

Cited Prior Art	Relevant Disclosure
GB 2518976	Key Exchange: “the M2M device deriving the cryptographic material from the public key of the server and the private key of the M2M device; and the server deriving the cryptographic material from the public key of the M2M device and the private key of the server.” 9:6-9.
US 5,664,017	Digital Signature: “A suitable digital signature can be applied using the RSA method described in the above-referenced Rivest et al publication.” 14:5-7. “generating a first random string at a first subscriber entity, signing the random string using the first subscriber entity’s private key and encrypting the random string using the second subscriber entity’s public key and transmitting to the second subscriber entity the first subscriber entity’s certificate and the encrypted and signed random number.” 17:4-11.
US 5,852,665	Digital Signature: “Still further in accordance with a preferred embodiment of the present invention the message and the encrypted session keys are sent signed by the sender subscriber.” 5:34-36. “A suitable digital signature can be applied using the RSA method described in the above-referenced Rivest et al publication.” 17:1-3.
US 5,995,625	Digital Signature: “With reference to FIG. 6B, when a subscriber uses the system, he signs a message 65 with his private key 67, to produce a subscriber signature 69. The subscriber’s private key 67 corresponds to the subscriber’s public key 48 for use with a public key cryptographic system.” 8:32-36.
US 6,038,322	Digital Signature: “FIG. 2 shows the same four exchanges as in FIG. 1, except that member M, in exchange (3), digitally signs Authentication-M to form a digital signature S_m ($\{ \text{Authentication-M} \} \text{Sig_member_M}$) and sends signature S_m, instead of Authentication-M, to key holder H.” 6:48-53. Diffie-Hellman Key Exchange: “The message transmitter and recipient often desire that a message be transmitted to and from the intended party and not be intercepted by a third party. Cryptography, including Diffie-Hellman exchanges, public/private key methodology, and other encryption techniques, provides different ways of message encryption and/or authentication which help solve this problem.” 1:12-18.
US 6,256,733	Digital Signature: “To further enhance the security of the transmission, the sending party may sign the message using its private signature key.” 1:48-50.

US 6,842,628	Digital Signature: "The received message is signed with the first user's private encryption key. A signature of the message is verified by using the first user's public encryption key." 3:5-9.
US 8,127,142	Diffie-Hellman Key Exchange: Asymmetric, or public key, cryptography is a form of cryptography which allows users to communicate securely without having prior access to a shared secret key. This is done by using a pair of mathematically related cryptographic keys, designated as the public key and the private key. The private decryption key is kept secret as it resides nowhere except on the portable token. The public key, however, may be distributed across the enterprise system. Examples of known asymmetric encryption systems that can be used with the inventive system include, but are not limited to Diffie-Hellman...." 5:42-52.
US 8,526,606	Digital Signature: "For example, digital signature algorithm, may be used where the sender uses a private key to sign the digital message and the public key is used to authenticate the signature thereby confirming that the sender is the authorized entity it purports itself to be." 5:49-53.
US 8,782,774	Diffie-Hellman Key Exchange: "Although not illustrated in FIG. 4, the secure session server 420 has selected a cipher suite that has a key exchange in which the certificate message transmitted in operation 4.3 does not include enough data to allow the client device to generate a premaster secret. For example, the selected cipher suite may use Diffie-Hellman as the key exchange mechanism (e.g., DHE RSA, DHE DSS). Because of this, the secure session server 420 will transmit a message to the client device 410 that conveys cryptographic information to allow the client device 410 and the secure session server 420 to each generate the same premaster secret. By way of a specific example where the key exchange mechanism is Diffie-Hellman, the cryptographic information includes a set of cryptographic parameters that may include the following: the prime modulus used for the Diffie-Hellman operation (p), the generator used for the Diffie-Hellman operation (g), and a Diffie-Hellman public value of the server ($g^X \text{ mod } p$, where X is the Diffie-Hellman private value of the server). The message that conveys the cryptographic information is referred to as a Server Key Exchange message. The Server Key Exchange message may need to be signed with the private key 450 corresponding to the public key of the server transmitted in the Certificate message transmitted in operation 4.3 (e.g., if the key exchange mechanism is DHE RSA or DHE DSS). As similarly described with respect to the embodiment discussed in FIG. 1, the secure session server 420 does not have local access to this private key 450. As a result, the secure session server 420 cannot sign the Server Key Exchange message with this private key 450." 10:51-11:12.
US 9,002,018	Digital Signature:

