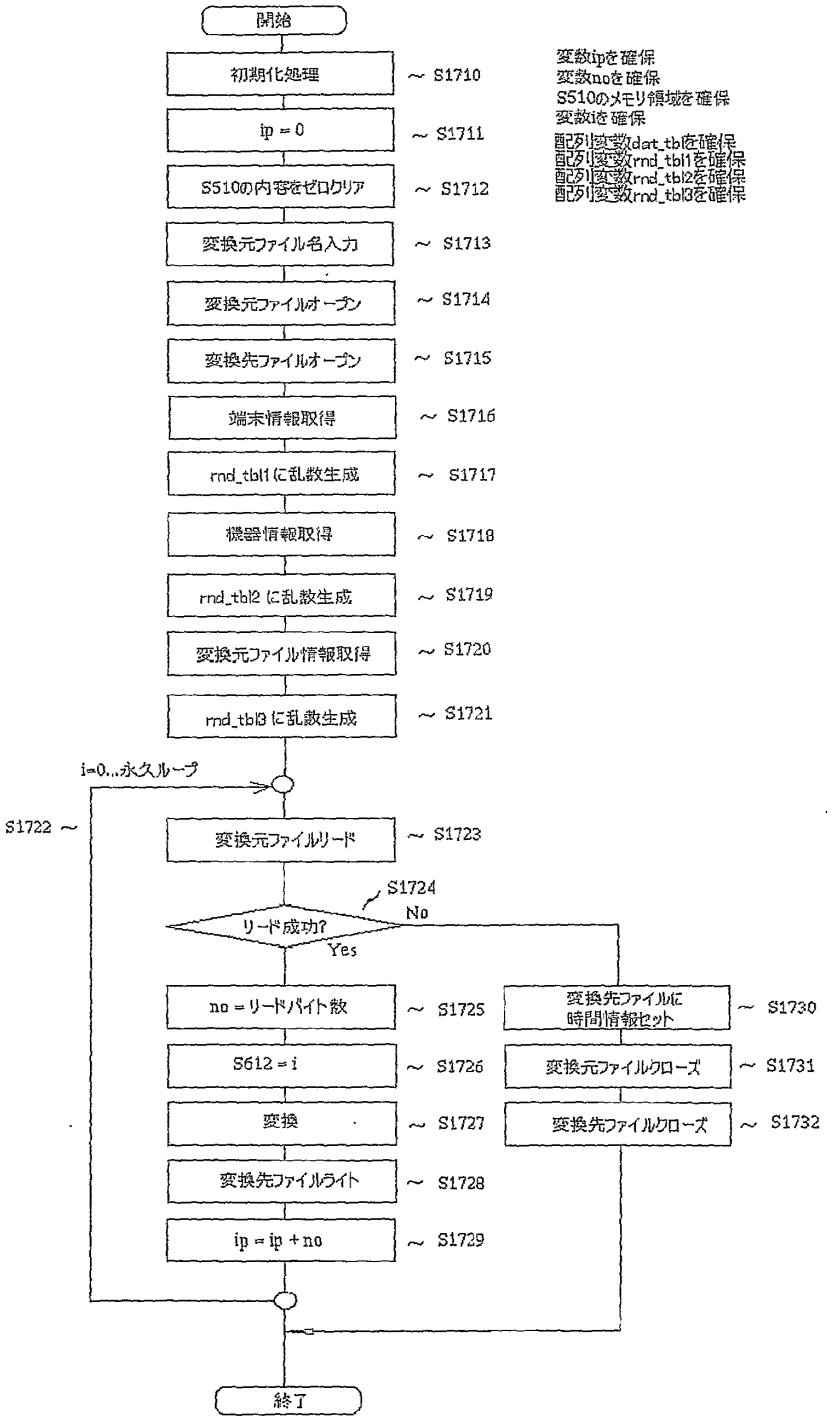
差替え用紙(規則26)

[図16]



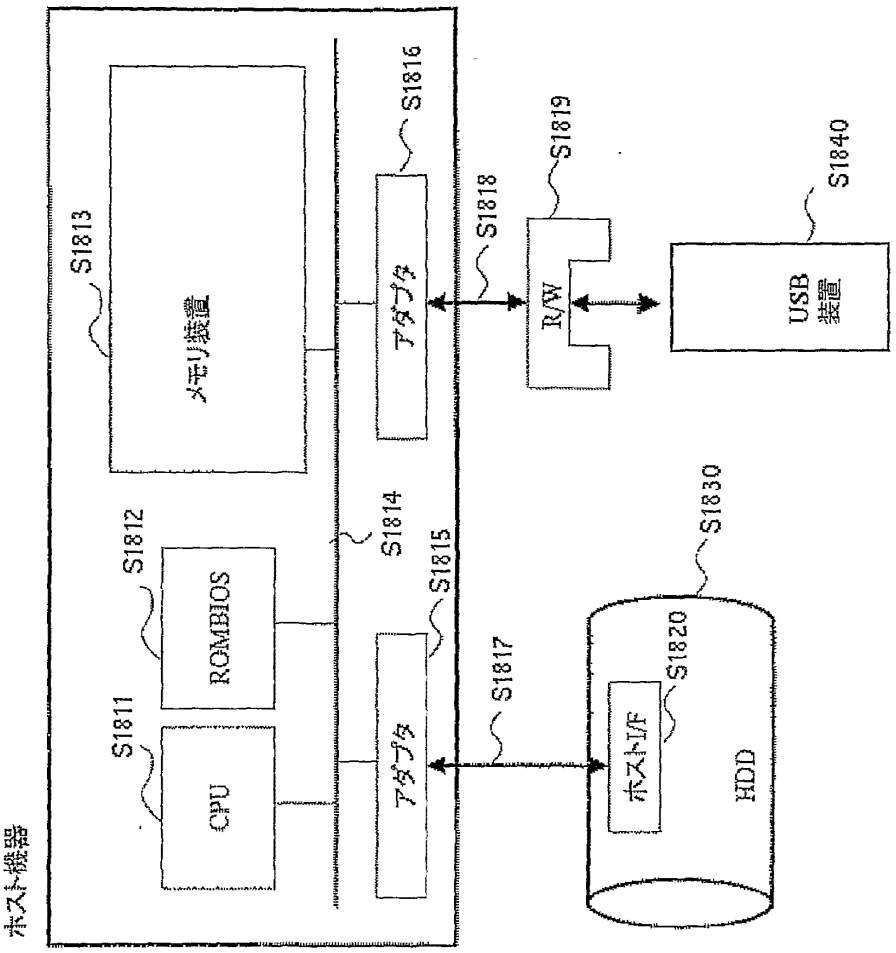
WAVELYNX EX. 1003, Pg. 940
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図17]



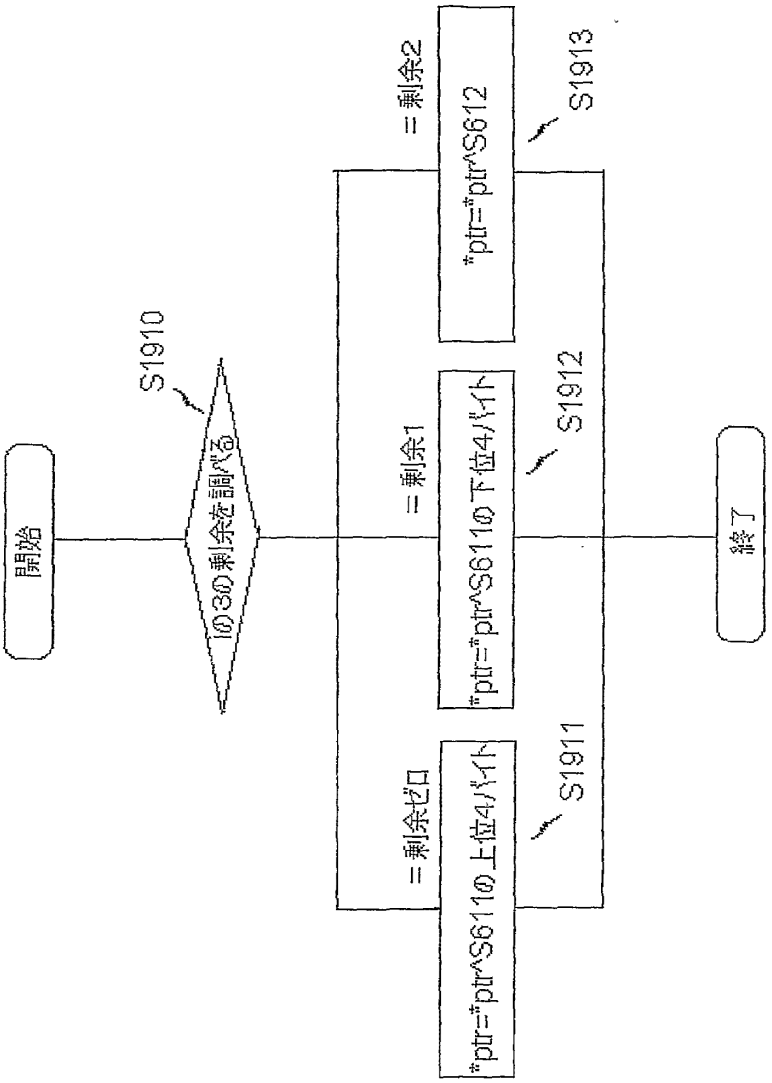
差替え用紙(規則26)

