

PETITIONER MASTERCARD'S PRESENTATION

IPR2023-01289, against U.S. Patent No. 10,032,171 ("171 Patent")

CONFIDENTIAL

© Copyright Baker Botts 2024. All Rights Reserved.

DEMONSTRATIVE EXHIBIT — NOT EVIDENCE



INTRODUCTION AND BACKGROUND 01

The '171 Patent-Priority Date

 US010032171B2	
(12) United States Patent Yeager	(10) Patent No.: US 10,032,171 B2 (45) Date of Patent: Jul. 24, 2018
(54) SYSTEMS AND METHODS FOR SECURE APPLICATION-BASED PARTICIPATION IN AN INTERROGATION BY MOBILE DEVICE	
(75) Inventor: Douglas C. Yeager , Austin, TX (US)	(58) Field of Classification Search None See application file for complete search history.
(73) Assignee: SimplyTapp, Inc. , Austin, TX (US)	(56) References Cited U.S. PATENT DOCUMENTS
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35	5,636,280 A * 6/1997 Kelly G06F 21/31 5,737,423 A * 4/1998 Mandulev G06K 19/0723

- The '171 Patent's Provisionals do not support its claims
- Patent Owner's stipulated priority date: **Aug. 30, 2012**

(22) Filed: **Aug. 30, 2012**

(65) **Prior Publication Data**

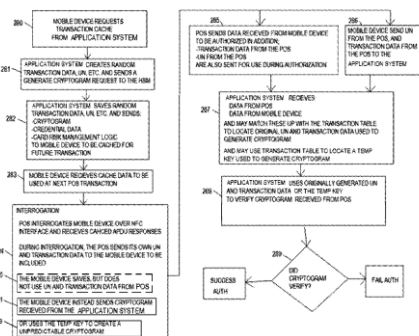
US 2013/0054474 A1 Feb. 28, 2013

Related U.S. Application Data

(60) Provisional application No. 61/685,863, filed on Mar. 26, 2012, provisional application No. 61/573,476, filed on Sep. 6, 2011, provisional application No. 61/575,846, filed on Aug. 30, 2011.

(Continued)

33 Claims, 24 Drawing Sheets



Here, Patent Owner offers a similar stipulation; that is, Patent Owner will not challenge Fiske's status as prior art in the IPR2023-01289 proceeding and will not allege in the IPR2023-01289 proceeding that any of claims 1-33 of the '171 Patent are entitled to a priority date any earlier than the August 30, 2012 filing date of the non-provisional application for the '171 Patent. Based on this stipulation, Petitioner's prediction that

Paper 14

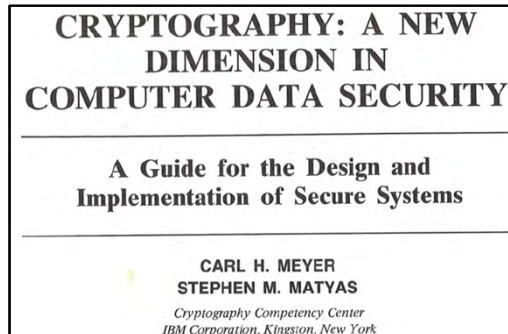
Paper 1 at 21-24

Prior Art Cryptogram/MAC Generation

- Payment textbooks since the 1980s have recommended generation of a Message Authentication Code (MAC) using a cryptographic key with each funds transfer message
- This prevents fraud by preventing someone without the secret key to “introduce a spurious message” into the funds transfer system
 - Only a device with the secret key is allowed to conduct a transaction

Message authentication is a technique which produces cryptographic check digits which are appended to the message. These digits are analogous to a parity check or cyclic redundancy check, except that in this case the check digits are cryptographically generated, using DES and a secret key. These digits, called the message authentication code or MAC, are generated by the originator, appended to the transmitted message, and then checked by the recipient, who also holds the same secret key used in the generation process. Should anyone attempt to modify the message between the time the MAC is generated and the time it is checked, he would be detected. Not knowing the secret key, he would be unable to generate the correct MAC for his modified message. Similarly, no one can successfully introduce a spurious message because he could not generate the proper MAC for this message.

EDP systems. Thus message authentication, by protecting the transaction against fraudulent modifications both over communications lines and within EDP systems, is the recommended approach.



Copyright ©1982 by John Wiley & Sons, Inc.

Ex[1034] 17-18

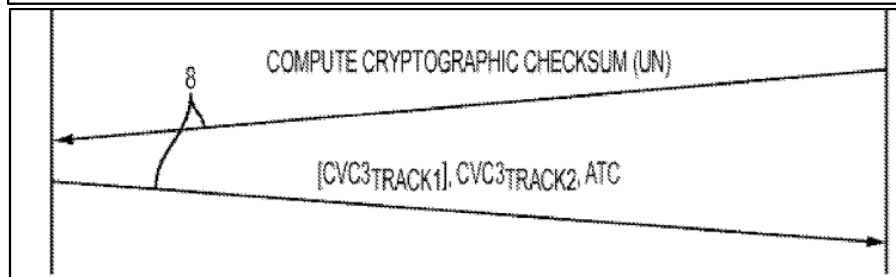
The '171 Patent Uses Standardized, Existing Cryptograms/MACs

'171 Patent:

called a "hash." A cryptogram is typically calculated from a series of input data that comprises some data and a cryptographic key, where the cryptographic key is used to encrypt some data. As used herein the term "cryptographic key" refers to a key used, at least in part, to calculate a cryptogram. A cryptogram is calculated based on an algorithm or calculation formula. As use herein the term "calculation formula" refers to an algorithm or formula that is specified by a card specification as the method to calculate a cryptogram. A calculation formula typically may be known by any entity. The output result of the calculation formula that uses the input data and a cryptographic key is a cryptogram. Examples of cryptograms at the time of this filing are defined by the following card specifications in table 4.

TABLE 4

Card specification	Cryptogram Designator or Name
MASTERCARD PAYPASS MSD	CVC3
VISA Contactless MSD	dCVV
DISCOVER ZIP MSD	dCVV
EMV, VISA QVSDC, VISA VSDC, MASTERCARD MCHIP	<u>AC, TC, AAC, ARQC</u>



Ex[1001] Fig. 2, 25:40-63

2008 EMV Standard (TC, ARQC, etc):

8 Application Cryptogram and Issuer Authentication

The aim of this section is to provide methods for the generation of the Application Cryptograms (TC, ARQC, or AAC) generated by the ICC and the Authorisation Response Cryptogram (ARPC) generated by the issuer and verified by the ICC.

8.1 Application Cryptogram Generation

8.1.1 Data Selection

An Application Cryptogram consists of a Message Authentication Code (MAC) generated over data:

A1.2 Message Authentication Code

The computation of an s-byte MAC ($4 \leq s \leq 8$) is according to ISO/IEC 9797-1 using a 64-bit block cipher ALG in CBC mode. More precisely, the computation of a MAC S over a message MSG consisting of an arbitrary number of bytes with a MAC Session Key Ks takes place in the following steps.

B1.1 Data Encryption Standard (DES)

The double-length key triple DES encipherment algorithm (see ISO/IEC 18033-3) is the approved cryptographic algorithm to be used in the encipherment and MAC mechanisms specified in Annex A1. The algorithm is based on the (single) DES algorithm standardised in ISO 16609.

EX[1020] 99-100, 141, 149

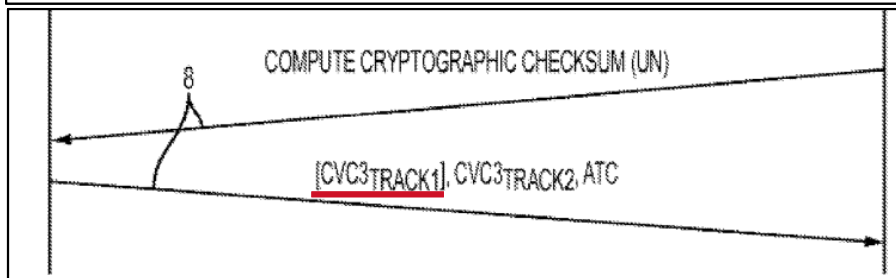
The '171 Patent Uses Standardized, Existing Cryptograms/MACs (Cont'd)

'171 Patent:

called a "hash." A cryptogram is typically calculated from a series of input data that comprises some data and a cryptographic key, where the cryptographic key is used to encrypt some data. As used herein the term "cryptographic key" refers to a key used, at least in part, to calculate a cryptogram. A cryptogram is calculated based on an algorithm or calculation formula. As use herein the term "calculation formula" refers to an algorithm or formula that is specified by a card specification as the method to calculate a cryptogram. A calculation formula typically may be known by any entity. The output result of the calculation formula that uses the input data and a cryptographic key is a cryptogram. Examples of cryptograms at the time of this filing are defined by the following card specifications in table 4.

TABLE 4

Card specification	Cryptogram Designator or Name
<u>MASTERCARD PAYPASS</u> MSD	<u>CVC3</u>
VISA Contactless MSD	dCvv
DISCOVER ZIP MSD	dCvv
EMV, VISA QVSDC, VISA VSDC, MASTERCARD MCHIP	AC, TC, AAC, ARQC



Ex[1001] Fig. 2, 25:40-63

Mastercard Paypass (CVC3):

[0053] In the context of an exemplary specification, such as that for MasterCard PayPass® Mag Stripe products and services, not only would the expanded cryptograms be available, but the additional information needed to verify those cryptograms with full security would also be present (e.g., unpredictable number (UN) and application transaction counter (ATC))—see "extra data" at message 451 in FIG. 4. Thus, when at least an extra ten bytes are available, rather than sending selected bits from the CVC3 cryptograms, it is possible to send two pairs of CVC3s, two bytes each along with the full UN and ATC. Thereby the issuer 2010, if it wishes, can verify two separate two-byte MACs (Message Authentication Codes) which provide, in effect, security approaching that of a four-byte MAC (i.e., security comparable to an EMV chip- and PIN-based transaction). The MasterCard PayPass® M/Chip application uses eight-byte cryptograms for the authorization request. Four-byte cryptograms are generally considered acceptable in the context of EMV.

EX[1016] [0053]

3.8.2 Dynamic CVC3 Generation

The $CVC3_{TRACK1}$ is generated using DES3 encipherment as follows:

Calculate O as follows:

$$O := \text{DES3}(KD_{CVC3})[D]$$

The two least significant bytes of O are the $CVC3_{TRACK1}$.

EX[1058], 109

The '171 Patent

- Patent Owner characterized the invention as the use of a non-permanent (single-use) key for cryptogram generation to conduct payment at an existing, standardized Point of Sale terminal:

Claim 30 addresses the problem of allowing an application running in the operating system of a mobile device to securely participate in interrogations with a POS terminal for digital credential data while retaining low latency. According to Claim 30, “the [interrogable] application [executable in the operating system of the mobile device] does not access a permanent cryptographic key associated with the account during interrogations.” Instead, as shown by the emphasized portions of Claim 30 below, the application generates response cryptograms during interrogations based on a non-permanent key that changes between at least two interrogations.