	“Additionally, Person A can encrypt a message with Person A’s private key and transmit it to Person B, who may decrypt the message using Person A’s public encryption key. The second scenario is called a ‘digital signature’ and serves to authenticate Person A to Person B, since it is assumed that only Person A has access to Person A’s private key.” 1:44-50. Diffie-Hellman Key Exchange: “Numerous prior art solutions have been developed to address security issues. Public Key Infrastructure (PKI) key exchange (also called Diffie-Hellman key exchange) was first published publicly in 1976 by Whitfield Diffie and Martin Hellman.” 1:29-33.
US 9,253,643	Digital Signature: “Described herein are digital signature methods. Digital signature methods may use public key cryptography.” 3:27-28. “Digital signatures in many cases is nothing more than an encrypted hash. While digital signature methods may use public/private key pairs, message authentication codes (MACs) may use a shared secret key.” 4:26-29.
US 10,169,587	Diffie-Hellman Key Exchange: “The DPP server can also include a set cryptographic algorithms, where the cryptographic algorithms can include a random number generator, a PKI key pair generation algorithm, a secure hash algorithm, an elliptic curve Diffie Hellman key exchange algorithm, a symmetric ciphering algorithm, and a digital signature algorithm.” 5:9-15. Digital Signature: “In embodiments where initiator 102 uses a device certificate for initiator 102 (which could be a Cert0.initiator 102m), authentication server 111 could verify that initiator 102 also records the private key SK0.initiator 102s, by (i) sending a challenge/random nonce and (ii) verifying a signature for the challenge/nonce from initiator 102 using the private key 102s for cert0.initiator 102m. Authentication server 111 can verify the signature using the public key for initiator 102 from Cert0.initiator 102m.” 31:39-48.
US 10,764,066	Digital Signature: “The eUICC computes a cryptographic signature over the payload using a key of the eUICC. For example, the key is a private key of a PKI public key-private key of the eUICC. The eUICC sends a response message containing the response payload and the eUICC signature to the device.” 2:41-46.
US 2001/0029581	Digital Signature: “Asymmetric key encryption technology can also be used to verify the authenticity of a message in a process known as signing a message. In this case, a one-way hash of a message is encrypted with the private key of the sender. This encrypted one-way hash is also known as a signature. The signature is sent along with the message itself. The recipient, upon receiving the message and the signature, decrypts the signature (encrypted hash) using the public key of

	the sender. The recipient also reproduces a hash of the received message using the same one-way hash function used by the sender. If the decrypted hash and the reproduced hash match, the message must be the same, unaltered message that was send by the sender. In order to assure the authenticity of the public key used to verify the signature, digital certificates have been developed. A digital certificate is a public key that has been signed by a trusted authority that can vouch for the authenticity of the public key. Digital certificates can be easily disseminated and authenticated to facilitate secure communications and authentications.” ¶ 41.
US 2002/0018569	Diffie-Hellman Key Exchange: “The base station enables registration by sending a short-lived elliptic curve public key along with its identifier to the mobile station. This information can either be broadcast on the overhead channel or it can be sent to an individual mobile station in response to a registration request from the mobile station. In response, the mobile station combines the short-lived base station public key with its own private key and generates two shared secret keys using the elliptic curve Diffie-Hellman method. The first of these two keys is used as a MAC key to authenticate the mobile station to the base station and the base station to the mobile station, and the second is used to establish secret session keys.” ¶¶ 50-51.
US 2006/0021063	Diffie-Hellman Key Exchange: “Examples of algorithms employed in the public key cryptosystem include RSA, EC-DH (Elliptic Curve Diffie-Hallman), and so forth. With the present embodiment, the EC-DH algorithm is employed. With the EC-DH, data is encrypted with a symmetric key (which is also referred to as ‘shared key’) shared between the transmitter and the receiving, using multiplication on an elliptic curve on an infinite field (The encryption algorithm will be referred to as ‘elliptic curve encryption algorithm’).” ¶ 70.
US 2007/0101400	Diffie-Hellman Key Exchange: “A secure connection is established between the first computer or computer network and a second computer or computer network using a common symmetric encryption key. The symmetric encryption key may be previously known to both parties or a unique symmetric encryption key may be generated for each session. One means of generating a unique common symmetric encryption key utilizes the public keys of both parties and a key exchange algorithm, such as Diffie-Hellman key exchange; however, any suitable key exchange algorithm may be used to generate the symmetric encryption key.” ¶ 26. “10. The method according to claim 1, wherein the symmetric encryption key is generated by both the first computer or computer network and the second computer or computer network using a Diffie-Hellman key exchange algorithm.” Cl. 10.
US 2009/0061097	Key Exchange:

	“The KMP derives the shared key using its key pair, the sender’s public key, and the shared key parameters (step 1008) using a protocol such as ECC. Once derived, the KMP uses the shared key to decrypt the symmetric key (step 1010).” ¶ 105.
US 2009/0077643	Diffie-Hellman Key Exchange: “For example, using Diffie-Hellman with a certificate authority (CA), a key exchange can take place between trusted entities. In such a scenario, the parties require mutual identity with verification by the CA.” ¶ 141.
US 2009/0125996	Digital Signature: “It should be noted that, the MNO directly or indirectly provides the necessary functions of a public key infrastructure. The MNO possesses a certificate Cert-MNO which has been issued by a certification authority CA. This certificate links the identity of the MNO to the public key K-pub-MNO, which is necessary for checking digital signatures.” ¶ 39. Diffie-Hellman Key Exchange: “For example, using Diffie-Hellman with a certificate authority (CA), a key exchange can take place between trusted entities. In such a scenario, the parties require mutual identity with verification by the CA.” ¶ 154.
US 2009/0191857	Digital Signature: “The DSE may also contain a private/public key pair for signing information sent to an operator. The public key may be accompanied by a certificate.” ¶ 61.
US 2009/0268909	Diffie-Hellman Key Exchange: “In the context of a concrete embodiment the key exchange is performed according to the Diffie-Hellmann algorithm.” ¶ 21.
US 2009/0313472	Diffie-Hellman Key Exchange: “1. A method for securing communications between a Universal Integrated Circuit Card (UICC) and a Terminal, the method comprising: generating a secure shared session key; and encrypting communications between the UICC and the Terminal with the secure shared session key. ... 23. The method of claim 1, wherein the generating includes performing a Diffie-Hellman key exchange protocol.” Cls. 1, 23.
US 2010/0023771	Diffie-Hellman Key Exchange: “In a first key agreement protocol shown in FIG. 2, an static elliptic curve Diffie Hellman (ECDH) key generation is employed with certificates T_A, T_B, signed using ECDSA. Initially, the correspondents 12, 14 exchange public keys A, B and the respective certificates T_A, T_B. The common key is $K=abG$, which can be computed for each

	party from the public information received and the private information it has. Thus $K_A = aB$ and $K_B = bA$.” ¶¶ 44-45.
US 2010/0211779	Diffie-Hellman Key Exchange: “Suppose A, B are the two entities (or parties, where A represents a computer system of a first party and B represents a computer system of a second party) that are attempting to authenticate and agree on a key. We will use A and B to represent their corresponding identities, which by definition also represent their public keys. Let $H1(A) = QA$ and $H1(B) = QB$ be the respective points on the elliptic curve corresponding to the public keys. In effect, one could refer to QA and QB as the public keys as well, since there is a one-to-one correspondence between the identities and the points on the curve obtained by applying H1. Let x be a random number chosen by A, and let y be a random number chosen by B. FIG. 1 illustrates the protocol exchanges between A and B. The protocol exchange 100 comprises of the following steps: A computes xP (i.e., P added to itself x times as a point on E, using the addition law on E) encrypts it using B's public key, and transmits it to B in step 102. In this step, encryption refers to identity based encryption described in section I above. Upon receipt of the encrypted message, B decrypts the message and obtains xP. Subsequently B computes yP, and encrypts the pair {xP, yP} using A's public key and then transmits it to A in step 104. Upon receipt of this message, A decrypts the message and obtains yP. Subsequently, A encrypts yP using B's public key and sends it back to B in step 106. Following this, both A and B compute xyP as the session key. Observe that A chose x randomly, and received yP in the second step of the protocol exchange. This allows A to compute xyP by adding yP to itself x times. Conversely, B chose y randomly, and received xP in the first step of the protocol exchange. This allows B to compute xyP by adding xP to itself y times. Note that any application of the protocol may utilize header data with the identities to ensure proper functioning of the protocol. This is relatively standard and applicable to almost any protocol exchange for key agreement. Note also that x is random but xP provides no information about x. Therefore, xP is a component of a key based on a random secret chosen by A. Likewise, y is random but yP provides no information about y. Hence, yP is a component of a key based on a random secret known only to B. Note further that xyP can serve as a session key. Also, the session key could be any known function of xyP. That is, the session key could equal $f(xyP)$, where f is known to both parties and is not required to be secret (i.e., known to the world). One practical requirement on f should be that f is hard to compute without