[図18]



WAVELYNX EX. 1003, Pg. 942
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[図19]



WAVELYNX EX. 1003, Pg. 943
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2007/063182

A. CLASSIFICATION OF SUBJECT MATTER
G09C1/00(2006.01)i, *G06F7/58*(2006.01)i, *H04L9/26*(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G09C1/00, G06F7/58, H04L9/26

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2007
Kokai Jitsuyo Shinan Koho	1971-2007	Toroku Jitsuyo Shinan Koho	1994-2007

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	JP 11-45173 A (Kabushiki Kaisha LE Tech), 16 February, 1999 (16.02.99), Particularly, Par. Nos. [0017] to [0027] (Family: none)	1,12,17,18 3,5
X Y	JP 64-10750 A (Mitsubishi Electric Corp.), 13 January, 1989 (13.01.89), Particularly, page 2, lower right column to page 3, upper right column (Family: none)	1,17,18 3,13,15

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
31 August, 2007 (31.08.07)	11 September, 2007 (11.09.07)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2007/063182

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CRYPTOGRAPHIC MERSENNE TWISTER AND FUBUKI STREAM/BLOCK CIPHER, Cryptology ePrint Archive: Report 2005/165. [online]. 2005.06.01 [retrieved on 2007-08-31]. Retrieved from the Internet:<URL:http://eprint.iarcr.org/2005/165/>especially 2.CRYPTMT, 3.MERSENNE TWISTER	2-18
Y	JP 2005-63399 A (Mieko TSUYUSAKI), 10 March, 2005 (10.03.05), Particularly, Par. Nos. [0043] to [0086] (Family: none)	2-18
Y	JP 8-79236 A (Mita Industrial Co., Inc.), 22 March, 1996 (22.03.96), Particularly, Par. Nos. [0029] to [0031] (Family: none)	5
Y	JP 2005-11356 A (Samsung Electronics Co., Ltd.), 13 January, 2005 (13.01.05), Particularly, Par. Nos. [0062], [0063] & US 2004/0258398 A1	16
P,Y	WO 2006/121149 A1 (Ochanomizu University), 16 November, 2006 (16.11.06), Particularly, Par. Nos. [0017] to [0033] & JP 2006-317802 A	2-18
A	Snake Oil Warning Signs:Encryption Software to Avoid. [online]. 1998.04.10 [retrieved on 2007-08-31]. Retrieved from the Internet:<URL: http://www.interhack.net/people/cmcurtin/snake-oil-faq.ps>especially 3.7 One-Time-Pads	1-18

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2007/063182

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

The inventions of claims of the present application are divided into the following two groups which do not have a common special technical feature.

1. Claims 1, 3, 12, 13, 15, 17, 18
2. Claims 2, 4-11, 14, 16

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☒ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

the

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, payment of a protest fee..

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

WAVELYNX EX. 1003, Pg. 946
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００７／０６３１８２

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00(2006.01)i, G06F7/58(2006.01)i, H04L9/26(2006.01)i

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00, G06F7/58, H04L9/26

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００７年
日本国実用新案登録公報	１９９６－２００７年
日本国登録実用新案公報	１９９４－２００７年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
X	JP 11-45173 A（株式会社エルイーテック）1999.02.16（ファミリなし）	1, 12, 17, 18
Y	特に第17-27段落	3, 5
X	JP 64-10750 A（三菱電機株式会社）1989.01.13（ファミリなし）	1, 17, 18
Y	特に第2頁右下欄-第3頁右上欄	3, 13, 15

☑ Ｃ欄の続きにも文献が列挙されている。　☐ パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー
「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの
「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「Ｏ」口頭による開示、使用、展示等に言及する文献
「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「＆」同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
３１．０８．２００７	１１．０９．２００７

国際調査機関の名称及びあて先
日本国特許庁（ＩＳＡ／ＪＰ）
郵便番号１００－８９１５
東京都千代田区霞が関三丁目４番３号

特許庁審査官（権限のある職員）　５Ｓ　９３６４
中里　裕正
電話番号　０３－３５８１－１１０１　内線　３５４６

様式ＰＣＴ／ＩＳＡ／２１０（第２ページ）（２００５年４月）

WAVELYNX EX. 1003, Pg. 947
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

Ｃ（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
Y	CRYPTOGRAPHIC MERSENNE TWISTER AND FUBUKI STREAM/BLOCK CIPHER, Cryptology ePrint Archive:Report 2005/165. [online]. 2005.06.01 [retrieved on 2007-08-31]. Retrieved from the Internet:<URL:http://eprint.iarcr.org/2005/165/> especially 2.CRYPTMT, 3.MERSENNE TWISTER	2-18
Y	JP 2005-63399 A（露崎美恵子）2005.03.10（ファミリなし） 特に第43-86段落	2-18
Y	JP 8-79236 A（三田工業株式会社）1996.03.22（ファミリなし） 特に第29-31段落	5
Y	JP 2005-11356 A（三星電子株式会社）2005.01.13 & US 2004/0258398 A1 特に第62,63段落	16
ＰＹ	WO 2006/121149 A1（国立大学法人お茶の水大学）2006.11.16 & JP 2006-317802 A　特に第17-33段落	2-18
A	Snake Oil Warning Signs:Encryption Software to Avoid. [online]. 1998.04.10 [retrieved on 2007-08-31]. Retrieved from the Internet: <URL: http://www.interhack.net/people/cmcurtin/snake-oil-faq.ps> especially 3.7 One-Time-Pads	1-18

WAVELYNX EX. 1003, Pg. 948
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００７／０６３１８２

第Ⅱ欄　請求の範囲の一部の調査ができないときの意見（第１ページの２の続き）

法第8条第3項（ＰＣＴ17条(2)(a)）の規定により、この国際調査報告は次の理由により請求の範囲の一部について作成しなかった。

１．☐ 請求の範囲＿＿＿＿＿＿＿＿＿＿ は、この国際調査機関が調査をすることを要しない対象に係るものである。
つまり、

２．☐ 請求の範囲＿＿＿＿＿＿＿＿＿＿ は、有意義な国際調査をすることができる程度まで所定の要件を満たしていない国際出願の部分に係るものである。つまり、

３．☐ 請求の範囲＿＿＿＿＿＿＿＿＿＿ は、従属請求の範囲であってＰＣＴ規則6.4(a)の第２文及び第3文の規定に従って記載されていない。

第Ⅲ欄　発明の単一性が欠如しているときの意見（第１ページの３の続き）

次に述べるようにこの国際出願に二以上の発明があるとこの国際調査機関は認めた。
本願の請求の範囲に係る発明は以下の２群に区分され、これら２群の発明は共通する特別な技術的特徴を有していない。

1.請求の範囲 1,3,12,13,15,17,18
2.請求の範囲2,4-11,14,16

１．☐ 出願人が必要な追加調査手数料をすべて期間内に納付したので、この国際調査報告は、すべての調査可能な請求の範囲について作成した。

２．☑ 追加調査手数料を要求するまでもなく、すべての調査可能な請求の範囲について調査することができたので、追加調査手数料の納付を求めなかった。

３．☐ 出願人が必要な追加調査手数料を一部のみしか期間内に納付しなかったので、この国際調査報告は、手数料の納付のあった次の請求の範囲のみについて作成した。

４．☐ 出願人が必要な追加調査手数料を期間内に納付しなかったので、この国際調査報告は、請求の範囲の最初に記載されている発明に係る次の請求の範囲について作成した。

追加調査手数料の異議の申立てに関する注意

☐ 追加調査手数料及び、該当する場合には、異議申立手数料の納付と共に、出願人から異議申立てがあった。
☐ 追加調査手数料の納付と共に出願人から異議申立てがあったが、異議申立手数料が納付命令書に示した期間内に支払われなかった。
☐ 追加調査手数料の納付を伴う異議申立てがなかった。

WAVELYNX EX. 1003, Pg. 949
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2008 年1 月24 日 (24.01.2008)

PCT

(10) 国際公開番号
WO 2008/010441 A1

(51) 国際特許分類:
H04L 9/22 (2006.01)

(21) 国際出願番号: PCT/JP2007/063797

(22) 国際出願日: 2007 年7 月11 日 (11.07.2007)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2006-199681 2006 年7 月21 日 (21.07.2006) JP

(71) 出願人 (米国を除く全ての指定国について): 日本電気株式会社 (NEC CORPORATION) [JP/JP]; 〒1088001 東京都港区芝五丁目７番１号 Tokyo (JP). 北陸日本電気ソフトウェア株式会社 (NEC SOFTWARE HOKURIKU, LTD.) [JP/JP]; 〒9202141 石川県白山市安養寺１番地 Ishikawa (JP).