Ex[1002] 617-618

Claim 30 thus claims a solution that helps address security issues that arise from using an interrogable application executable in the operating system of the mobile device to respond to interrogations at least because the key used to generate response cryptograms changes, limiting the key’s usefulness if stolen by other code running in the operating system of the mobile device.

Moreover, Claim 30 provides this security while reducing latency and network delays experienced by other solutions. In particular, cryptogram generation is done at the mobile device reducing/eliminating the commands/requests that the mobile device sends between a POS and remote secure element. For example, the Specification describes that:

Also, sharing a temporary cryptographic key with the mobile device wherein the mobile device calculates an unpredictable cryptogram would be an alternative using a predetermined cryptogram to achieve a similar system performance result with the context of 100% transaction caching.

Application at ¶ 152.

- Notably, the non-permanent key was **not** in the provisionals

The '171 Patent

- Non-permanent/changing keys for MAC/Cryptogram generation were already the standard in the art:

'171 Patent:

As used herein the term “temporary cryptographic key” refers to a cryptographic key that changes (or is dynamic) from one interrogation to the next over the life of a digital credential (e.g., an SE representation). A temporary crypto-

Ex[1001] 27:33-36

2008 EMV Standard’s “temporary key” based on an incrementing transaction counter:

8.1.2 Application Cryptogram Algorithm

The method for Application Cryptogram generation takes as input a unique 16-byte ICC Application Cryptogram Master Key MK_{AC} and the data selected as described in section 8.1.1, and computes the 8-byte Application Cryptogram in the following two steps:

1. Use the session key derivation function specified in Annex A1.3 to derive a 16-byte Application Cryptogram Session Key SK_{AC} from the ICC Application Cryptogram Master Key MK_{AC} and the 2-byte Application Transaction Counter (ATC) of the ICC.
2. Generate the 8-byte Application Cryptogram by applying the MAC algorithm specified in Annex A1.2 to the data selected and using the 16-byte Application Cryptogram Session Key derived in the previous step.

D3 Application Transaction Counter Considerations

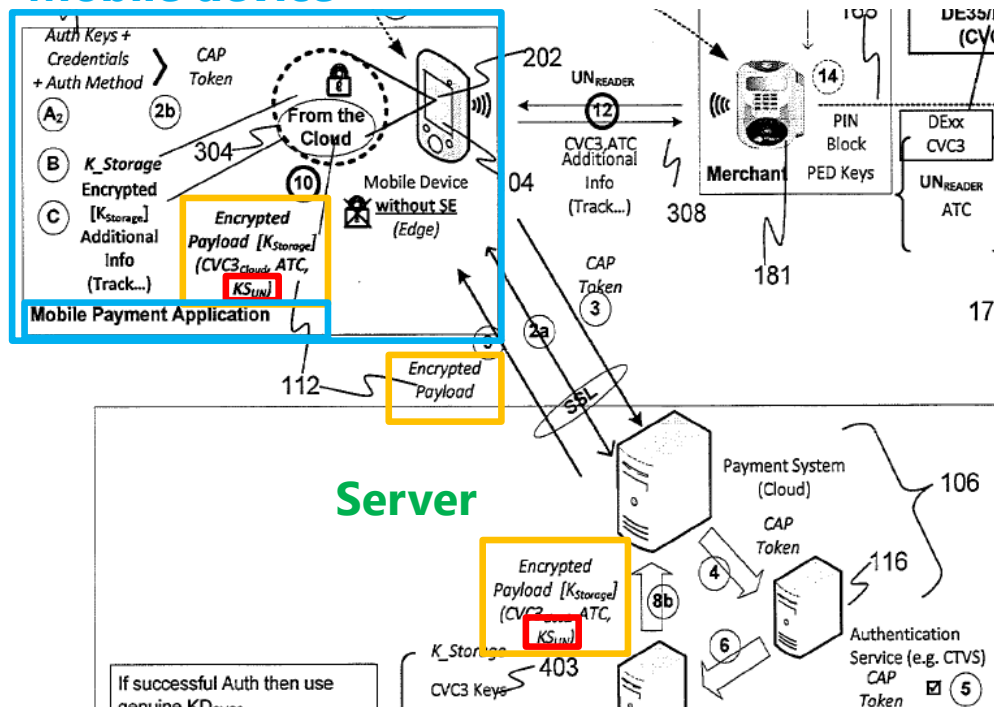
This specification describes a two byte (16 bit) counter (the ATC) that is incremented during each transaction from a nominal starting value of '0000' to a maximum of 'FFFF'. With one increment per card session it gives an expected card life of 65,535 transactions.

Ex[1020] §8.1.2, AnnexD3 p.101, 163

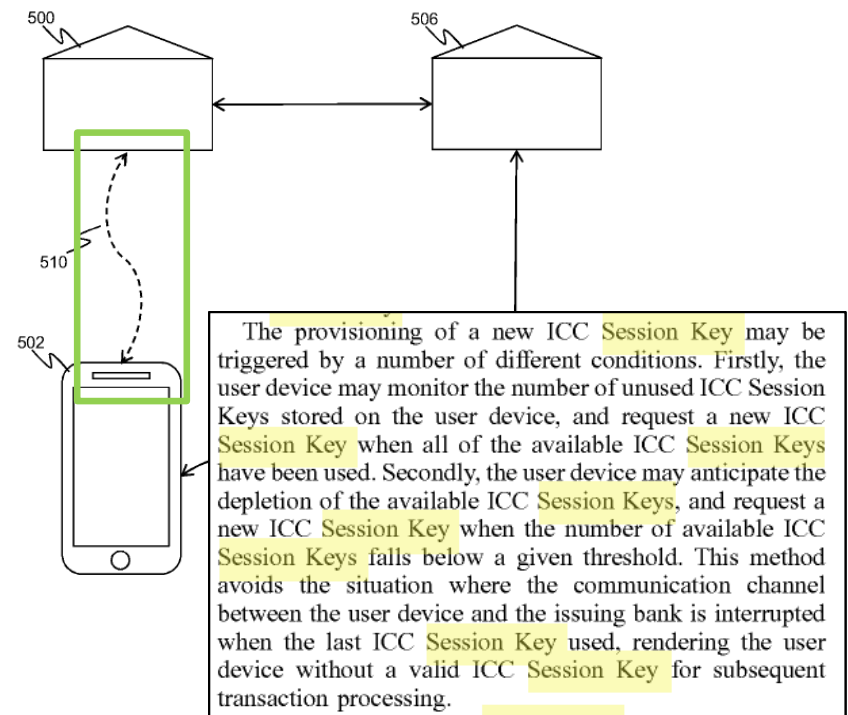
The Prior Art – Collinge and Fiske

- The idea of sending a single-use/session key to a phone from a server was disclosed by both Collinge and Fiske as their inventions.
 - For Collinge, it is the KSUN
 - For Fiske, it is EMV's ICC Session Key

Mobile device



Collinge, Ex[1009] Fig. 6



Fiske Ex[1012] Fig. 5, 14:21-34

BACKGROUND ART 02

EMV and Associated Payment Standards Must Be Considered For Their Background Context

- Petitioner expressly relied upon the background knowledge of the person skilled in the art in its grounds:

Ground	Patent Claims	Basis for Challenge
2	1-33	Collinge renders the claims obvious under § 103 in view of <u>the knowledge of a POSA</u>
4	1-33	Fiske renders the claims obvious under § 103 in view of <u>the knowledge of a POSA</u>

Paper 1 at 13

- Standards are well-established background art to be considered in an IPR obviousness analysis (Paper 26, at 7, 25)
 - “we do not see why, even if the CableLabs specification is not incorporated by reference in Gonder itself, it cannot be considered as showing the background knowledge that a [POSITA] would have had in reading Gonder.” *Dish Network L.L.C. v. Broadband Itv, Inc.*, IPR2020-01267, Paper 25 at 15 (P.T.A.B. Apr. 1, 2021)
- The card payment industry is highly standardized. (See Pet. at 16-18). Therefore, a POSITA would need to be familiar with standard practices to understand and implement these techniques—it is not relevant whether any part of the standard is “mandatory.” (Paper 26, at 1, 7, 8; Ex[1050], ¶¶ 20-21.)

EMV and Associated Payment Standards Must Be Considered For Their Background Context



Ariosa Diagnostics v. Verinata Health, Inc., 805 F.3d 1359 (Fed. Cir. 2015)

"...that **the Board was declining to consider Exhibit 1010, even as evidence of the background understanding of skilled artisans** ...simply because the brochure had not been identified at the petition stage as one of the pieces of prior art defining a combination for obviousness. **If that is what the Board meant, the Board erred.** **[Background] Art can legitimately serve to document the knowledge that skilled artisans would bring to bear in reading the prior art identified as producing obviousness"**

Paper 26 at 4-5, 7



Randall Mfg. v. Rea, 733 F.3d 1355 (Fed. Cir. 2013)

"...the Supreme Court criticized a rigid approach to determining obviousness based on the disclosures of individual prior-art references, with little recourse to the knowledge, creativity, and common sense that an ordinarily skilled artisan would have brought to bear when considering combinations or modifications.... **Rejecting a blinkered focus on individual documents, the Court required an analysis that reads the prior art in context, taking account of ... 'the background knowledge** possessed by a person having ordinary skill in the art,'..."

Paper 26 at 4-5, 7

EMV and Associated Payment Standards

- The '171 Patent (and the Collinge and Fiske references) presume reader's background knowledge of the EMV standard and EMV-compliant payment standards such as PayPass:

'171 Patent:

50 The details of an interrogation between a point-of-sale and a digital credential, card, SE representation, and mobile device are defined in "card specifications" and implemented in point-of-sale terminals. Examples of these card specifications at the time of this filing are MASTERCARD PAY-PASS-Mag Stripe v3.3, Contactless EMV, DISCOVER NETWORK RF Contactless Specification-Supplement Guide For Functional Support, AMEX ALIS v1.7, and VISA Contactless Payment Specification v2.0.2. As used herein

Depending on the interrogation sequence selected by the point-of-sale, it may not always be possible to implement a system set forth in FIG. 17. Many point-of-sale interrogation sequences carried out at the time of this filing are based on card specifications such as:

- MASTERCARD, PAYPASS, MChip
- AMERICAN EXPRESS, EXPRESS PAY
- DISCOVER ZIP
- Visa QVSDC, VSDC
- Contactless EMV

60

Ex[1001] 26:50-59, 34:55-60

Collinge:

As used herein, in an embodiment, a transaction is distinguished from an authentication transaction, which is used to get access to payment credentials managed in the cloud-based transaction data generation system) and the payment transaction (i.e., a standard PayPass® Magstripe transaction). For example, it

10 (KS_{UN}, Cloud_CVC3_{TRACK1/2}, ATC) and the additional assets are stored.
The Mobile Payment Application is ready to support a PayPass Magstripe Payment using a Mobile Device

11 Standard shopping experience.

12 Standard PayPass Magstripe payment experience using a Mobile Device.

The Mobile Payment Application use a specific process to generate the CVC3 using (KS_{UN}, Cloud_CVC3_{TRACK1/2}, ATC and UN_{READER})

Ex[1009] 14, 22

Fiske:

55 Hence, it is an objective of the present invention to provide improved methods for providing payment card functionality on a user device, while minimizing any inconvenience to the user of the device and without requiring modification of the EMV infrastructure, or existing PoS hardware.