	knowledge of x or y, and the output is of a satisfactory length from a cryptographic perspective, e.g., around 128 bits or more. Some of the properties of the protocol include: Immunity from key escrow: Observe that all the steps in the protocol exchange are encrypted using IBE. So clearly the KGF can decrypt all the exchanges. However, the KGF can not compute the session key. This is because of the hardness of the elliptic curve Diffie-Hellman problem. In other words, given xP and yP, it is computationally hard to compute xyP.” ¶¶ 31-44.
US 2011/0055553	Diffie-Hellman Key Exchange: “After deployment, each sensor node computes a shared secret key with KDC 105 for later authentication and access control process. The present invention is based on Elliptic Curve Diffie-Hellman (ECDH) to establish a key agreement between each sensor node 102, 103, 104 and KDC 105. ECDH is a key agreement protocol allowing two parties to establish a shared secret key that can be used for private key algorithms. It has been shown that ECDH with 160-bit key size can achieve the same security level with 1024-bits RSA Diffie-Hellman secret sharing protocol. To establish a shared secret key with KDC, a sensor node, say S, computes $R_S=(x_S, y_S)=k_S \times Q_{KDC}$. KDC also computes $R_{KDC}=(x_{KDC}, y_{KDC})=k_{KDC} \times Q_S$. Since $k_S \times Q_{KDC}=k_S \times k_{KDC} \times P=k_{KDC} \times Q_S$, therefore $R_S=R_{KDC}$ and hence $x_S=x_{KDC}$. As a result, x_S is used as a shared secret key between node S and KDC. This key agreement is done only once for the whole network lifetime. As a consequence, it does not consume much energy overall. It can be performed before or right after network deployment.” ¶¶ 18-19.
US 2011/0158411	Diffie-Hellman Key Exchange: “With reference now to FIG. 5, there is shown a process diagram 500 of a manner in which the unique key generating module 224 generates a unique symmetric key for a client device 140 a, according to an embodiment. As shown therein, the unique key generating module 224 receives as inputs, the private key of the device registration server 510 and the public key of the client device A 520 a. In addition, the unique key generating module 224 generates the unique symmetric key for the client device A 540 a. According to an embodiment, the unique key generating module 224 generates the unique symmetric key 540 a for the client device 140 a using a cryptographic function. Examples of suitable cryptographic functions include an Elliptic Curve Diffie-Hellman (ECDH) function or a Diffie-Hellman (DH) function. Thus, for instance, the unique key generating module 224 may input the private key of the device registration server 510 and the public key of the client device A 140 a into either of the ECDH function or the DH function to generate the unique symmetric key 540 a for the client device A 140 a. Moreover, the unique key generating module 224 also generates unique symmetric keys 540 b-540 n for the remaining client devices 140 b-140 n through similar operations. By way of particular example in which the client device A 140 a has a private key, x, and the device registration server 110 has a private key, y, the public key 520 a of the