(72) 発明者; および

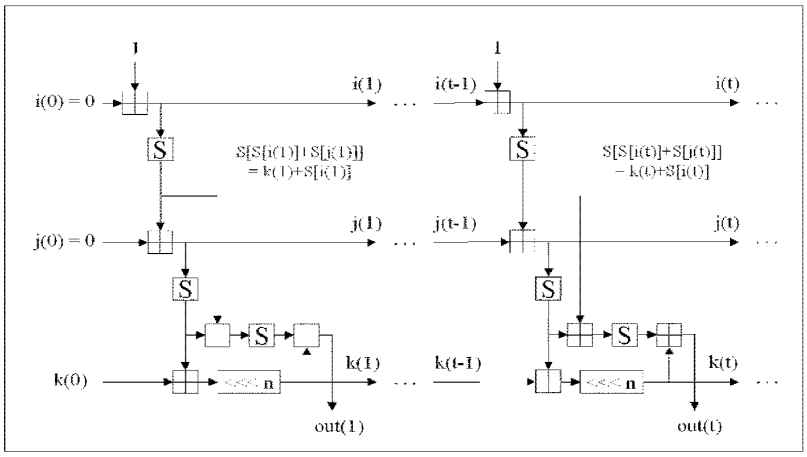
(75) 発明者/出願人 (米国についてのみ): 角尾 幸保 (TSUNOO, Yukiyasu) [JP/JP]; 〒1088001 東京都港区芝五丁目７番１号 日本電気株式会社内 Tokyo (JP). 齊藤 照夫 (SAITO, Teruo) [JP/JP]; 〒9202141 石川県白山市安養寺１番地 北陸日本電気ソフトウェア株式会社内 Ishikawa (JP). 久保 博靖 (KUBO, Hiroyasu) [JP/JP]; 〒9202141 石川県白山市安養寺１番地 北陸日本電気ソフトウェア株式会社内 Ishikawa (JP). 洲崎 智保 (SUZAKI, Tomoyasu) [JP/JP]; 〒9202141 石川県白山市安養寺１番地 北陸日本電気ソフトウェア株式会社内 Ishikawa (JP).

(74) 代理人: 加藤 朝道 (KATO, Asamichi); 〒2220033 神奈川県横浜市港北区新横浜３丁目２０番１２号 ダヴィンチ望星７階 加藤内外特許事務所 Kanagawa (JP).

/続葉有/

(54) **Title:** ENCRYPTION DEVICE, PROGRAM, AND METHOD

(54) **発明の名称:** 暗号装置及びプログラムと方法



(57) **Abstract:** Provided is an encryption device having a high safety for confidential data when performing data communication and accumulation.The encryption device generates a pseudo-random number according to a secret key and applies the pseudo-random number sequence to a plain text so as to generate an encrypted text. An internal state used for generating the pseudo-random number sequence is an internal state based on the state in accordance with rearrangement of a finite number of numeric values. At least one of the temporary variable used for generation of the pseudo-random number sequence is a temporary variable using the result of execution of a predetermined leftward or rightward rotate shift and depending on a number smaller than the number of internal state in accordance with the result of the linear or non-linear, or a combination of linear and non-linear using one or more numeric values of the internal state. The pseudo-random number generated is generated by performing calculation by using one or more numeric values among the internal states and the temporary variable.

(57) 要約: データの通信や蓄積の際にデータを秘匿するための安全性の高い暗号装置を提供する。秘密鍵に基づき擬似乱数列を生成し、前記擬似乱数列を平文に作用させることにより暗号文を生成する暗号装置であって、前記擬似乱数列の生成に用いる内

/続葉有/

WO 2008/010441 A1

(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

部状態として、有限個の数値の列の並び替えを基礎とする状態に基づく内部状態を用い、前記擬似乱数列の生成に用いる一時変数の少なくとも一つが、前記内部状態のうちの一つ、又は、複数個の数値を使用した、線形、又は、非線形、線形及び非線形の組み合わせの結果に基づき、内部状態の数より小さい数に依存し、あらかじめ定められた、左又は右ローテイトシフトを実行した結果を値とする一時変数であり、前記生成される擬似乱数を、前記内部状態のうちの１つ又は複数個の数値と、前記一時変数との演算によって生成する。

WAVELYNX EX. 1003, Pg. 951
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

明 細 書

暗号装置及びプログラムと方法

技術分野

[0001] [関連出願の記載]

本発明は、日本国特許出願:特願2006－199681号(平成18年7月21日出願)の優先権主張に基づくものであり、同出願の全記載内容は引用をもって本書に組み込み記載されているものとする。

本発明は、データの通信や蓄積の際にデータを秘匿するための暗号装置及びコンピュータ・プログラムと方法に関する。

背景技術

[0002] ＜ストリーム暗号概説＞

データを秘匿するための手法に暗号がある。暗号の中で、高速な暗号化及び復号を行うために、平文をビット単位あるいはバイト単位などで逐次、暗号化するストリーム暗号がある。典型的なストリーム暗号は、鍵ストリームを生成する鍵ストリーム生成部と、鍵ストリームと平文を結合する結合部と、からなる。例えば、暗号処理では、秘密鍵をシードとして擬似乱数を生成し(鍵ストリーム生成部)、その擬似乱数と平文をXORし(結合部)、暗号文を生成する。ここで、XORはビット毎の排他的論理和を意味する。

[0003] 平文をP、擬似乱数をR、暗号文をCとすると、

P XOR R = C

の関係となる。

[0004] 復号処理では、同じシードから擬似乱数を生成して暗号文とXORすることで、平文を得ることができる。

[0005] C XOR R = (P XOR R) XOR R

= P XOR (R XOR R)

= P XOR 0

= P

の関係となり、

WAVELYNX EX. 1003, Pg. 952
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

C XOR R = P

が得られる。

[0006] ＜ストリーム暗号の安全性＞

ストリーム暗号は、結合部がXORなど簡単な処理で実現されることが多い。そのためストリーム暗号の安全性は、鍵ストリーム生成部が生成する擬似乱数の安全性に依存する。

[0007] ここでいう、擬似乱数の安全性は、観測済みの擬似乱数列から、今後発生する擬似乱数列が予測できないことに依存している。

[0008] 例えば、鍵ストリーム生成部が、a, b, c, a, b, c, ... のように繰り返す鍵ストリームを生成したならば、観測済みの擬似乱数から、今後発生する擬似乱数の予測がつくために、暗号文が解読されてしまう。それは、結合部の逆演算を実行できるからである。

[0009] すなわち、擬似乱数Rの予測がつけば、暗号文Cを観測して、

C XOR R = P

により、平文Pを得ることができる。

[0010] 前述のように考えると、鍵ストリーム生成部が生成する擬似乱数が予測不可能な乱数(真の乱数)を生成するならば、そのストリーム暗号は安全であるといえる。逆にいえば、鍵ストリーム生成部が生成する擬似乱数が真の乱数と区別できるような情報を見つけることができれば、そのストリーム暗号は何がしかの安全性が低下していると考えることが出来る。

[0011] ＜ストリーム暗号の安全性評価手法の一例＞

前述ような考え方に基づいて、ストリーム暗号の安全性を評価する手法がある。出力された暗号文や擬似乱数列が、真の乱数列と区別がつくことを示す暗号の攻撃手法を、「Distinguishing Attack」とよぶ。

[0012] 真の乱数との区別は、出力された暗号文や擬似乱数列が、何がしかの偏りや、特徴を持っていることが示せれば、区別できたと判断する。

[0013] Distinguishing Attackでは、このような偏りや特徴を示す手段を、「Distinguisher 」と呼び、このような偏りや特徴を示す手段を発見したり作成したりすることを、「Distinguisherを構成する」と言う。Distinguisher が構成できるならば、Distinguishing Attackが

WAVELYNX EX. 1003, Pg. 953
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

適用できることになる。

[0014] ある暗号に対して、Distinguishing Attackが適用できるならば、その暗号は平文または鍵の情報を漏らしている可能性があるので、安全な暗号であると保証することが出来ない。

[0015] 従って、Distinguishing Attackが適用できる暗号に修正を加え、Distinguishing Attackが適用出来なくなれば、暗号の安全性が向上したと考えることが出来る。

[0016] ＜ストリーム暗号の具体例＞

RC4は、Ron Rivest氏によって開発された暗号方式であり、RFC2246(TLS), WEP, WPA など暗号標準としてよく利用されるストリーム暗号である。RC4の仕様は、RFC2246(TLS)などで公開されていると言ってよい。

[0017] RC4は、処理単位nビットが可変という特徴を持ちながらも、処理単位nを大きくするとメモリが2n必要となり、鍵スケジュールも極端に低速になる。

[0018] そのため、現実的に、nは8ビットを越える実装は少なく、32ビットを越える実装は不可能であった。

[0019] よって、RC4は、32bit/64bitプロセッサといった最近のプロセッサ・サイズに適した実装が出来なかった。

[0020] 32ビットRC4ではこれらの制限が生じないよう、32bit/64bitプロセッサで高速かつ小メモリで実装可能なアルゴリズムに改良された。2005年にG. Gongらが発表した論文（非特許文献1）である。

[0021] 非特許文献1の論文では、処理単位が32bitの場合、速度は、RC4の約3.1倍高速、メモリはRC4の約2^{-22}倍に抑えることに成功している。

[0022] また、内部変数kを追加することにより、過去に報告されたRC4の脆弱性（統計的な偏り）が生じないアルゴリズムに改良されている。

[0023] ＜発明が対象とする攻撃手法の要点＞

G. Gongらのアルゴリズムの場合、連続する出力の最下位ビットが必ず一致するというdistinguisherの構成が可能となる。

[0024] このdistinguisherにより、約2^{30}のデータ量で真の乱数系列と区別可能となる。

[0025] ＜攻撃手法の例示のための、G. Gongら提案した改良アルゴリズムの説明＞

WAVELYNX EX. 1003, Pg. 954
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

図2は、非特許文献1において、G. Gongらによって提案されたRC4の改良アルゴリズム(32bit RC4)を示す図である。G. Gongらによって提案されたRC4型ストリーム暗号は、配列Sのエントリ数が2^{n}個、配列Sのエントリサイズがmビットである。