Ex[1012], 2:54-59

The '171 Patent

- The claimed invention in the '171 Patent relies on existing payment standards to operate, e.g. the patent references "PayPass," an EMV-compliant Mastercard payment specification, **47 times** throughout the specifications, and in multiple figures, both as background and for implementation of the invention, e.g.:

Col. 7:

CARD **PAYPASS** Mag Stripe application running on an SE, according to "Mag Stripe.pdf" (Ver. 3.3) presently available at <http://www.PayPass.com/approved-products.html>. The term "digital credential data" as used herein refers to at least a portion of all data, including any data that are required to calculate the data, that are necessary to authorize approval of a transaction at a POS terminal.

FIG. 2 describes the detailed commands and responses expected when a reader interrogates a MASTERCARD **PAYPASS** Mag Stripe Ver. 3.3 application running on a SE. This sequence is provided in the MASTERCARD **PAYPASS** card specification. Shown in FIG. 2 are communications 5, 6, 7, 8 and 9 with details including abbreviations and acronyms explained subsequently herein.

FIG. 3 describes example transactions at the ISO/IEC 7816-4 Application Protocol Data Unit (APDU) protocol level between a reader and a card. The transaction processing depicted in FIG. 3 is representative of the MASTERCARD **PAYPASS** Mag Stripe application card specification v. 3.3. The notation ">CARD" denotes commands to the card and the notation "CARD>" denotes responses from the card. Communication 9 defines some of the initial parameters used in the transaction which include an unpredictable number issued by the reader, a transaction counter maintained on the SE. Communication 10 describes the command that the reader issues to the SE to find out what applications exist on the SE. This example shows that the MASTERCARD **PAYPASS** application responds as an active one. Next, communication 11 shows how the reader commands the SE to properly select the **PAYPASS** application for further processing. Communication 12 shows how the reader receives the type of processing options that the **PAYPASS** application supports. Communication 13 shows how the reader commands and then receives the **PAYPASS** application specific credential information about its card holder including card account number, holder name, expiration date, and more. Communication 14 shows how the

Col. 14:

It is important to note that at the ISO/IEC 7816-4 (APDU) level, an entire interrogation similar to those described in FIG. 1, and FIG. 2 may be carried out from a RFID POS reader 21 through an NFC enabled mobile device 626, through a data connection over the internet 35 to a remote system 631 that contains the actual SE 634. An example of reader commands is illustrated in element 638 as a **PAYPASS** reader. The **PAYPASS** reader 638 shows an exemplary sequence of ISO/IEC 7816-4 APDU commands and responses that may be pushed through the reader 21 and end up at the remote SE 634. Element 637 shows an example of APDU commands that are received by the remote SE 634 and the appropriate response that is issued as defined by the 7816-4 application for MASTERCARD **PAYPASS**. Elements 638 and 637 show how FIG. 2 may be broken up so that the **PAYPASS** card may be hosted on a remote SE 634 on a remote system 631 and show that the **PAYPASS** reader may be an RFID POS 621.

Col. 47:

In some embodiments disclosed herein, a specific payment standard issued from MASTERCARD International. This standard is based on reference documentation published by **PAYPASS**—Mag Stripe (V3.3).pdf and other derivations. In this card specification, **PAYPASS** contactless payment card reader and card interrogation is documented. This document discusses specifically how an RFID reader interrogator would interact with a card containing a SE to extract and build Track1/Track2 equivalent data that is compatible for existing processing infrastructure, but contains the more secure and dynamic aspects of SE driven credential data.

Col. 25:

Examples of cryptograms at the time of this filing are defined by the following card specifications in table 4.

TABLE 4	
Card specification	Cryptogram Designator or Name
MASTERCARD PAYPASS MSD	CVC3
VISA Contactless MSD	dCVV
DISCOVER ZIP MSD	dCVV
EMV, VISA QVSDC, VISA VSDC,	AC, TC, AAC, ARQC
MASTERCARD MCHIP	

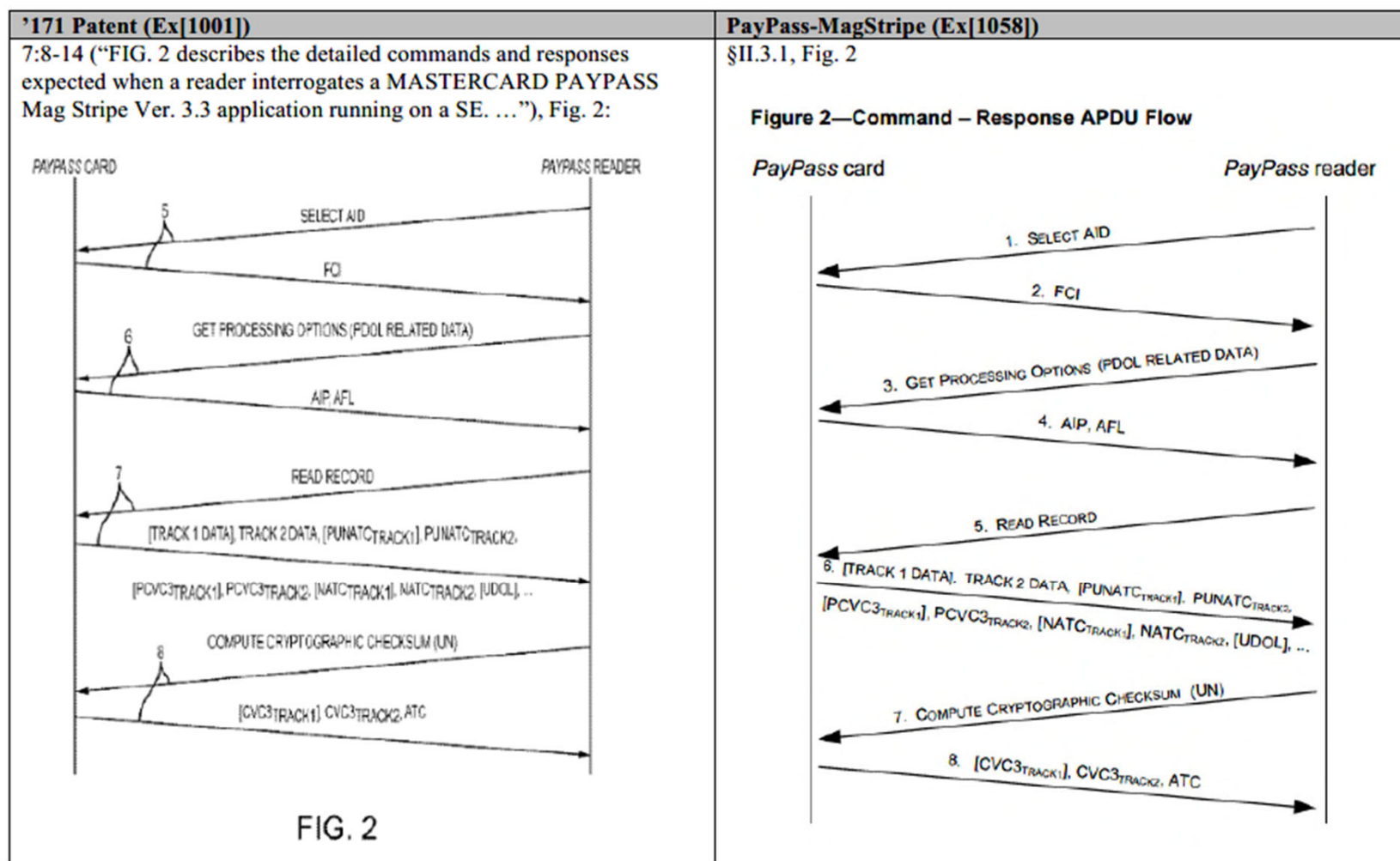
Col. 31:

system set forth in FIG. 17. Many point-of-sale interrogation sequences carried out at the time of this filing are based on card specifications such as:

- MASTERCARD, **PAYPASS**, MChip
- AMERICAN EXPRESS, EXPRESS PAY
- DISCOVER ZIP
- Visa QVSDC, VSDC
- Contactless EMV

The '171 Patent

- The '171 Patent Outright Copies MasterCard's PayPass-MagStripe Specification, Because That is What It Was Trying to Implement



The '171 Patent

- The '171 Patent Outright Copies MasterCard's PayPass-MagStripe Specification, Because That is What It Was Trying to Implement