	client device A may be computed as: $g^x \bmod p$, Equation (1): in which “g” represents a generator, which is the generating element in a finite cyclic group “G” and “p” represents a “Prime Number”, which is the modulo. In addition, the public key of the device registration server 110 is computed as: $g^y \bmod p$, Equation (2): in which “g” also represents the same generator, which is the generating element in the finite cyclic group “G” and “p” also represents the “Prime Number”, which is the modulo. Moreover, in this example, the unique symmetric key 540 a for the device registration server 110 may be generated from the public key 520 a ($g^x \bmod p$) of the client device A and the private key 510 (y) of the device registration server 110 using the DH function as: $(g^x \bmod p)^y = g^{xy} \bmod p$. Equation (3): In addition, in this example, the unique symmetric key 540 a for the device 140 a may be generated from the public key 720 ($g^y \bmod p$) of the device registration server 110 and the private key 710 a (x) of the device 140 a using the DH function as: $(g^y \bmod p)^x = g^{xy} \bmod p$. Equation (4): As shown from the example above, both the device registration server 110 and the client devices 140 a-140 n are able to compute the same unique symmetric key. In one regard, ECDH can be used instead of DH and is generally preferred over DH because the public key size for ECDH is smaller with equivalent security strength, for example, a 256 byte DH public key may provide a similar level of security strength to a 32 byte ECDH public key.” ¶¶ 31-38.
US 2012/0023336	Diffie-Hellman Key Exchange: “Upon receiving the public key from the server, the client generates a secret key of length ‘m’ bits using elliptic curve Diffie hellman algorithm and shares the secret key with the server. The server in return generates an ‘m’ bit session key based on elliptic curve Diffie hellman algorithm. The secret key and the corresponding session key are generated using elliptic curve Diffie hellman algorithm. Reference numeral 100 of FIG. 1 denotes the step of generation of session key of length ‘m’ bits using elliptic curve Diffie hellman algorithm. For all the subsequent communications and transactions between the server and the client, the session key is used.” ¶ 79.
US 2012/0030461	Digital Signature: “When each device receives its certificate, it verifies the signature in the certificate with CA 50’s public key; if the certificate is valid, the device will sign the signature on the certificate with its private key and send this signature to CA 50 as the acknowledgement.” ¶ 33.
US 2012/0087493	Diffie-Hellman Key Exchange:

	“4. The method of claim 1 wherein generating the storage key from the second private key and the first public key utilizes an elliptic curve Diffie-Hellman scheme.” Cl. 4.
US 2013/0145151	Diffie-Hellman Key Exchange: “In one example, the first device 200 and the second device 250 may use one another’s short-term public keys in a key agreement protocol, such as a Diffie-Hellman protocol, to establish a shared secret that can be used for symmetric encryption.” ¶ 57.
US 2013/0166915	Digital Signature: “A digital certificate (also known as a public key certificate or identity certificate) generally refers to an electronic document which uses a digital signature to bind a public key with an identity—information such as the name of a person or an organization, their address, etc. The certificate can be used to verify that a public key belongs to an individual. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit.” ¶ 55.
US 2014/0003604	Diffie-Hellman Key Exchange: “In some embodiments, the key agreement protocol in FIG. 4 (and described herein generally) is an asymmetric key agreement protocol. An asymmetric key agreement protocol is a key agreement protocol that utilizes one or more asymmetric keys, i.e., one or more private and/or public keys, to agree upon a shared symmetric key. This differs from what some might call a ‘symmetric key agreement protocol’ in which a shared symmetric key is derived from one or more previously shared symmetric values, such as a previously shared symmetric key. Examples of asymmetric key agreement protocols include the Diffie-Hellman key agreement protocol and the MQV key agreement protocol.” ¶ 46.
US 2014/0082358	Diffie-Hellman Key Exchange: “To be able to establish a profile encryption key (PEK) using ECC, a key agreement exchange may take place between the MNO SM-DP and each UICC. One example of a key exchange algorithm that may be employed is a Diffie-Hellman exchange (ECDH) algorithm.” ¶ 33. Digital Signature: “Generally, target device 106 receives the profile in packets that are transmitted through a node path 110. As part of the outer layer encryption process the packets have a cryptographic signature for each node along the required node path. FIG. 2 illustrates a packet 112 in simplified form, which has cryptographic signatures from each of the intermediate nodes 105 (‘Int. #1 Sig.’ etc.), as well as an end-to-end cryptographic signature (shown as ‘Source Sig.’), which is signed over the payload (e.g., sensitive data), which may also include a profile. While each of the intermediate nodes 108 may cryptographically sign a packet as part of the outer layer encryption process, a required path node can be verified, in some embodiments, without a signature from those of the intermediate nodes that are trusted.” ¶ 19.