[0026] また、非特許文献1の提案では、n=8, m=32のモデルについてのみKSAの初期定数aiを定義しているので、本明細書における解析でも、n=8, m=32のモデルについて、詳しく解析を行う。以下では、基本処理単位n, mを元に便宜上GGHN(n,m)と記述する。

[0027] 図2に示されるように、G. Gongらによって提案されたRC4型ストリーム暗号GGHN(n, m)は、KSA(K,S)とPRGA(S)の二つの処理から構成される。

[0028] KSA(K,S)は、いわゆる初期設定であり、40ビットから256ビットの鍵Kを基にして、32ビット256個の配列を並べ替え、初期状態Sを作り出す処理を行う。

[0029] PRGA(S)は、鍵ストーリームを生成する処理であり、状態Sに基づいて毎時刻、擬似乱数を生成する。

[0030] ここで、+ はmod N上またはmod M上の算術加算を表し、N=2^{8}、M=2^{32}である。また L は秘密鍵のバイト数を表している。

[0031] まず、KSA(K,S)の動作を説明する。

[0032] KSAでは、配列Sの初期値として、初期変数aiを代入し(S[i]=a_i)、Sエントリどうしのswap(Swap[S[i],S[j]])と算術加算(S[i]＝S[i]＋S[j] mod M)を繰り返すことで、Sエントリを攪拌している。

[0033] KSAにおいて、内部変数kもSエントリ(k＝k＋S[i] mod M)を使って初期化されるため、PRGAにおけるkの初期値は未知である。

[0034] Sエントリの攪拌において、ループ回数rが可変であるが、Sエントリの出現確率がランダムになるように、m=32のときr=20を選択するよう定められている。G. Gongらによる提案では、m=64のときは、r=40を設定するよう定められている。

[0035] KSA(K,S)が終了した直後でPRGA(S)が開始されていない状態を時刻t＝0とする。時刻t＝0のとき、KSA(K,S)の動作が終了し、秘密鍵Kによって配列Sの状態が十分攪拌されていることが期待されている。

[0036] 次に、PRGA(S)の動作を説明する。

WAVELYNX EX. 1003, Pg. 955
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0037] PRGAでは、インデックスi、jに基づく配列Sの参照結果(S[(S[i]+S[j])mod N])と、変数kの算術加算を行って、1ワード(1ワード=32ビット)を鍵ストリーム(out=S[(S[i]+S[j]) mod N]+k) mod M)として出力する。また鍵ストリームを生成するために参照したSエントリ(S[(S[i]+S[j])mod N])を、鍵ストリーム出力直後にkを用いて更新している(S[(S[i]+S[j])mod N]=k+S[i] mod M)。図2における out(=S[(S[i]+S[j])mod N]=k+S[i] mod M)が出力となる鍵ストリームである。

[0038] 図3では、時刻t＝1における動作(PRGAの状態遷移)を示している。配列Sにおいて、アドレス1の値S[1]がA、アドレスAの値S[A]がBの時に、アドレスA+Bの値S[A＋B]は、k0+A+Bになる。

[0039] 図4では、時刻t＝2における動作(PRGAの状態遷移)を示している。配列Sにおいて、アドレス1の値S[1]がA、アドレスAの値S[A]がB、アドレス2の値S[2]がC、アドレスA+Cの値S[A+C]がD、アドレスA+Bの値S[A+B]がk0+A+B、の時に、アドレスC+Dの値S[C+D]が、k0+B+C+Dになる。

[0040] G. Gongらによって提案されたRC4型ストリーム暗号の安全性は、彼らの提案論文(非特許文献1)のなかでで報告されている。

[0041] それによれば、鍵ストリームはSエントリに変数kを算術加算することでマスクされるため、KSAによってkが一様分布に従うとするならば出力系列に偏りが生じないことを示している。

[0042] また内部メモリのサイズはRC4の4倍となり、Sエントリが算術加算によって更新されるため、内部メモリを求める攻撃に対しても安全性が向上していると報告されている。

[0043] ただし、全てのS-boxエントリ(配列Sの要素)と変数kが同時に偶数になると、以降常に偶数が続くという「weak state」が存在する。しかし、内部メモリのサイズからみて、weak stateの存在確率は起こりえないくらいに十分小さいので、安全性に問題がないとされている。

[0044] 非特許文献1:G. Gong, K.C. Gupta, M. Hell, and Y. Nawaz,"Towards a General RC4-Like Keystream Generator,"SKLOIS Conference on Information Security and Cryptology, CISC 2005,LNCS 3822, pp.162-174, Springer Verlag, 2005.

非特許文献2:I. Mantin, and A. Shamir: "A Practical Attack on Broadcast RC4,"Fas

WAVELYNX EX. 1003, Pg. 956
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

t Software Encryption, FSE 2001, LNCS 2355, pp.152-164,Springer-Verlag, 2001

非特許文献3:S. Paul, B. Preneel, and G. Sekar: "Distinguishing Attacks on the StreamCipher Py," eSTREAM, the ECRYPT Stream Cipher Project, Report 2005/081, 2005.

発明の開示

発明が解決しようとする課題

[0045] 非特許文献1～3の開示事項は、本書に引用をもって繰り込み記載されているものとする。以下の分析は、本発明によって与えられたものである。

[0046] ＜発明が対象とするGGHN(n,m)攻撃手法 Distinguishing Attackの説明＞

GGHN(8,32)を解析するにあたり、変数の表記や定義について説明する。

[0047] ・は、算術乗算を表す。

‖ は、データの連結(concatenation)を表す。

X <<< nは、データXの左nビットローテイトを表す。

またlsb(X)はデータXの最下位ビット、LSB(X)はデータXの最下位バイトとし、

lsb(X) = X mod 2

LSB(X) = X mod 2^{8}

となる。

時刻tにおける変数i, j, kをit, jt, ktと表記する。

また時刻tにおけるx番目のS-boxエントリをSt[x]と表記する。

時刻tで出力される鍵ストリームをOtとし、最初の鍵ストリームを出力する時刻をt=1とする。

[0048] ここで、PRGAの初期値をi0 = 0, j0 = 0と定義し、k0は未知となる。

[0049] また解析を行う上で、攻撃者は鍵ストリームを自由に入手可能であるものとする。

[0050] ＜1番目の出力ワードと2番目の出力ワード間の偏り＞

まずGGHN(8,32)の1番目の出力ワードと2番目の出力ワード間に偏りが生じることを説明するため、以下の＜ケース1＞の条件が成り立つ場合について考察する。

[0051] ＜ケース1＞

1. LSB(S1[i1] + S1[j1]) = LSB(S1[i1])、ただし、LSB(S1[i1]) ≠ 1

2. LSB(S2[i2] + S2[j2]) = i2

[0052] ケース1の時、t=1、2における配列Sの最下位バイトの状態遷移について図5及び図6に示す。図5には、配列Sにおいて、アドレス1の値S[1]がAであり、アドレスAの値S[A]がk0+Aであるべきところが0となり、矛盾することが示されている。図6には、配列Sにおいて、アドレス1の値S[1]がA、アドレスAの値S[A]がk0+A、アドレスA＋Cの値S[A+C]が2−C、アドレス2の値S[2]がk0+2であるべきところが、Cとなり、矛盾することが示されている。

[0053] 図2より、t=1のとき、i1=1であり、LSB(S1[i1]) = Aとおくと、j1 = Aとなる。

[0054] ここで、ケース1の条件1を満たす時、

LSB(S1[1] + S1[A]) = LSB(S1[1])

LSB(S1[A]) = 0 ・・・ (1)

となるため、

LSB(k1) = LSB(k0 + S1[j1]) = LSB(k0)

となる。

[0055] ただし、A = 1のとき、

LSB(S1[1] + S1[1]) = LSB(S1[1])

LSB(S1[1]) = 0 ≠ 1

となり、式(1)と矛盾するため、

LSB(S1[i1]) ≠ 1の条件が導かれる。

[0056] t=1で出力される鍵ストリームにおいて以下の関係が成り立つ。

LSB(O1) = LSB(k0) ・・・ (2)

[0057] 同様にt=2のとき、i2=2であり、LSB(S2[i2]) = Cとおくと、j2 = A + Cとなる。

[0058] ここで、ケース1の条件2を満たす時、

LSB(S2[2] + S2[A+C]) = 2

LSB(S2[A+C]) = 2 − C

となる。

[0059] t=2で出力される鍵ストリームにおいて、以下の関係が成り立つ。

LSB(O2) = LSB(k0 + 2) ・・・ (3)

[0060] よって、式(2)、(3)より、1番目と2番目の出力ワードO1、O2について以下の関係式が必ず成り立つ。

[0061] lsb(O1) = lsb(O2) ・・・ (4)

[0062] また、ケース2についても同様に考える。

[0063] ＜ケース2＞

1. LSB(S1[i1] + S1[j1]) = LSB(S1[i1])、ただし、LSB(S1[i1]) ≠ 1

2. LSB(S2[i2] + S2[j2]) = j2

[0064] ケース2の場合、t=1、2における配列Sの最下位バイトの状態遷移について、図5及び図7に示す。図7には、アドレス1の値S[1]がA、アドレス2の値S[2]がC、アドレスAの値S[A]がk0+A、アドレスA＋Cの値S[A+C]がk0+A＋Cであるべきところが Aとなり、矛盾することが示されている。