'171 Patent (Ex[1001])	PayPass-MagStripe (Ex[1058])
7:15-46 ("FIG. 3 describes example transactions at the ISO/IEC 7816-4 Application Protocol Data Unit (APDU) protocol level between a reader and a card. The transaction processing depicted in FIG. 3 is representative of the MASTERCARD PAYPASS Mag Stripe application card specification v. 3.3. ...), Fig. 3: <pre> 9 { RUN: 3 UN: 00 00 08 99 ATC: 00 5E CVC3TRACK1: 2 LEAST SIGNIFICANT BYTES OF DES3(KDCVC3) [77 80 00 00 08 99 00 5E] CVC3TRACK1: FB C7 CVC3TRACK2: 2 LEAST SIGNIFICANT BYTES OF DES3(KDCVC3) [99 D3 00 00 08 99 00 5E] CVC3TRACK2: B8 92 10 { • SELECT PPSE COMMAND >CARD: 00 A4 04 00 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 00 CARD>: 6F 23 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 11 BP 0C 0E 61 0C 4F 07 A0 00 00 00 04 10 10 87 01 01 90 00 11 { • SELECT AID COMMAND >CARD: 00 A4 04 00 07 A0 00 00 00 04 10 10 00 CARD>: 6F 17 84 07 A0 00 00 00 04 10 10 A5 0C 50 0A 4D 61 73 74 65 72 43 61 72 64 90 00 12 { • GET PROCESSING OPTIONS COMMAND >CARD: 80 A8 00 00 02 83 00 00 CARD>: 77 0A 82 02 00 00 94 04 08 01 01 00 90 00 13 { • READ RECORD COMMAND >CARD: 00 B2 01 0C 00 CARD>: 70 7F 9F 6C 02 00 01 56 3E 42 35 34 31 33 31 32 33 34 35 36 37 38 34 38 30 30 5E 53 55 50 50 4C 49 45 44 2F 4E 4F 54 5E 30 39 30 36 31 30 31 33 33 30 30 30 33 33 30 30 30 32 32 32 32 30 30 30 30 31 31 31 30 9F 64 01 03 9F 62 06 00 00 00 38 00 00 9F 63 06 00 00 00 00 30 E0 9F 65 02 00 0E 9F 66 02 0E 70 9F 6B 13 54 13 12 34 56 78 48 00 D0 90 61 01 90 00 99 00 00 00 0F 9F 67 01 03 90 00 14 { • COMPUTE CRYPTOGRAPHIC CHECKSUM COMMAND >CARD: 80 2A 8E 80 04 00 00 08 99 00 CARD>: 77 0F 9F 61 02 B8 92 9F 60 02 FB C7 9F 36 02 00 5E 90 00 </pre> FIG. 3	§A.4 A.4 Transaction Processing <pre> nUN: 3 UN: 00 00 08 99 ATC: 00 5E CVC3TRACK1: 2 least significant bytes of DES3(KDCVC3) [77 80 00 00 08 99 00 5E] CVC3TRACK1: FB C7 CVC3TRACK2: 2 least significant bytes of DES3(KDCVC3) [99 D3 00 00 08 99 00 5E] CVC3TRACK2: B8 92 • SELECT PPSE Command >CARD: 00 A4 04 00 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 00 CARD>: 6F 23 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 11 BP 0C 0E 61 0C 4F 07 A0 00 00 00 04 10 10 87 01 01 90 00 • SELECT AID Command >CARD: 00 A4 04 00 07 A0 00 00 00 04 10 10 00 CARD>: 6F 17 84 07 A0 00 00 00 04 10 10 A5 0C 50 0A 4D 61 73 74 65 72 43 61 72 64 90 00 • GET PROCESSING OPTIONS Command >CARD: 80 A8 00 00 02 83 00 00 CARD>: 77 0A 82 02 00 00 94 04 08 01 01 00 90 00 • READ RECORD Command >CARD: 00 B2 01 0C 00 CARD>: 70 7F 9F 6C 02 00 01 56 3E 42 35 34 31 33 31 32 33 34 35 36 37 38 34 38 30 30 5E 53 55 50 50 4C 49 45 44 2F 4E 4F 54 5E 30 39 30 36 31 30 31 33 33 30 30 30 33 33 30 30 30 32 32 32 32 30 30 30 30 31 31 31 30 9F 64 01 03 9F 62 06 00 00 00 38 00 00 9F 63 06 00 00 00 00 30 E0 9F 65 02 00 0E 9F 66 02 0E 70 9F 6B 13 54 13 12 34 56 78 48 00 D0 90 61 01 90 00 99 00 00 00 0F 9F 67 01 03 90 00 • COMPUTE CRYPTOGRAPHIC CHECKSUM Command >CARD: 80 2A 8E 80 04 00 00 08 99 00 CARD>: 77 0F 9F 61 02 B8 92 9F 60 02 FB C7 9F 36 02 00 5E 90 00 </pre>

The '171 Patent

- The '171 Patent Outright Copies MasterCard's PayPass-MagStripe Specification, Because That is What It Was Trying to Implement

'171 Patent (Ex[1001])	PayPass-MagStripe (Ex[1058])
29:30-39 with each interrogation by a mobile device. The CVC3 is cryptogram that is expected to be calculated with this algorithm from the MASTERCARD PAYPASS MSD v3.3 card specification: First KDevc3 is created: $Z_L := \text{DES3}(\text{IMK}_{\text{cvc3}})[Y]$ $Z_R := \text{DES3}(\text{IMK}_{\text{cvc3}})[Y \oplus ('FF' \ 'FF' \ 'FF' \ 'FF' \ 'FF' \ 'FF' \ 'FF' \ 'FF')]$ and define: $\text{KD}_{\text{cvc3}} := (Z_L \ Z_R).$	§III.3.8.1 3.8.1 KD_{CVC3} This section specifies the key derivation method used to generate the ICC Derived Key for CVC3 Generation (KD_{CVC3}). KD_{CVC3} is a 16-byte DES3 key derived from the Issuer Master Key for CVC3 Generation (IMK_{CVC3}) as follows: 1. Concatenate from left to right the PAN (without any 'F' padding) with the PAN sequence number⁷ (if the PAN sequence number is not available, then it is replaced by a '00' byte). If the result X is less than 16 digits long, pad it to the left with hexadecimal zeros in order to obtain an eight-byte number Y in numeric (n) format. If X is at least 16 digits long, then Y consists of the 16 rightmost digits of X in numeric (n) format. 2. Compute the two eight-byte numbers: $Z_L := \text{DES3}(\text{IMK}_{\text{CVC3}})[Y]$ $Z_R := \text{DES3}(\text{IMK}_{\text{CVC3}})[Y \oplus ('FF' \ 'FF' \ 'FF' \ 'FF' \ 'FF' \ 'FF' \ 'FF' \ 'FF')]$ and define : $\text{KD}_{\text{CVC3}} := (Z_L \ Z_R).$

The '171 Patent

- The '171 Patent Outright Copies MasterCard's PayPass-MagStripe Specification, Because That is What It Was Trying to Implement

'171 Patent (Ex[1001])	PayPass-MagStripe (Ex[1058])																																																																																																																																																																																																																																
7:49-63 ("FIG. 4 describes how the reader 21 in FIG. 5 formats the response data received from the SE into accepted Track1 and Track2 data strings that are used in existing processing systems. ... "), Fig. 4: THE PAYPASS READER GENERATES THE TRACK 2 DATA TO BE INCLUDED IN THE AUTHORIZATION MESSAGE AS FOLLOWS: CVC3_{TRACK2} IN BASE 10: 47250 ATC IN BASE 10: 94 => TRACK 2 DATA: <table> <tr> <td>54</td><td>13</td><td>12</td><td>34</td><td>56</td><td>78</td><td>48</td><td>00</td><td>D0</td><td>90</td><td>61</td><td>01</td> </tr> <tr> <td>9</td><td>094</td><td>99</td><td>899</td><td>250</td><td></td><td></td><td>3</td><td>F</td><td></td><td></td><td></td> </tr> <tr> <td>↑↑↑</td><td>↑↑↑</td><td>↑↑↑</td><td>↑↑↑</td><td>↑</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>ATC</td><td>UN</td><td>CVC3_{TRACK2}</td><td>UN</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> THE PAYPASS READER GENERATES THE TRACK 1 DATA TO BE INCLUDED IN THE AUTHORIZATION MESSAGE AS FOLLOWS: CVC3_{TRACK1} IN BASE 10: 64455 ATC IN BASE 10: 94 => TRACK 1 DATA: <table> <tr> <td>42</td><td>35</td><td>34</td><td>31</td><td>33</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>34</td><td>38</td><td>30</td> </tr> <tr> <td>30</td><td>5E</td><td>53</td><td>55</td><td>50</td><td>50</td><td>4C</td><td>49</td><td>45</td><td>44</td><td>2F</td><td>4E</td><td>4F</td><td>54</td><td>5E</td><td>30</td> </tr> <tr> <td>39</td><td>30</td><td>36</td><td>31</td><td>30</td><td>31</td><td>33</td><td>33</td><td>34</td><td>35</td><td>35</td><td>33</td><td>33</td><td>33</td><td>30</td><td>39</td> </tr> <tr> <td>34</td><td>32</td><td>32</td><td>32</td><td>32</td><td>32</td><td>38</td><td>39</td><td>39</td><td>31</td><td>31</td><td>31</td><td>31</td><td>31</td><td>31</td><td>33</td> </tr> </table> ASCII DUMP: B5413123456784800^SUPPLIED/NOT^0906101 334553330942222289911113 ↑↑↑ ↑↑↑ ↑↑↑ ↑ CVC3_{TRACK1} ATC UN UN	54	13	12	34	56	78	48	00	D0	90	61	01	9	094	99	899	250			3	F				↑↑↑	↑↑↑	↑↑↑	↑↑↑	↑								ATC	UN	CVC3 _{TRACK2}	UN									42	35	34	31	33	31	32	33	34	35	36	37	38	34	38	30	30	5E	53	55	50	50	4C	49	45	44	2F	4E	4F	54	5E	30	39	30	36	31	30	31	33	33	34	35	35	33	33	33	30	39	34	32	32	32	32	32	38	39	39	31	31	31	31	31	31	33	§A.4 The PayPass reader generates the track 2 data to be included in the authorization message as follows: CVC3_{TRACK2} in base 10: 47250 ATC in base 10: 94 => track 2 data: <table> <tr> <td>54</td><td>13</td><td>12</td><td>34</td><td>56</td><td>78</td><td>48</td><td>00</td><td>D0</td><td>90</td><td>61</td><td>01</td> </tr> <tr> <td>9</td><td>094</td><td>99</td><td>899</td><td>250</td><td></td><td></td><td>3</td><td>F</td><td></td><td></td><td></td> </tr> <tr> <td>↑↑↑</td><td>↑↑↑</td><td>↑↑↑</td><td>↑↑↑</td><td>↑</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>ATC</td><td>UN</td><td>CVC3_{TRACK2}</td><td>UN</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> The PayPass reader generates the track 1 data to be included in the authorization message as follows: CVC3_{TRACK1} in base 10: 64455 ATC in base 10: 94 => track 1 data: <table> <tr> <td>42</td><td>35</td><td>34</td><td>31</td><td>33</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>34</td><td>38</td><td>30</td> </tr> <tr> <td>30</td><td>5E</td><td>53</td><td>55</td><td>50</td><td>50</td><td>4C</td><td>49</td><td>45</td><td>44</td><td>2F</td><td>4E</td><td>4F</td><td>54</td><td>5E</td><td>30</td> </tr> <tr> <td>39</td><td>30</td><td>36</td><td>31</td><td>30</td><td>31</td><td>33</td><td>33</td><td>34</td><td>35</td><td>35</td><td>33</td><td>33</td><td>33</td><td>30</td><td>39</td> </tr> <tr> <td>34</td><td>32</td><td>32</td><td>32</td><td>32</td><td>32</td><td>38</td><td>39</td><td>39</td><td>31</td><td>31</td><td>31</td><td>31</td><td>31</td><td>31</td><td>33</td> </tr> </table> ASCII dump: B5413123456784800^SUPPLIED/NOT^0906101 334553330942222289911113 ↑↑↑ ↑↑↑ ↑↑↑ ↑ CVC3_{TRACK1} ATC UN UN	54	13	12	34	56	78	48	00	D0	90	61	01	9	094	99	899	250			3	F				↑↑↑	↑↑↑	↑↑↑	↑↑↑	↑								ATC	UN	CVC3 _{TRACK2}	UN									42	35	34	31	33	31	32	33	34	35	36	37	38	34	38	30	30	5E	53	55	50	50	4C	49	45	44	2F	4E	4F	54	5E	30	39	30	36	31	30	31	33	33	34	35	35	33	33	33	30	39	34	32	32	32	32	32	38	39	39	31	31	31	31	31	31	33
54	13	12	34	56	78	48	00	D0	90	61	01																																																																																																																																																																																																																						
9	094	99	899	250			3	F																																																																																																																																																																																																																									
↑↑↑	↑↑↑	↑↑↑	↑↑↑	↑																																																																																																																																																																																																																													
ATC	UN	CVC3 _{TRACK2}	UN																																																																																																																																																																																																																														
42	35	34	31	33	31	32	33	34	35	36	37	38	34	38	30																																																																																																																																																																																																																		
30	5E	53	55	50	50	4C	49	45	44	2F	4E	4F	54	5E	30																																																																																																																																																																																																																		
39	30	36	31	30	31	33	33	34	35	35	33	33	33	30	39																																																																																																																																																																																																																		
34	32	32	32	32	32	38	39	39	31	31	31	31	31	31	33																																																																																																																																																																																																																		
54	13	12	34	56	78	48	00	D0	90	61	01																																																																																																																																																																																																																						
9	094	99	899	250			3	F																																																																																																																																																																																																																									
↑↑↑	↑↑↑	↑↑↑	↑↑↑	↑																																																																																																																																																																																																																													
ATC	UN	CVC3 _{TRACK2}	UN																																																																																																																																																																																																																														
42	35	34	31	33	31	32	33	34	35	36	37	38	34	38	30																																																																																																																																																																																																																		
30	5E	53	55	50	50	4C	49	45	44	2F	4E	4F	54	5E	30																																																																																																																																																																																																																		
39	30	36	31	30	31	33	33	34	35	35	33	33	33	30	39																																																																																																																																																																																																																		
34	32	32	32	32	32	38	39	39	31	31	31	31	31	31	33																																																																																																																																																																																																																		