[0065] t=1における内部変数kはケース1と同じため、配列Sの状態遷移や鍵ストリームにおける関係式(2)も同じとなる。

[0066] t=2のときケース2の条件2を満たす時、

LSB(S2[2] + S2[A+C]) = A + C

LSB(S2[A+C]) = A

となる。

[0067] t=2で出力される鍵ストリームにおいて以下の関係が成り立つ。

[0068] LSB(O2) = LSB(k0 + 2･S1[1]) ・・・ (5)

[0069] よって、ケース2の条件1、2を満たす時も式(2)、(5)より、式(4)が必ず成り立つ。

[0070] このように、ケース1、2ともに、1番目の出力ワードO1と2番目の出力ワードO2間で同じ関係式(4)が成り立つ。

[0071] 次に、この式がdistinguisherとして利用できることを説明する。

[0072] ＜Distinguisherが成り立つ確率と必要なデータ量＞

ここでは、distinguisherとして利用する式(4)の成立確率について説明する。

[0073] もしGGHN(8,32)の出力する系列が真の乱数系列であれば、distinguisherである式(1)が成り立つ確率は2^{-1}となる。

[0074] 式(4)の成立確率は、PRGAの構造に依存したものであり、KSAの構造には依存しな

い。

[0075] よって、以下の検討では、KSA終了後の変数kと配列Sがそれぞれ独立で一様分布に従うものとする。

[0076] まず、ケース1、2の条件1、2が成り立つ確率p1、p2は以下のとおりである。ここで条件2が成り立つ確率p2はケース1、2を考慮した確率である。

[0077] p1 = 1/256・255/256

p2 = 1/256・1/256 + 255/256・2/256

[0078] ここで条件1、2のいずれかの条件を満たさない時、式(1)が成り立つ確率を、理想的に1/2と仮定すると、GGHN(8,32)の出力系列において、式(4)が成り立つ確率Pdは、以下で与えられる。

[0079] pd = 1・p1・p2+1/2・(1−p1・p2)

≒ 1/2・(1+2^{−15.01})

[0080] よって、真の乱数系列における確率1/2と比べて大きくなる。

[0081] 次に、式(4)をdistinguisherとしたとき、GGHN(8,32)の出力系列と真の乱数系列を区別するために必要なデータ量を検討する。

[0082] 非特許文献2によると、2つの分布を区別するのに必要なデータ量は、以下のようになることが示されている。

[0083] 確率pで発生する事象分布Xと、確率p(q+1)で発生する事象分布Yにおいてあるイベントeが生じる時、無視できない成功確率でXとYを区別するには、O(1/pq^{2})のサンプルが必要である。

[0084] ただし、上記定理は、p<<1のときに成り立つ。

[0085] 非特許文献3には、p=1/2のとき、2つの分布を区別するのに必要なデータ量は以下のようになることが示されている。

[0086] 確率p = 1/2で発生する事象分布Xと確率1/2(q+1)で発生する事象分布Yにおいてあるイベントeが生じる時、無視できない成功確率でXとYを区別するにはO(1/q^{2})のサンプルが必要である。

[0087] 本解読におけるイベントeは、式(4)が成り立つ事象であり、乱数におけるイベントeの分布をX、GGHN(8,32)の出力系列におけるイベントeの分布をYとみなすことができる

WAVELYNX EX. 1003, Pg. 960
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

。

[0088] よって、p = 2^{-1}、q = 2^{-15.01}と考えることができるため、解読に必要なデータ量はO(2^{30.02})となる。

[0089] ここで求まるデータ量は、GGHN(8,32)のKSAが完全にランダムなpermutationという仮定の元での値であり、PRGAの構造的な偏りから求まる理論的なデータ量である。

[0090] よって、GGHN(8,32)は、理論的に約2^{30}の秘密鍵における鍵ストリームの先頭2ワードを利用することで、真の乱数列との区別が可能である。

[0091] ＜1番目の出力ワードと2番目の出力ワード間の偏り＞の説明（段落0050から0070）では、鍵ストリームの先頭2ワードについてdistinguisherの構成方法を説明したが、ケース1については任意の時刻tにおいて連続する鍵ストリーム2ワードについて同様の関係が成り立つ。

[0092] よって、鍵ストリームの先頭数ワードを捨てるという対策は効果が無い。

[0093] 前述した＜発明が対象とするGGHN(n,m)攻撃手法 Distinguishing Attackの説明＞は、図5乃至図9に、まとめることができる。

[0094] 図9は、解読に必要なデータ量（理論値）をあらわしており、初期処理によってS-boxエントリが一様にランダムで出力均等と仮定した場合の、解読に必要なデータ量の理論値を求める流れを説明する図である。

[0095] 図9では、初期処理によってS-boxエントリが一様にランダムで出力均等と仮定している。これは、攻撃する時刻において、Sの値256通りがすべて出現する可能性があるということである。図9における、確率p1(=1/256)、確率p2(=255/256)は、図5における、条件1が発生する確率、と、条件2が発生する確率にそれぞれ対応する。また、図9における、確率p3 (=(1/256)・(1/256)+(255/256)(2/256)=512/256^2)は、図7における条件3、または、図6における条件4が発生する確率に対応する。解読に必要なデータ量（理論値）は、O(q^{-2})=O($2^{30.02}$)となる。

[0096] 図8に示すように、G. Gongら提案した改良アルゴリズムの distinguisher を構成することが出来る。図8は、

lsb(O1) = lsb(O2)

の式を、distinguisherとすることを説明する図である。

[0097] 連続する出力の下位8bitは以下のようになる。

O1=k0

O2=k0＋2A （条件3）、

O2＝k0+2 （条件4）

[0098] よって、式(4)をdistinguisherとすれば、GGHN(8,32)の出力系列と真の乱数系列との区別が可能となる。

[0099] 本発明者等は、このことを、確めるための実験を行ったので、以下に説明する。図10と図11は、実験の結果をまとめたものである。図10には、計算機実験によって得られた確率と解読に必要なデータ量が示されている。図11には、計算機実験によって、与えるデータ量Nを変化させながら、図8のdistinguisherが機能するか確かめた結果が示されている。

[0100] すなわち、図11では、与えるデータ量Nを変化させながら、図8のdistinguisherについて成立回数Xを求めた。100通りの秘密鍵について実験を行い棄却率を求めた。

$X - 2^{N-1} > (1/2)\cdot\sqrt{(2^{N-1} - 2^{N-2})}$

を満たすとき、乱数でないと棄却する(乱数であれば30.5%となる)。以下に、該実験の説明をする。

[0101] ＜実験結果の説明＞

式(4)をdistinguisherとしたとき、図9のように、GGHN(8,32)の出力系列と真の乱数系列が区別可能であるかどうか確認した。実験の手順は以下のとおりである。

[0102] 1. 秘密鍵を2^{w}回ランダムに変更し、GGHN(8,32)の鍵ストリームをそれぞれ2ワード生成する。

[0103] 2. 1. で生成した2^{w}個の鍵ストリームについて式(1)の成立回数をカウントする。

[0104] 3. 2. でカウントされた回数xが以下の関係式を満たした時、出力系列が乱数でないと棄却する。ここでμは平均値、σは標準偏差を表す。

[0105] $\mu - x > \sigma/2$

[0106] よって本実験においては、

$2^{w-1} - x > 1/2\cdot(2^{w-1} - 2^{w-2})^{-1/2}$

の関係式を満たした時、出力系列が乱数でないとして棄却する。

WAVELYNX EX. 1003, Pg. 962
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0107] 4. 1. で与えた2^{w}個の秘密鍵グループを独立に100通り与え、1～3を繰り返し、棄却率を求める。

[0108] 実験結果を表として示した図11によると、2^{28}のデータを与えたとき、棄却率が85%となり、乱数における棄却率と比べて、50%以上のアドバンテージが得られた。

[0109] よって、式(4)を、distinguisherとしたときの攻撃手法によって、GGHN(8,32)の出力系列は、約2^{30}ワードの鍵ストリームを用いることによって、極めて高い確率で真の乱数列との区別が可能である事が実験的にも確認できた。

[0110] このように、従来のGGHN(8,32)の出力系列は、式(4)

{lsb(O1) = lsb(O2)}

を、distinguisherとしたときの攻撃手法によって、鍵ストリームが高い確率で真の乱数列との区別が可能であり、安全性が低いという課題がある。

[0111] 本発明は、本発明者等による上記課題の認識に基づき創案されたものであって、その目的は、データを秘匿するための安全性の高い暗号装置、プログラム、方法を提供することにある。

課題を解決するための手段

[0112] 本願で開示される発明は、前記課題を解決するため、概略以下の構成とされる。

[0113] 本発明は、上記課題となった解析手法に、耐性を持つ対策を提供する。また、対策の実装において、暗号の設計者らの主張する安全性と実装性を損なうことのないよう考慮している。

[0114] 本発明の1つのアスペクト(側面)に係る暗号装置は、秘密鍵に基づき擬似乱数列を生成し、前記擬似乱数列を平文に作用させることにより暗号文を生成する暗号装置であって、前記擬似乱数列の生成に用いる内部状態として、有限個の数値の列の並び替えを基礎とする状態に基づく内部状態を用い、前記擬似乱数列の生成に用いる一時変数の少なくとも一つが、前記内部状態のうちの一つ、又は、複数個の数値を使用した、線形、又は、非線形、線形及び非線形の組み合わせの結果に基づき、内部状態の数より小さい数に依存し、あらかじめ定められた、左又は右ローテイトシフトを実行した結果を値とする一時変数であり、前記生成される擬似乱数を、前記内部状態のうちの1つ又は複数個の数値と、前記一時変数との演算によって生成する。

WAVELYNX EX. 1003, Pg. 963
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0115] 本発明において、前記有限個の数値の列の並び替えを基礎とする状態に基づく内部状態であって、前記内部状態の更新を、並び替え以外の、線形演算、及び、非線形演算を用いて行うようにしてもよい。

[0116] 本発明において、前記有限個の数値の列の並び替えを基礎とする状態に基づく内部状態であって、前記内部状態の更新として、並び替え以外の、線形演算及び非線形演算を用いることにより、前記内部状態の状態数が、単調増加する、ようにしてもよい。あるいは、前記内部状態の状態数が、単調減少する、ようにしてもよい。

[0117] 本発明において、前記有限個の数値の列の並び替えを基礎とする状態に基づく内部状態であって、前記内部状態の更新として、並び替え以外の線形演算及び非線形演算を用いることにより、内部状態の状態数を振動させる、ようにしてもよい。

[0118] 本発明においては、前記内部状態の更新を、前記擬似乱数列の出力ごとに行うようにしてもよい。あるいは、前記擬似乱数列の出力よりも多く行うようにしてもよい。あるいは、前記擬似乱数列の出力よりも少なく行うようにしてもよい。

[0119] 本発明においては、前記ローテイトシフトの方向及び／又は数値（シフト数）を、前記内部状態の数値に依存して動的に変更するようにしてもよい。

[0120] 本発明においては、前記ローテイトシフトの方向及び数値（シフト数）を、あらかじめ定められたテーブルの値に従って変更するようにしてもよい。

[0121] 本発明の別のアスペクトに係る装置は、配列Sの要素の並び替えと算術加算を繰り返すことで配列Sの要素を攪拌し配列Sの初期状態を作成し、その際、内部変数kの初期値を配列Sの要素から求める第1の処理部（KSA）と、第1、第2のインデックス変数iとjに関して配列Sの要素S[j]と内部変数kの加算結果をローテイトシフトした値で内部変数kの値を更新し、S[i]＋S[j]による配設Sの参照結果S[(S[i]＋S[j])]と内部変数kとの加算結果に基づき、鍵ストリームを出力し、前記鍵ストリームを生成するために参照した配列Sの要素S[(S[i]＋S[j])]を、鍵ストリーム出力の直後に内部変数kと配列要素S[i]を用いて更新する第2の処理部（PRGA）と、を備えている。

[0122] 本発明の別のアスペクトに係る装置は、配列Sの要素の並び替えと算術加算を繰り返すことで配列Sの要素を攪拌し配列Sの初期状態を作成し、その際、内部変数kの初期値を配列Sの要素から求める第1の処理部（KSA）と、第1のインデックス変数iの

WAVELYNX EX. 1003, Pg. 964
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

配列要素S[i]を第1のシフト数ローテイトシフトした値と、第2のインデックス変数jとを算術加算した結果に基づき、第2のインデックス変数jの値を更新し、第2のインデックス変数jの配列要素S[j]を第2のシフト数ローテイトシフトした値と内部変数kとの算術加算結果に基づき内部変数kの値を更新し、S[i]＋S[j]による配設要素S[(S[i]＋S[j])]を第3のシフト数ローテイトシフトした値と内部変数kとの算術加算結果に基づき、鍵ストリームを出力し、前記鍵ストリームを生成するために参照した配列要素S[(S[i]＋S[j])]を、鍵ストリーム出力の直後に内部変数kと配列要素S[i]を用いて更新する第2の処理部(PRGA)と、を備えている。

[0123] 本発明においては、初期設定配列aを用い、配列aの並べ替えを行って攪拌し、内部変数kを対応する配列要素aをローテイトシフトしたものと算術加算して求め、配列Sを、配列aの要素をローテイトシフトしたものと配列Sの要素との所定の演算によって求める第1の処理部(KSA)と、第1、第2のインデックス変数i、jに関して配列Sの参照結果S[j]と内部変数kとを算術加算し、S[i]＋S[j]による配設Sの参照結果と内部変数kに基づき、鍵ストリームを出力し、鍵ストリームを生成するために参照したSのエントリを、鍵ストリーム出力直後に内部変数kを用いて更新する第2の処理部(PRGA)と、を備えた構成としてもよい。上記第1の処理部(KSA)と第2の処理部(PRGA)は、コンピュータ・プログラム(ソフトウェア)として実装してもよい。

[0124] また、本発明に係る方法においては、擬似乱数列の生成に用いる内部状態として、有限個の数値の列の並び替えを基礎とする状態に基づく内部状態を用い、前記擬似乱数列の生成に用いる一時変数の少なくとも一つを、前記内部状態のうちの一つ、又は、複数個の数値を用いた、線形、又は、非線形、線形及び非線形の組み合わせの結果に基づき、内部状態の数より小さい数に依存し、あらかじめ定められた、左又は右ローテイトシフトを実行した結果を値とする一時変数とし、前記擬似乱数を、前記内部状態のうちの1つ又は複数個の数値と、前記一時変数との予め定められた所定の演算によって生成する。本発明によれば、上記第1の処理部(KSA)と第2の処理部(PRGA)の各手順を含む方法が提供される。

発明の効果

[0125] 本発明によれば、GGHN(n,m)におけるdistinguisherの構成を困難とし、GGHN(n,m)

WAVELYNX EX. 1003, Pg. 965
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

がもつ、速度性能の低下を回避することができる。このため、本発明は、データの通信や蓄積の際にデータを秘匿するための安全性の高い暗号装置を提供することができる。

図面の簡単な説明

[0126] [図1]本発明の一実施形態を示すブロック図である。

[図2]32bit RC4 (CISC 2005)で提案されたRC4の改良アルゴリズムを示す図である。

[図3]PRGAの状態遷移 (1)を示す図である。

[図4]PRGAの状態遷移 (2)を示す図である。

[図5]PRGAの解析 (1)を示す図である。

[図6]PRGAの解析 (3)を示す図である。

[図7]PRGAの解析 (2)を示す図である。

[図8]PRGAの解析 (4)を示す図である。

[図9]解読に必要なデータ量(理論値)を示す図である。

[図10]解読に必要なデータ量(実験値)を示す図である。

[図11]実験結果 (distinguisher)を示す図である。

[図12]G. Gong らによってしめされたストリーム暗号のアルゴリズムを示す図である。

[図13]本発明の変更アルゴリズムを示す図である。

[図14]解決しようとする課題の具体例1を示す図である。

[図15]解決しようとする課題の具体例2を示す図である。

[図16]対策案 (1)を示す図である。

[図17]対策案 (2)を示す図である。

発明を実施するための最良の形態

[0127] 上記した本発明についてさらに詳細に説明する。図13は、本発明の変更アルゴリズムを説明するための図である。図13には、本発明において、元のアルゴリズムの変更すべき箇所が示されている。

[0128] 上記課題となった解析手法は、PRGAの構造的な脆弱性を利用したものである。

[0129] しかも、解析手法では、KSAによって全ての内部記憶が一様分布に従うとみなしていた。

WAVELYNX EX. 1003, Pg. 966
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0130] よって、本発明の1つの側面においては、KSAに変更は施さず、PRGAのみ改良をおこなった。

[0131] 図13に示す対策手段におけるアルゴリズムの変更(図2のPRGAのアルゴリズムからの変更)は、変数kの更新の際に、左ローテイト処理 k=((k+S[j])<<<n) mod M

を行っている点である(ただし、N=2^{n}、M=2^{m})。

[0132] ローテイト数は、GGHN(n,m)としたとき、nビットとする。

[0133] 本発明によれば、擬似乱数列の生成に用いる一時変数の少なくとも一つである内部変数kを、前記内部状態のうちの一つ又は、複数個の数値を使用した、線形、又は、非線形、線形及び非線形の組み合わせの結果に基づき、nビットローテイトシフト((k+S[j])<<<n)を実行した結果とし、前記生成される擬似乱数を、前記内部状態のうちの1つ又は複数個の数値と、前記一時変数との演算によって生成する(out=(S[(S(i)+S[j]) mod N] +k) mod M)。

[0134] ここで、本発明による対策手段の効果を説明するため、ケース1における内部変数k0とS1[A]、S2[A+C]をバイト単位の変数を用いて以下のように表現する。

[0135] ここで、右側が下位ビットであり、LSB(k0) = k00である。

ただし、式(1)より、LSB(S1[A]) = B0 = 0である。

[0136] k0 = k03 || k02 || k01 || k00

S1[A] = B3 || B2 || B1 || B0

S2[A+C] = A3 || A2 || A1 || A0

[0137] このとき、算術加算で生じるバイトを跨ぐ桁あがりの影響を無視すると、図13に示したPRGAは、最下位バイトにおいて、一様分布に従う独立な変数が必ず2個以上挿入されるため、式(1)に、偏りが生じなくなる。