FIG. 4

EMV and Associated Payment Standards

- Numerous contemporaneous sources confirm that a POSITA working with card or mobile device payment systems would be very familiar with EMV and PayPass specifications

It should be noted that the skilled artisan will be familiar with the EMV specifications. Nevertheless, out of an abundance of caution, the following documents are expressly incorporated herein by reference in their entirety for all purposes (the same are published by EMVCo and available on EMVCo's web site):

"Mechanism to allow the use of disposable cards on a system designed to accept cards conforming to the standards of the global payments industry", Ex[1052] 4:56-57

The preferred embodiment includes an implementation of the EMV system. It is appreciated that someone of ordinary skill in the art would be familiar with the specifications of the EMV system which were referenced earlier. The preferred embodiment implements the EMV system and incorporates several improvements, among others a dynamic off-line authentication of the data sent by the card to the terminal.

"Multi-purpose transaction card system", Ex[1055] 45:33-35

It should be noted that the skilled artisan will be familiar with the EMV specifications. Nevertheless, out of an abundance of caution, the following documents are expressly incorporated herein by reference in their entirety for all purposes (the same are published by EMVCo and available on EMVCo's web site):

need to read from you and what process shall we use?" The various commands and arguments are familiar to the skilled artisan from the EMV, PAYPASS® M/CHIP® & PAYPASS® MAGSTRIPE standards (registered marks of MasterCard International Incorporated, Purchase, N.Y., USA). Step 306

"Apparatus, method, and computer program product for recovering torn smart payment device transactions", Ex[1053] 5:14-40, 10:47-52

EMV issuer back office 104 over an EMV interface 106. The EMV issuer back office 104, the EMV interface 106, and the various support structures therefor are well-known to persons having ordinary skill in the art and will not be described here. The EMV card-reader terminal 102, on the other hand,

"EMV Transactions in Mobile Terminals", Ex[1054] 3:33-44

EMV and Associated Payment Standards

- Numerous contemporaneous sources confirm and corroborate that a POSITA working with card or mobile device payment systems would be very familiar with EMV and PayPass specifications (cont'd)

[0045] By way of an example to aid understanding of the skilled artisan, one example of a payment infrastructure is the EMV infrastructure, i.e. a payments system incorporating EMV, such as that operated by MasterCard International Inc. in conjunction with Issuers, Acquirers, and Merchants. Further, one example of a payment infrastructure is an Automatic Fare Collection (AFC) system. Optionally, payment module 212 can be further configured to interrogate the electronic payment device 210 to obtain profile data pertaining to a holder of the electronic payment device. In this

"Apparatus and method for integrated payment and electronic merchandise transfer" Ex[1056] [0045]

[0287] The preferred embodiment includes an implementation of the EMV system. It is appreciated that someone of ordinary skill in the art would be familiar with the specifications of the EMV system which were referenced earlier.

"Multi-purpose transaction card system",
Ex[1057] [0287]

occur. This software is implemented using based upon known knowledge of mobile device 110 internals and application platforms, NFC, smartcard internals and application platforms, payment protocols (e.g. PayPass), and the working/workflow associated with POS and POE terminals, and the transaction and management servers. In addition, the

The present invention can also be interfaced with certain known and implanted payment protocols, such as Paypass.

"Using a mobile device for point of entry
NFC transactions", Ex[1061] 6:60-66, 7:40-60

chants, and charge cards. Valued identification may be in the form of a card or other media that contains valued identification information, such as for example: magnetic strips; integrated circuits; computer readable mediums; contactless media using near field communications technologies, such as, for example, radio frequency identification tags, PayPass™, contactless payment stickers, mobile communication devices; or other storage media known to one skilled in the art. The valued identification information may include

"Systems and methods for reducing the
unauthorized resale of event tickets",
Ex[1062] [0134]

EMV and Associated Payment Standards

- It was well known that Mastercard's PayPass specification, recited in Collinge and the '171 Patent, is an EMV-compliant standard.

PayPass Specification:

MasterCard provides this manual for members and their authorized agents. Specifically, the following personnel will find this manual useful:

- Customers and vendors who want to implement the *PayPass* – Mag Stripe application on a card or reader
- Those involved in type approval services and in verifying actual implementations against these specifications



Note

It is assumed that readers already have an understanding of chip card technology in general, and of ISO/IEC 14443, ISO/IEC 7816-4 and EMV in particular.

PayPass Spec., Ex[1058], at 9

All data objects available for personalization are stored in persistent memory of the card and are listed in *[M/CHIP4]*. This section specifies only the specific personalization requirements for the *PayPass* – M/Chip 4 application.



Note When the *PayPass* – M/Chip 4 application is personalized according to the EMV Card Personalization Specification as defined in *[M/CHIP4 CPS]*, then the data groupings for the *PayPass* specific data objects are as listed in Annex B.

PayPass Spec., Ex[1059], at 35

EMV and Associated Payment Standards

- In addition, as demonstrated by Petitioner's expert and corroborated by PayPass specifications themselves, it was well known to a POSITA that PayPass recited in Collinge is an EMV-compliant standard.

Aspect / Notes	PayPass-MagStripe Ex[1058]	EMV (MagStripe Mode)
Transaction flow Shows both specifications perform a select AID and a GPO to determine whether the transaction will continue in Mag stripe mode or in EMV mode.	From § II.3.1 at 23-24 1. The <i>PayPass</i> reader issues the SELECT AID command to select the <i>PayPass</i> – Mag Stripe application. 2. The card responds with the <i>FCI</i> . The <i>FCI</i> may contain the <i>PDOL</i> . The <i>PDOL</i> is a list of tags and lengths of reader-resident data objects needed by the <i>PayPass</i> card in the GET PROCESSING OPTIONS command. 3. The <i>PayPass</i> reader issues the GET PROCESSING OPTIONS command. If there is no <i>PDOL</i> in the <i>FCI</i> , then the <i>PayPass</i> reader uses the command data field '8300'. Otherwise the command data field contains a <i>Command Template</i> (tag '83') comprising the concatenated list of data objects resulting from processing the <i>PDOL</i> . 4. The card returns the <i>AIP</i> and the <i>AFL</i> . The <i>AFL</i> has a fixed value indicating that all data to be read by the <i>PayPass</i> reader are included in record 1 of the file with SFI 1	From EMV Book B (Ex[1024]): §3.3.3.4 Entry Point shall send the SELECT (AID) command with the ADF Name of the selected Combination (with Extended Selection if appended). From EMV Book C-2 (Ex[1026]): §3.4 Upon receipt of an ACT Signal, the Kernel initiates the transaction on the Card through a GET PROCESSING OPTIONS command. Based on the response from the Card, in particular the <i>Application Interchange Profile</i> , the Kernel continues with either a mag-stripe mode or an EMV mode transaction.

Fiske Suppl. Decl., Ex[1050], Annex A

EMV and Associated Payment Standards

- In addition, as demonstrated by Petitioner's expert and corroborated by PayPass specifications themselves, it was well known to a POSITA that PayPass recited in Collinge is an EMV-compliant standard.

Aspect / Notes	PayPass-MagStripe Ex[1058]	EMV (MagStripe Mode)
	From §II.3.2: 3.2.1.1 The <i>PayPass</i> reader must perform the following logical functions in the order shown: 1. Get Processing Options Processing as specified in Section 3.4.1 of PART II. 2. Read Application Data as specified in Section 3.4.2 of PART II. 3. Application Version Number Checking as specified in Section 3.4.3 of PART II. 4. Compute Cryptographic Checksum Processing as specified in Section 3.4.4 of PART II. 5. Completion as specified in Section 3.4.5 of PART II.	From <i>id.</i> §3.4.2: For a mag-stripe mode transaction, after the GET PROCESSING OPTIONS command, the Kernel continues with the following steps: 1. It reads the data records from the Card, containing <i>Track 1 Data</i> and <i>Track 2 Data</i>, together with instruction on how to populate the discretionary data. 2. It issues the COMPUTE CRYPTOGRAPHIC CHECKSUM command, including <i>Unpredictable Number (Numeric)</i> to the Card, requesting the Card to return a CVC3 cryptogram, calculated over <i>Unpredictable Number (Numeric)</i>. 3. It populates the <i>Track 2 Data</i> with the <i>Unpredictable Number (Numeric)</i>, the <i>Application Transaction Counter</i>, and <i>CVC3 (Track2)</i>. 4. If <i>Track 1 Data</i> is present, it populates the <i>Track 1 Data</i> with the <i>Unpredictable Number (Numeric)</i>, the <i>Application Transaction Counter</i>, and <i>CVC3 (Track1)</i>. 5. It sets <i>nUN</i> equal to the (meaningful) length of the <i>Unpredictable Number (Numeric)</i> and populates <i>Track 2 Data</i> and (if present) <i>Track 1 Data</i> with this value. 6. It requests the transaction to be sent online.