[0138] 同様に、どのバイト間で比較しても、一様な分布に従う独立な変数が必ず2個以上挿入されることから、偏りは生じないと考えられる。

[0139] 実際には、桁上がりがバイトを超えて影響するが、基本的な考え方に影響は無い。

[0140] また、図13の対策手段における実装性を考慮すると、処理の増加は、ローテイト処理1回分である。

[0141] 図13の比較例として、元のアルゴリズムを図12に示す。変数kの更新は、

WAVELYNX EX. 1003, Pg. 967
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

k=(k+S[j]) mod M

にて行っている。

[0142] また、本発明において、実装ターゲットとして、32ビット／64ビットプロセッサ上でのソフトウェアの実装を想定した場合、nビットの左ローテイト処理は速度の劣化が小さいものと思料される。

[0143] また、本発明によれば、新たな内部メモリを用いることなく、改良が可能であるため、メモリの増加も無い。

[0144] よって、本発明による、図13の対策手段は、

・設計者らの設計方針を崩すことなく、

・実装性の利点を損なうこともない、

暗号モデルを実現できるものと期待される。

[0145] 次に、図16に、本発明による、KSA(K, S)の変更による対策手段の例を示す。a[i]を左8xrビットローテイトしたものと変数kとを算術加算した値で、変数kを更新している。S[i]と、a[i]を左8x(3-r)ビットローテイトしたものとの演算結果（排他的論理和演算）にて、S[i]を更新している。

[0146] 図16の、KSA(K, S)の変更による対策手段でも、S-boxはある程度、ランダムになることが実験により確認できており、かつ、各バイトで見たとき出力均等となる。

[0147] そして、KSAの速度は、約2.5倍高速（RC4のKSAと比べて約8倍低速）となり、初期値テーブルを使用しないので、例えば1Kbyteのメモリを削減することができる。

[0148] なお、図16の対策手段は、ローテイトシフトの方向及び／又はシフト数を、前記内部状態の数値に依存して動的に変更することに対応する。ローテイトシフトの方向及びシフト数を、あらかじめ定められたテーブルの値に従って変更するようにしてもよい。

[0149] 図17に、PRGA(S)の変更による対策手段の別の例を示した。S[i]を右24ビットローテイトシフトしたもの（S[i]>>>24）とjとの算術加算結果でjの値を更新し、S[j]を右16ビットローテイトシフトしたもの（S[j]>>>16）とkとの算術加算結果でkの値を更新し、S[(S[i]+S[j]) mod N]を右8ビットローテイトしたものとkとの算術加算結果をoutとしている。

[0150] この対策手段の場合、鍵ストリーム間の関係（1番目と2番目の出力バイトO1とO2）

WAVELYNX EX. 1003, Pg. 968
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

が、

O1 = k0 + XO2 = k0 + Y

となり、Xの値に応じて、Yの値が決定する関係となる。

[0151] よって、インデックスの仮定で参照されるS-boxエントリを選択できたとしても。仮定で用いたインデックス情報を、上記の関係式に持ち込めない(X, Yの情報によらない恒等式に持ち込めない)ため、Distinguishing Attackの適用が困難になる。

[0152] なお、本発明において、前記有限個の数値の列の並び替えを基礎とする状態に基づく内部状態であって、前記内部状態の更新を、並び替え以外の、線形演算、及び、非線形演算を用いて行うようにしてもよい。

[0153] 本発明において、前記有限個の数値の列の並び替えを基礎とする状態に基づく内部状態であって、前記内部状態の更新として、列の並び替え以外の、線形演算及び非線形演算を用いることにより、前記内部状態の状態数を、単調増加させるようにしてもよい。あるいは、前記内部状態の更新として、列の並び替え以外の、線形演算及び非線形演算を用いることにより、前記内部状態の状態数を、単調減少させるようにしてもよい。

[0154] 本発明において、前記有限個の数値の列の並び替えを基礎とする状態に基づく内部状態であって、前記内部状態の更新として、並び替え以外の線形演算及び非線形演算を用いることにより、内部状態の状態数を振動させる、ようにしてもよい。

[0155] 本発明においては、前記内部状態の更新を、前記擬似乱数列の出力ごとに行うようにしてもよい。あるいは、前記擬似乱数列の出力よりも多く行うようにしてもよい。あるいは、前記擬似乱数列の出力よりも少なく行うようにしてもよい。以下実施例に即して説明する。

実施例

[0156] 図1は、本発明の一実施例を説明するための図である。図14、図15は、比較例として、本発明が解決しようとする課題の具体例1、2を示すブロック図である。以下、本発明の一実施例を説明する。図1に示すように、<<<nの処理では、k(0)に対して、左ローテイト処理(k+S[j])<<<nを行ってk(1)としている。(S[(S[i]+S[j]) mod N]+k(1))mod Mをout(1)としている。

WAVELYNX EX. 1003, Pg. 969
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

[0157] 図14は、本発明が解決しようとする課題の具体例1として、図5から図7（ケース2の条件でのt=1、2における配列Sの最下位バイトの状態遷移）への処理の過程を示した図である。状態変化は、Sの箱で示されており、図には表れていない。はじめに、図14の比較例を参照して説明する（アルゴリズムは図12に示される）。

[0158] 時刻1において、インデックスiが1となる（j(0)の矢印の先の箱から出力される1参照）。インデックスi(0)と1を入力する十字ボックスは、算術加算を表しており、i＝（i＋1）mod N＝1を出力する。

[0159] アドレスiの値S[1]（下位8ビット）がAであり、インデックスjがAとなる。なお、図1、図14、図15において、Sを四角で囲むボックスはS－boxであり、例えばインデックス（例えば1）を入力し、S[1]がAの場合、ボックスからは、Aが出力される図式とされる。j(0)とAを入力とするボックス（算術加算器）は、j(0)=（j(0)+A） mod N＝Aを出力している。

[0160] アドレスjの値S[j]（下位8ビット）が0であり、

変数kの値（下位8ビット）は

0＋k0＝k0

となる。

[0161] アドレスiの値S[i]（下位8ビット）がAであり、アドレスjの値S[j]（下位8ビット）が0であるため、出力O1を生成するために必要なアドレスは、

S[i]＋S[j]＝A＋0＝A

となる。

[0162] アドレスAの値S[A]（下位8ビット）が0であり、変数kの値（下位8ビット）がk0であるため、出力O1の値（下位8ビット）は、

0＋k0＝k0

となる。

[0163] 出力O1を生成した後、アドレスAの値S[A]（下位8ビット）は、

k0＋A

に更新される。

[0164] 時刻2において、インデックスiは2となる。

[0165] アドレスiの値S[i]（下位8ビット）がCであり、インデックスjは、A＋Cとなる。

[0166] アドレスjの値S[j]（下位8ビット）がAであり、変数kの値（下位8ビット）は、

k0＋A

となる。

[0167] アドレスiの値S[i]（下位8ビット）がCであり、アドレスjの値S[j]（下位8ビット）がAであるため、出力O2を生成するために必要なアドレスは、

S[i]＋S[j]＝A＋C

となる。

[0168] アドレスA＋Cの値S[A+C]（下位8ビット）がAであり、変数kの値（下位8ビット）がk0＋Aであるため、出力O2の値（下位8ビット）は、

A＋k0＋A＝k0＋2A

となる。

[0169] 出力O2を生成した後、アドレスA＋Cの値S[A+C]（下位8ビット）は、

k0＋A＋C

に更新される。

[0170] 図15は、図1の比較例として、本発明が解決しようとする課題の具体例2を示す図であり、図5から図6（ケース1の条件でのt=1、2における配列Sの最下位バイトの状態遷移）への処理の過程を示した図である。状態変化はSの箱で示されており、図には表れていない。

[0171] 図15を参照すると、時刻1において、インデックスiが1となる。

[0172] アドレスiの値S[i]（下位8ビット）がAであり、インデックスjがAとなる。

[0173] アドレスjの値S[j]（下位8ビット）が0であり、変数kの値（下位8ビット）は、 0＋k0＝k0

となる。

[0174] アドレスiの値S[i]（下位8ビット）がAであり、アドレスjの値S[j]（下位8ビット）が0であるため、出力O1を生成するために必要なアドレスは、

S[i]＋S[j]＝A＋0＝Aとなる。

[0175] アドレスAの値S[A]（下位8ビット）が0であり、変数kの値（下位8ビット）がk0であるため、出力O1の値（下位8ビット）は、

WAVELYNX EX. 1003, Pg. 971
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

0＋k0＝k0

となる。

[0176] 出力O1を生成した後、アドレスAの値S[A]（下位8ビット）は、

k0＋A

に更新される。

[0177] 時刻2において、インデックスiは2となる。

[0178] アドレスiの値S[i]（下位8ビット）がCであり、インデックスjはA＋Cとなる。

[0179] アドレスjの値S[j]（下位8ビット）が2－Cであり、変数kの値（下位8ビット）は、

k0＋2－C

となる。

[0180] アドレスiの値S[i]（下位8ビット）がCであり、アドレスjの値S[j]（下位8ビット）が2－Cであるため、出力O2を生成するために必要なアドレスは、

S[i]＋S[j]＝C＋2－C＝2

となる。

[0181] アドレス2の値S[2]（下位8ビット）がCであり、変数kの値（下位8ビット）がk0＋2－Cであるため、出力O2の値（下位8ビット）は、

C＋k0＋2－C＝k0＋2

となる。

[0182] 出力O2を生成した後、アドレス2の値S[2]（下位8ビット）は、

k0＋2－C＋C＝k0＋2

に更新される。

[0183] 図14、図15の比較例と相違して、本発明の一の実施例では、図1に示すように、処理の過程にローテイトシフト（<<<n）が追加されている。k(0)=ローテイトシフト（<<<n）は、((k+S[j])<<<n) mod M