Fiske Suppl. Decl., Ex[1050], Annex A

EMV and Associated Payment Standards

- In addition, as demonstrated by Petitioner's expert and corroborated by PayPass specifications themselves, it was well known to a POSITA that PayPass recited in Collinge is an EMV-compliant standard.

Aspect / Notes	PayPass-MagStripe Ex[1058]	EMV (MagStripe Mode)
Card Key Derivation	From §III.3.8.1: This section specifies the key derivation method used to generate the <i>ICC Derived Key for CVC3 Generation (KDCVC3)</i>. <i>KDCVC3</i> is a 16-byte DES3 key derived from the <i>Issuer Master Key for CVC3 Generation (IMKVC3)</i> as follows: 1. Concatenate from left to right the PAN (without any 'F' padding) with the PAN sequence number⁷ (if the PAN sequence number is not available, then it is replaced by a '00' byte). If the result X is less than 16 digits long, pad it to the left with hexadecimal zeros in order to obtain an eight-byte number Y in numeric (n) format. If X is at least 16 digits long, then Y consists of the 16 rightmost digits of X in numeric (n) format. 2. Compute the two eight-byte numbers: $ZL := \text{DES3}(\text{IMKVC3})[Y]$ $ZR := \text{DES3}(\text{IMKVC3})[Y \oplus (\text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'})]$ and define : $KDCVC3 := (ZL \parallel ZR)$.	From EMV Book 2 (Ex[1020]), §A1.4.1: 1. Concatenate from left to right the decimal digits of the Application PAN with the PAN Sequence Number (if the PAN Sequence Number is not present, then it is replaced by a '00' byte). If the result X is less than 16 digits long, pad it to the left with hexadecimal zeros in order to obtain an 8-byte number Y in numeric format. If X is at least 16 digits long, then Y consists of the 16 rightmost digits of X in numeric format. 2. Compute the two 8-byte numbers $ZL := \text{DES3}(\text{IMK})[Y]$ and $ZR := \text{DES3}(\text{IMK})[Y \oplus (\text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'})]$ and define $Z := (ZL \parallel ZR)$

Fiske Suppl. Decl., Ex[1050], Annex A

EMV and Associated Payment Standards

- In addition, as demonstrated by Petitioner's expert and corroborated by PayPass specifications themselves, it was well known to a POSITA that PayPass recited in Collinge is an EMV-compliant standard.

Aspect / Notes	PayPass-MagStripe Ex[1058]	EMV (MagStripe Mode)
Cryptogram generation	From §III.3.8.2: - Concatenate the data listed in Table 44 in the order specified to obtain an 8 byte data block (D): Table 44—Track 1 CVC3 Data Objects Data Object Length IVCVC3TRACK1 2 bytes UN 4 bytes ATC 2 bytes - Calculate O as follows: $O := \text{DES3}(KDCVC3)[D]$ - The two least significant bytes of O are the CVC3TRACK1.	From <i>id.</i> §8.1.2 at 170: The 8-byte Application Cryptogram shall be generated using the MAC algorithm specified in Annex A1.2 and ISO/IEC 9797-1 Algorithm 3 with DES, and $s=8$. <i>id.</i> §A1.2: Cryptogram Computation Process the 8-byte blocks $X_1, X_2, \dots, X_k$ with the block cipher in CBC mode using the leftmost MAC Session Key block KSL: $H_i := \text{ALG}(\text{KSL})[X_i \text{ xor } H_{i-1}]$, for $i = 1, 2, \dots, k$ with initial value $H_0 := ('00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00')$. Compute the 8 byte block H_{k+1} in one of the following two ways. □ According to ISO/IEC 9797-1 Algorithm 3: $H_{k+1} := \text{ALG}(\text{KSL})[\text{ALG-1}(\text{KSR})[H_k]]$

Fiske Suppl. Decl., Ex[1050], Annex A

EMV and Associated Payment Standards

- In addition, as demonstrated by Petitioner's expert and corroborated by PayPass specifications themselves, it was well known to a POSITA that PayPass recited in Collinge is an EMV-compliant standard.

Aspect / Notes	PayPass-MChip Ex[1059]	EMV (EMV Mode)												
Transaction Flow The various PayPass M/Chip transaction flows may be determined by the application states in Figure 2.1 and the initiation commands as shown in §2.2.2 A typical flow may involve: • Select(AID) to move to the SELECTED state • Get Processing Options to move to the INITIATED state • Multiple READ RECORD commands within the INITIATED state • A GENERATE AC command to move to either the ONLINE state if an ARQC is returned, or to the SCRIPT state otherwise If no SCRIPT command is received, the card application	From §§2.2.1-2.2.2 Figure 2.1—State Machine of the PayPass – M/Chip 4 Application Table 2.2—Initiation Commands for the PayPass – M/Chip Select 4 Application <table><tr><th>Initiation Commands</th><th>SW1-SW2</th></tr><tr><td>GET CHALLENGE</td><td>'9000'</td></tr><tr><td>GET DATA</td><td>'9000'</td></tr><tr><td>INTERNAL AUTHENTICATE</td><td>'9000'</td></tr><tr><td>VERIFY</td><td>'9000', '6983', '63C4'</td></tr><tr><td>READ RECORD</td><td>'9000'</td></tr></table>	Initiation Commands	SW1-SW2	GET CHALLENGE	'9000'	GET DATA	'9000'	INTERNAL AUTHENTICATE	'9000'	VERIFY	'9000', '6983', '63C4'	READ RECORD	'9000'	From EMV Book C-2 (Ex[1026]) §3.4.3 For an EMV mode transaction, after the GET PROCESSING OPTIONS command, the Kernel continues with the following steps: 1. It determines which form of Offline Data Authentication to perform. 2. It reads the data records of the Card (using READ RECORD commands). If the same transaction involving the same Card is recognized in the Kernel's internal log of torn transactions, then an attempt is made to recover the transaction – see section 3.7. 3. It performs Terminal Risk Management and Terminal Action Analysis, and selects a cardholder verification method for the transaction. 4. It requests an <i>Application Cryptogram</i> from the Card by issuing a GENERATE AC command. If a response is not received from the Card, the Kernel considers the transaction as “torn”, and stores the transaction details in its internal log of torn transactions, before terminating – see section 3.7. 5. It performs Offline Data Authentication as appropriate.
Initiation Commands	SW1-SW2													
GET CHALLENGE	'9000'													
GET DATA	'9000'													
INTERNAL AUTHENTICATE	'9000'													
VERIFY	'9000', '6983', '63C4'													
READ RECORD	'9000'													

EMV and Associated Payment Standards

- By 2012, there were a large number of EMV-compliant cards and terminals worldwide, with a well recognized trend toward migration in the U.S.

globally have focused on chip + PIN enablement. Whatever the format, smart chips are the basis of the technical standard behind more than 1.24 billion payment cards and 15.4 million POS terminals,³ with almost all of those cards and acceptance devices residing outside the United States. Europe, Canada, Latin America and Asia/Pacific are all in various stages of EMV chip migration and usage, leaving the U.S. region—the largest user of payment cards in the world—as the last major hold-out for implementing the otherwise global standard.

 Payment industry experts generally agree that a chip-based standard will come to the U.S., but the predictions of when and in what form vary dramatically. While pundits have been saying the U.S. is far from ready for it, there is a distinct possibility that the change may come sooner rather than later.

First Data White Paper, Ex[1041] at 2.

- Indeed, Patent Owner's expert himself acknowledged that the U.S. had targeted transition to EMV by 2015.

37. Bankrate.com states “Major credit card issuers asked that most U.S. businesses that accept credit cards move toward an EMV-compliant credit card point of sale (POS) system by Oct. 1, 2015 (for fuel retailers, the EMV liability deadline was April 2021).”⁴ This is moving towards EMV compliance by 2015 and a deadline of 2021. Even at that time, a company need not implement all of the EMV standard, but rather a minimal subset⁵.

Easttom Decl., Ex[2009] 37.

EMV and Associated Payment Standards

- Therefore, a POSITA in 2012 working with payment systems would be familiar with EMV as background knowledge.

13. PO's expert argues that EMV had low adoption in the U.S. by 2012. Ex[2009] ¶¶37-43. This is irrelevant. U.S. adoption rate is irrelevant because a POSITA would have been aware of EMV and motivated to build products that comply with it, due to its wide adoption worldwide and a well-known trend to migrate in the U.S, which PO's expert admits. Ex[2009] ¶37. By 2012, U.S. issuers have started making cards EMV compliant so that they would work worldwide with EMV-compliant terminals—there were more than 1.2 billion EMV-compliant cards in circulation worldwide as of 2011, and with a well-known target migration date of 2015 in the U.S. as well. Ex[1041], 2; Ex[2009] ¶37.

Fiske Supp. Decl., Ex[1050] ¶ 13.

DISCLOSURE INCORPORATED BY REFERENCE

03

Prior Art Status of Disclosures in Provisional Applications

- Petitioner does not contend the Collinge Provisional is prior art.
- The Collinge publication (Ex[1011]) is prior art, and its disclosure includes the disclosures from the provisional applications that are incorporated by reference.
- Incorporated disclosures are properly considered
 - *Advanced Display Sys. Inc. v. Kent State University*, 212 F.3d 1272 (Fed. Cir. 2000) (“[m]aterial not explicitly contained in the single, prior art document may still be considered for purposes of anticipation if that material is incorporated by reference”)
 - *Comcast Cable Comms, LLC v. Rovi Guides, Inc.*, IPR2017-01065, Paper 35 (P.T.A.B. Oct. 15, 2018) (“we are persuaded that [the provisional application] can be relied upon for its disclosure, having been properly incorporated by reference...into [the published patent]”).

Petition:

⁵ The provisional applications are incorporated by reference into Collinge, and forms part of the disclosure thereof. Ex[1011], Cover.

Paper 1 at 36

RELATED APPLICATIONS

[0001] This application claims the priority benefit of commonly assigned U.S. Provisional Application No. 61/619,095, filed Apr. 2, 2012, entitled “Method and System for Processing mobile Payments for Devices Without Secure Elements”; U.S. Provisional Application No. 61/639,248, filed Apr. 18, 2012, entitled “Method and System for Processing mobile Payments by Provisioning Credentials to Mobile Devices Without Secure Elements”; U.S. Provisional Application No. 61/735,383, filed Dec. 10, 2012, entitled “Systems and Methods for Processing Mobile Payments by Provisioning Credentials to Mobile Devices Without Secure Elements,” by Mehdi Collinge et al.; and U.S. Provisional Application No. 61/762,098, filed Feb. 7, 2013, entitled “Systems and Methods for Processing Mobile Payments by Provisioning Credentials to Mobile Devices Without Secure Elements,” by Mehdi Collinge et al., which are each herein incorporated by reference in their entirety.