[0184] 本実施例においても、図14と同様、時刻1において、インデックスiが1となる。アドレスiの値S[i]（下位8ビット）がAであり、インデックスjがAの下位8ビットとなる。

[0185] アドレスjの値S[j]（下位8ビット）がB（下位8ビットが0）であり、変数kの値はRoln（k0＋B）となる。なお、Rolnは、Rotate-Left-Shift by n-bitを表している。すなわち、Rol

n(k0＋B)は、図13のk＝((k＋S[j]<<<n) mod Mに対応し、k0＝((k0＋B)<<<n)mod Mが実行される。

[0186] アドレスiの値S[i](下位8ビット)がAであり、アドレスjの値S[j]がB(下位8ビットが0)であるため、出力O1を生成するために必要なアドレスは、

S[i]＋S[j]＝A＋0＝A

となる。

[0187] アドレスAの値S[A]がB(下位8ビットが0)であり、変数kの値がRoln(k0＋B)であるため、出力O1の値(下位8ビット)は、

0＋Roln(k0＋B)＝Roln(k0＋B)

となる。

[0188] 出力O1を生成した後、アドレスAの値S[A](下位8ビット)は、

Roln(k0＋B)＋A

に更新される。

[0189] 時刻2において、インデックスiは2となる。

[0190] アドレスiの値S[i](下位8ビット)がCであり、インデックスjはA＋Cの下位8ビットとなる。

[0191] アドレスjの値S[j](下位8ビット)がAであり、変数kの値は、

Roln(Roln(k0＋B)＋A)

となる。

[0192] アドレスiの値S[i](下位8ビット)がCであり、アドレスjの値S[j](下位8ビット)がAであるため、出力O2を生成するために必要なアドレスは、

S[i]＋S[j]＝A＋C

となる。

[0193] アドレスA＋Cの値S[A+C](下位8ビット)がAであり、変数kの値がRoln(Roln(k0＋B)＋A)であるため、出力O2の値(下位8ビット)は、

A＋Roln(Roln(k0＋B)＋A)

となる。

[0194] 出力O2を生成した後、アドレスA＋Cの値S[A+C](下位8ビット)は、

Roln(Roln(k0＋B)＋A)＋C

に更新される。

[0195] 次に、図1の本実施例において、図15に基づくと、時刻1において、インデックスiが1となる。

[0196] アドレスiの値S[i](下位8ビット)はAであり、インデックスjはAの下位8ビットとなる。

[0197] アドレスjの値S[j](下位8ビット)はB(下位8ビットが0)であり、変数kの値は、

Roln(k0＋B)

となる。

[0198] アドレスiの値S[i]がAであり、アドレスjの値S[j]がB(下位8ビットが0)であるため、出力O1を生成するために必要なアドレスは、

S[i]＋S[j]＝A＋0＝A

となる。

[0199] アドレスAの値S[A](下位8ビット)がB(下位8ビットが0)であり、変数kの値がRoln(k0＋B)であるため、出力O1の値(下位8ビット)は、

0＋Roln(k0＋B)＝Roln(k0＋B)

となる。

[0200] 出力O1を生成した後、アドレスAの値S[A](下位8ビット)は、

Roln(k0＋B)＋A

に更新される。

[0201] 時刻2において、インデックスiは2となる。

[0202] アドレスiの値S[i](下位8ビット)がCであり、インデックスjは、A＋Cの下位8ビットとなる。

[0203] アドレスjの値S[j](下位8ビット)が2－Cであり、変数kの値は、

Roln(Roln(k0＋B)＋2－C)

となる。

[0204] アドレスiの値S[i]がCであり、アドレスjの値S[j]が2－Cであるため、出力O2を生成するために必要なアドレスは、

S[i]＋S[j]＝C＋2－C＝2

となる。

[0205] アドレス2の値S[2]（下位8ビット）がCであり、変数kの値が、

Roln（Roln（k0＋B）＋2－C）

であるため、出力O2の値（下位8ビット）は、

C＋Roln（Roln（k0＋B）＋2－C）

となる。

[0206] 出力O2を生成した後、アドレス2の値S[2]（下位8ビット）は、

Roln（Roln（k0＋B）＋2－C）＋C

に更新される。

[0207] 本発明によれば、データの通信や蓄積の際にデータを秘匿するための安全性の高い暗号装置を得ることができる。

[0208] なお、図13、図16、図17を参照して説明した本発明に係るプログラムは、鍵ストリームを生成する任意のアプリケーションに適用いることができる。本発明に係る暗号装置は、例えばサーバ装置のCPU、記憶装置、ネットワーク等を含む構成とされる。秘密鍵情報は、サーバ装置の記憶装置に記憶される。図1に示した構成において、ローテイトシフト演算は、CPUのALU（演算装置）で行われる。

[0209] 以上、本発明を上記実施例に即して説明したが、本発明は上記実施例の構成にのみ制限されるものでなく、本発明の範囲内で当業者であればなし得るであろう各種変形、修正を含むことは勿論である。

WAVELYNX EX. 1003, Pg. 975
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

請求の範囲

[1] 秘密鍵に基づき擬似乱数列を生成し、前記擬似乱数列を平文に作用させることにより暗号文を生成する暗号装置であって、

前記擬似乱数列の生成に用いる内部状態として、有限個の数値の列の並び替えを基礎とする状態に基づく内部状態を用い、

前記擬似乱数列の生成に用いる一時変数の少なくとも一つを、

前記内部状態のうちの一つ、又は、複数個の数値を用いた、線形、又は、非線形、線形及び非線形の組み合わせの結果に基づき、内部状態の数より小さい数に依存し、あらかじめ定められた、左又は右ローテイトシフトを実行した結果を値とする一時変数とし、

前記擬似乱数を、前記内部状態のうちの1つ又は複数個の数値と、前記一時変数との予め定められた所定の演算によって生成する手段を備えている、

ことを特徴とする暗号装置。

[2] 前記有限個の数値の列の並び替えを基礎とする状態に基づく内部状態に関して、

前記内部状態の更新を、並び替え以外の、線形演算、および、非線形演算を用いて行う、ことを特徴とする請求項1に記載の暗号装置。

[3] 前記有限個の数値の列の並び替えを基礎とする状態に基づく内部状態に関して、

前記内部状態の更新を、並び替え以外の、線形演算及び非線形演算を用いて行い、前記内部状態の状態数を、単調増加と単調減少のうちの一方にしたがって変化させる、ことを特徴とする請求項1に記載の暗号装置。

[4] 前記有限個の数値の列の並び替えを基礎とする状態に基づく内部状態に関して、

前記内部状態の更新を、並び替え以外の、線形演算及び非線形演算を用いて行い、前記内部状態の状態数を振動させる、ことを特徴とする請求項1に記載の暗号装置。

[5] 前記内部状態の更新を、

前記擬似乱数列の出力ごとに行う、

前記擬似乱数列の出力よりも多く行う、

前記擬似乱数列の出力よりも少なく行う、

WAVELYNX EX. 1003, Pg. 976
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018

のうち選択されたいずれか一つで行う、ことを特徴とする、請求項1乃至4のいずれか一項に記載の暗号装置。

[6] 前記ローテイトシフトの方向及び／又はシフト数を、前記内部状態の数値に依存して動的に変更する、ことを特徴とする、請求項1乃至5のいずれか一項に記載の暗号装置。

[7] 前記ローテイトシフトの方向及びシフト数を、あらかじめ定められたテーブルの値に従って変更する、ことを特徴とする、請求項1乃至5のいずれか一項に記載の暗号装置。

[8] 前記擬似乱数列の生成に用いる内部状態として、配列Sの要素の並び替えと算術加算を繰り返すことで配列Sの要素を攪拌し配列Sの初期状態を作成し、その際、前記一時変数をなす内部変数kの初期値を配列Sの要素から求める第1の処理部と、

前記擬似乱数（以下、「鍵ストリーム」と称する）を生成するにあたり、第1、第2のインデックス変数iとjに関して配列Sの要素S[j]と内部変数kの加算結果をローテイトシフトした値で内部変数kの値を更新し、

S[i]＋S[j]による配設Sの参照結果S[(S[i]＋S[j])]と内部変数kとの加算結果に基づき、鍵ストリームを出力し、

前記鍵ストリームを生成するために参照した配列Sの要素S[(S[i]＋S[j])]を、鍵ストリーム出力の直後に内部変数kと配列要素S[i]を用いて更新する第2の処理部と、

を備えている、ことを特徴とする請求項1記載の暗号装置。

[9] 配列Sの要素数Nとローテイトシフトのシフト数nとが$N=2^n$の関係とされる、ことを特徴とする請求項8記載の暗号装置。

[10] 前記擬似乱数列の生成に用いる内部状態として、配列Sの要素の並び替えと算術加算を繰り返すことで配列Sの要素を攪拌し配列Sの初期状態を作成し、その際、前記一時変数をなす内部変数kの初期値を配列Sの要素から求める第1の処理部と、

前記擬似乱数（以下、「鍵ストリーム」と称する）を生成するにあたり、

第1のインデックス変数iの配列要素S[i]を第1のシフト数ローテイトシフトした値と、第2のインデックス変数jとを算術加算した結果に基づき、第2のインデックス変数jの

WAVELYNX EX. 1003, Pg. 977
WaveLynx Technologies v. ASSA Abloy, IPR2023-01018