Collinge Non-Provisional, Ex[1011]

Prior Art Status of Disclosures in Provisional Applications

- Collinge's incorporation statement is proper and effective
 - *Harari v. Lee*, 656 F.3d 1331, 1335 (Fed. Cir. 2011) (holding incorporation statement that read "The disclosures of the two applications are hereby incorporate[d] by reference" was "broad and unequivocal" and thereby sufficient to incorporate the entirety of the two patent applications into the document); *Paice LLC v. Ford Motor Co.*, 881 F.3d 894, 907 (Fed. Cir. 2018); *Comcast Cable Commc'ns, LLC v. Rovi Guides, Inc.*, No. IPR2017-01065, Paper 35 (P.T.A.B. Oct. 15, 2018); *Ex Parte Gary K. Michelson*, No. APPEAL 2012-004146, 2015 WL 5118373, at *3-4 (P.T.A.B. Aug. 27, 2015).
- Incorporating allegedly "nonpublic" information is effective as long as such information is public on publication/issuance.
 - *In re Hawkins*, 486 F.2d 569, 574 (C.C.P.A. 1973) ("the function of section 112 ...is only violated if the disclosure is not complete at the time it is made public, i.e., at the issue date") (emphasis in original); *In re Lundak*, 773 F.2d 1216, 1222 (Fed. Cir. 1985) (disclosure requirements satisfied by "public access after grant" of the patent).
- By Office rule, the Collinge provisionals were public at least as of Collinge's publication
 - 37 C.F.R. § 1.14(a)(1)(iv)-(v)
 - Ex 1009, Ex 1010

GROUNDS 1/2 - COLLINGE 04

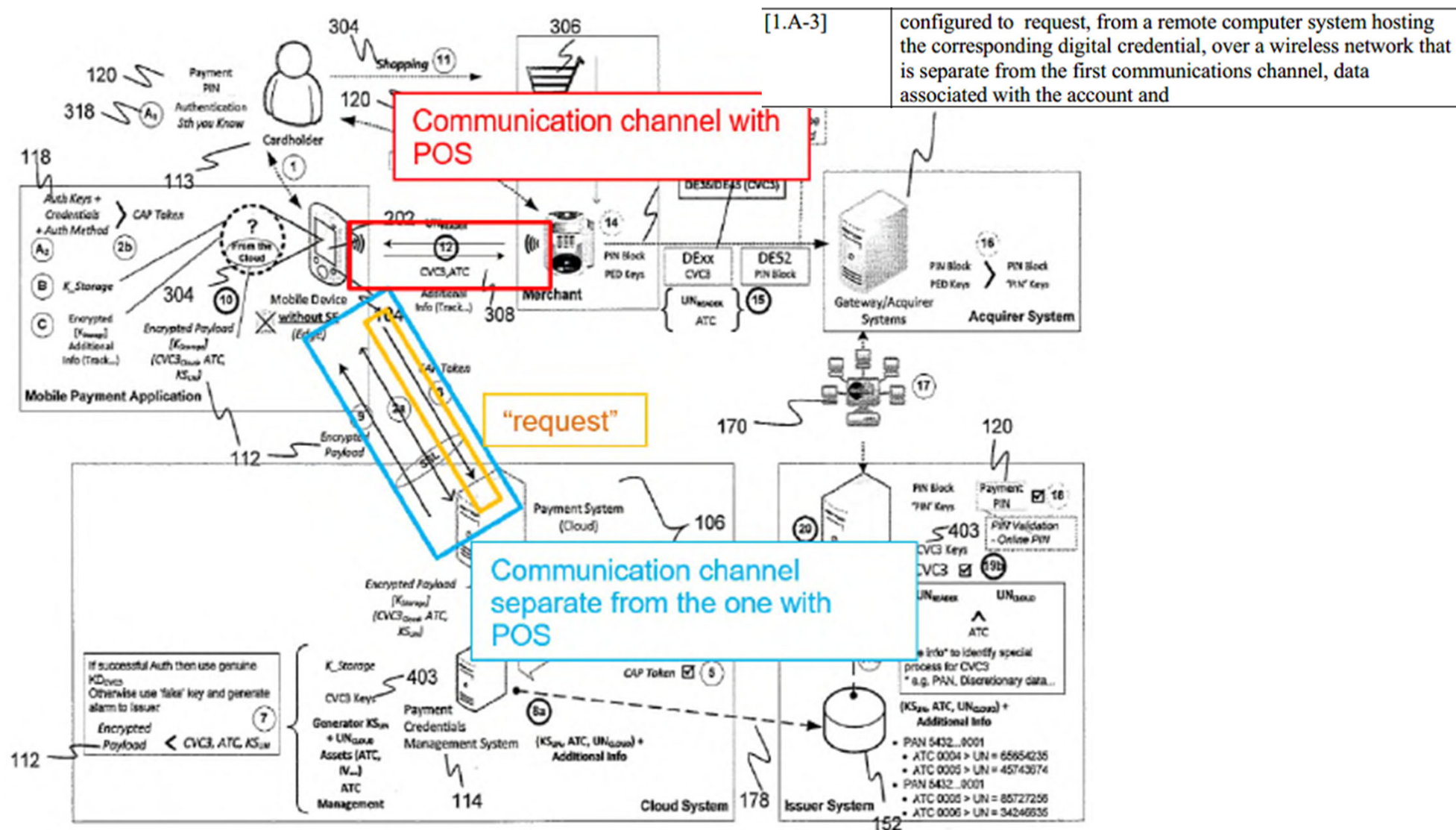
Claim 1

[1.PRE]	A method for secure application-based participation in a payment card transaction authorization process by a mobile device, the method comprising:
[1.A-1]	at a mobile device, executing an application in an operating system of the mobile device,
[1.A-2]	the application interrogable by an electronic reader, over a first communications channel, for digital credential data corresponding to an account having a corresponding digital credential and
[1.A-3]	configured to request, from a remote computer system hosting the corresponding digital credential, over a wireless network that is separate from the first communications channel, data associated with the account and
[1.A-4]	generate cryptograms requested during interrogations over the first communications channel using the data associated with the account received from the remote computer system, wherein the application does not access a permanent cryptographic key issued for the digital credential during the interrogations
[1.B]	requesting by the application, from the remote computer system, over the wireless network, a first set of data associated with the account;
[1.C]	receiving by the application, from the remote computer system, over the wireless network, the first set of data associated with the account, the first set of data associated with the account usable by the application to formulate an application protocol data unit response, the first set of data comprising a first non-permanent cryptographic key associated with the account;
[1.D]	locally storing the first non-permanent cryptographic key at the mobile device as a local cryptographic key associated with the account;
[1.E]	participating by the application in an interrogation between a point-of-sale (POS) terminal and the mobile device comprising:
[1.E-1]	receiving at least one POS command communication sent by the POS terminal over the first communications channel, wherein the at least one POS command communication comprises a request

	for digital credential data to authorize a transaction against the account, the request for digital credential data including a cryptogram request;
[1.E-2]	generating a response cryptogram based on a set of inputs and the local cryptographic key associated with the account, wherein the response cryptogram does not include the local cryptographic key associated with the account used to generate the response cryptogram; and
[1.E-3]	responding to the at least one POS command communication, wherein responding to the at least one POS command communication comprises sending at least one device response communication from the mobile device to an electronic reader through the communication channel, the at least one device response communication comprising at least one response application data protocol unit containing the response cryptogram and an account identifier for the account;
[1.F]	subsequent to the interrogation, requesting by the application, from the remote computer system, over the wireless network, a second set of data associated with the account;
[1.G]	receiving by the application, from the remote computer system, over the wireless network, the second set of data associated with the account, the second set of data comprising a second non-permanent cryptographic key associated with the account; and
[1.H]	storing the received second non-permanent cryptographic key as the local cryptographic key associated with the account to change the local cryptographic key associated with the account between at least two interrogations.

[1.A-3] "request"

- Collinge discloses credential communications initiated by the transmission of a CAP token from the user's phone/device:



Ex[1009] Fig. 7 (annotated)

[1.A-3] “request”

- Collinge explicitly characterizes the provisioning of the payment credential as “a pull request”:

19. The method of claim 18, wherein the provisioning of the payment credentials is a pull request from the mobile device.

Ex[1009], 42

- Provisioning of the credential is initiated by transmission of a CAP token for authentication:

18. A method for generating and provisioning payment transaction data to a mobile device from a Cloud-based transaction data generation system (Cloud), the method comprising:

authenticating the mobile device to the Cloud based upon authentication credentials previously provisioned to the mobile device;

Ex[1009], 41

[1.A-3] “request”

- Collinge discloses a deterministic sequence initiated by the transmission of a CAP token:

3	Mobile Payment Application <u>sends a CAP Token to the Cloud</u>
4	The Payment System (in the Cloud) validates the CAP Token using a CAP Token Validation Service (CTVS).
5	The CTVS validates the CAP Token.
6	The result of the CAP Token validation is sent to the Payment Credentials Management System.
7	Upon successful authentication, a genuine KD_{CVC3} is used and $(KS_{UN}, Cloud_CVC3_{TRACK1/2}, ATC)$ is returned.
8b	$(KS_{UN}, Cloud_CVC3_{TRACK1/2}, ATC)$ is returned to the front-end of the Cloud system for delivery to the Mobile Payment Application.
9	<u>$(KS_{UN}, Cloud_CVC3_{TRACK1/2}, ATC)$ is returned to the Mobile Payment Application.</u> This can encompass additional payment assets.
12	Standard <i>PayPass</i> Magstripe payment experience using a Mobile Device. The Mobile Payment Application use a specific process to generate the CVC3 using $(KS_{UN}, Cloud_CVC3_{TRACK1/2}, ATC)$ and UN_{READER}

Ex[1009], 21, 22
Paper 1 at 38

[1.A-3] “request”

- Collinge discloses a deterministic sequence that was initiated by the transmission of a CAP token:

[0082] In stage 3, the mobile payment application 106 may transmit the generated CAP token to the remote-SE system 100. At stage 4, the CAP token may be forward to a CAP token validation service (CTVS) 702. At stage 5, the CTVS 702 may validate the CAP token using methods apparent to persons having skill in the relevant art. At stage 6, the results of the validation may be sent to the payment credentials management system 704. In some embodiments, the payment credentials management system 704 may be operated by or on behalf of the issuer 108.

[0083] At stage 7, the payment credentials management system 704 may generate an encrypted payload. As part of the encrypted payload, the payment credentials management system 704 may generate a session key unpredictable number (KS_{UN}), a cloud unpredictable number (UN_{CLOUD}), and may identify and/or store a plurality of dynamic card validation code (CVC3) keys and the $K_{STORAGE}$. Methods and systems for the generation of unpredictable numbers will be apparent to persons having skilled in the relevant art. The encrypted payload may include at least one CVC3 key, the KS_{UN} , and the application transaction counter and may be generated using a derived CVC3 key (KD_{CVC3}). In one embodiment, the KD_{CVC3} used may be fake if the CAP token validation is unsuccessful. The payload may be encrypted using the $K_{STORAGE}$.

[0084] At stage 8a, the payment credentials management system 704 may perform a synchronization process with the issuer 108. The synchronization process may include defining rules for the validity of the values generated by the mobile payment application 106 for conducting a payment transaction. In one embodiment, the process may include transmitting at least the KS_{UN} , application transaction counter, and UN_{CLOUD} to the issuer 108 for storage in the issuer database 310.

[0085] At stage 8b, the encrypted payload may be transmitted to the cloud system 302 for transmitting to the mobile payment application 106 in stage 9. At stage 10, the encrypted payload may be decrypted using the $K_{STORAGE}$ and stored in the storage 304. At stage 11, the user 102 may shop at the merchant 306 and select goods or services for purchase. As

Ex[1011], [0082]-[0085]

[1.A-3] “request”

- Petitioner also showed that a “request” was an obvious way to obtain the needed information among a finite number of predictable alternatives

Petition:

To the extent Collinge does not explicitly disclose that the application would “request” a first set of data from the remote computer system, in light of the fact that such data is retrieved from a remote system, a POSA would have found it obvious to configure the mobile application to request it as that was one of a finite number of predictable ways to retrieve data. Ex[1011], [0082], [0083]; Ex[1006] ¶ 88. Moreover, it would be a waste of bandwidth and resources for the issuer to push information regardless of the status of the user device. Ex[1006] ¶ 88.

Paper 1 at 40-41

[1.C] “associated with”

response, the first set of data comprising a first non-permanent cryptographic key associated with the account;

- PO’s argument is premised on the assumption that a key is only “associated with” an account if the account number can be recovered from a key

in its generation. Pet. 43. This is incorrect for two reasons. First, as Petitioner’s expert acknowledged, information sent as part of an account-specific request that cannot be used to determine which account it relates to would not be associated with the account. EX2035 at 84:19-86:11. And, as Dr. Fiske also recognized, a POSITA would not be able to determine any of the account information, including the PAN, using KS_{UN}. EX2035 at 81:6-13; EX2009 ¶112-113. Second, Petitioner’s contention that KS_{UN}

Paper 22 at 18

- PO offers no support for this narrow construction, and abandons it in Sur-Reply

The POR does not contend that KS_{UN} in Collinge is not *associated with the account* merely because it cannot be used to recover the account number. POR 18-19. Instead, the POR explains that KS_{UN} being

Paper 28 at 5

Paper 26 at 10

[1.C] “associated with”

response, the first set of data comprising a first non-permanent cryptographic key associated with the account;

- Collinge’s Non-Permanent Key (KS_{UN}) is “associated with the account” in three ways:

- 1) It is generated in response to an account-specific request, and only provided if validation succeeds
- 2) It is generated based on an account-specific counter

KS_{UN}

Generate KS_{UN} using:

$U_L := DES3(KD_{UN})[(ATC \parallel 'F0' \parallel '00' \parallel UN_{CLOUD})]$

$U_R := DES3(KD_{UN})[(ATC \parallel '0F' \parallel '00' \parallel UN_{CLOUD})]$

$KS_{UN} := (U_L \parallel U_R)$

Ex1009 at 28

- 3) It is generated based on an account specific key (KD_{UN}) that is computed based on the account number (PAN)

[0069] Concatenate from left to right the PAN (without any ‘F’ padding) with the PSN (if the PAN sequence number is not available, then it is replaced by a ‘00’ byte). .

[0070] If the result X is less than 16 digits long, pad it to the left with hexadecimal zeros in order to obtain an eight-byte number Y in numeric (n) format. .

KD_{UN}

Reuse Value Y as defined above

Generate KD_{UN} using:

$Z_L := DES3(IMK_{UN})[Y]$

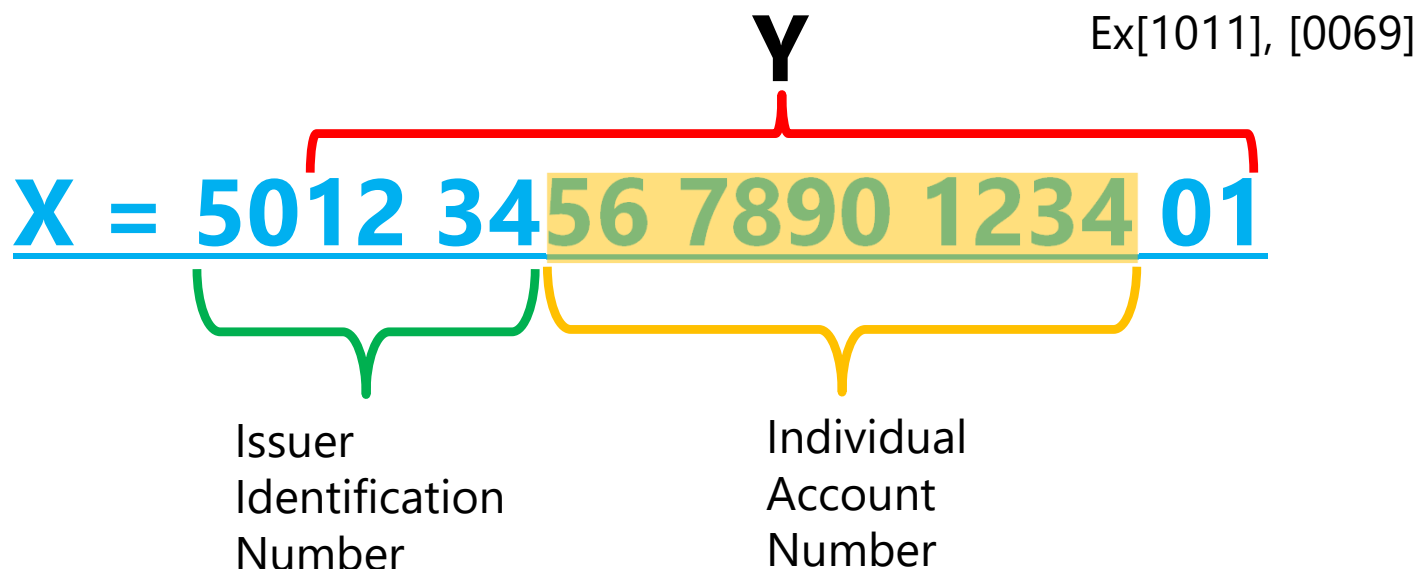
$Z_R := DES3(IMK_{UN})[Y \oplus ('FF' \parallel 'FF' \parallel 'FF' \parallel 'FF' \parallel 'FF' \parallel 'FF' \parallel 'FF' \parallel 'FF')]$

$KD_{UN} := (Z_L \parallel Z_R)$

[1.C] “associated with”

- **Generation of K_{SUN} requires generation of value “Y”**
- Y uniquely identifies an account at a given issuer because it contains the entirety of the individual account number:

[0071] If X is at least 16 digits long, then Y consists of the 16 rightmost digits of X in numeric (n) format.



7.4.1 Primary Account Number

The Primary Account Number (PAN) comprises a six-digit Issuer Identification Number (IIN), a variable length (maximum 12 digits) individual account number and a check digit, and is as defined in ISO/IEC 7812-1. ISO/IEC 7812-2 describes the application and registration procedures for IINs.

Ex[1065] §7.4.1

[1.C] “associated with”

response, the first set of data comprising a first non-permanent cryptographic key associated with the account;

KD_{UN}

Reuse Value Y as defined above

Generate KD_{UN} using:

$Z_L := \text{DES3}(\text{IMK}_{\text{UN}})[Y]$

$Z_R := \text{DES3}(\text{IMK}_{\text{UN}})[Y \oplus (\text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'})]$

$\text{KD}_{\text{UN}} := (Z_L \parallel Z_R)$

Ex[1009], 27

KS_{UN}

Generate KS_{UN} using:

$U_L := \text{DES3}(\text{KD}_{\text{UN}})[(\text{ATC} \parallel \text{'F0'} \parallel \text{'00'} \parallel \text{UN}_{\text{CLOUD}})]$

$U_R := \text{DES3}(\text{KD}_{\text{UN}})[(\text{ATC} \parallel \text{'0F'} \parallel \text{'00'} \parallel \text{UN}_{\text{CLOUD}})]$

$\text{KS}_{\text{UN}} := (U_L \parallel U_R)$

Ex[1009], 28

- **Generation of KD_{UN} using IMK_{UN} and Y**
- Because IMK_{UN} (IMK = Issuer Master Key (see Ex[1009], 31]) is unique to an issuer, the resulting KD_{UN} is uniquely associated with the account because a different issuer would have a different IMK_{UN} and a different account at that issuer would have a different Y.
- **Generation of KS_{UN} using KD_{UN} and ATC (transaction counter)**
- Because KD_{UN} is uniquely associated with the account, the resulting KS_{UN} is also uniquely associated with the account because a different account would have a KD_{UN}

[1.C/1E-3] APDU Response

[1.C]

receiving by the application, from the remote computer system, over the wireless network, the first set of data associated with the account, the first set of data associated with the account usable by the application to formulate an application protocol data unit response, the first set of data comprising a first non-permanent cryptographic key associated with the account;

- Limitation 1.C expressly has to do with receiving a “first set of data”
- The claim limitations related to that data being used to formulate and “application protocol data unit” are separately claimed and analyzed in the petition with element 1.E-3.

[1.E-3]

responding to the at least one POS command communication, wherein responding to the at least one POS command communication comprises sending at least one device response communication from the mobile device to an electronic reader through the communication channel, the at least one device response communication comprising at least one response application data protocol unit containing the response cryptogram and an account identifier for the account;

- PO merely ignores the Petition’s explanation as to application protocol data units

[1.C/1E-3] APDU Response

- The Petition explains how Collinge's "first set of data" is used to generate a cryptogram and pass it to the POS via a "standardized EMV" APDU

Petition:

It is well known in the art that, during the interrogation, commands issued by the terminal and responses from the "Card" are communicated in command- and response-application protocol data units (C-APDUs and R-APDUs), respectively and it would be obvious to implement Collinge in this way to ensure interoperability with EMV-standardized devices. Ex[1037] at 46-48; Ex[1021] § 6.5; Ex[1006] ¶ 95.

Paper 1 at 43-44

(Stage 11-20), [0063]. This is also consistent with standardized EMV transactions (with which a POSA reading Collinge will understand its disclosure as compatible, Ex[1006] ¶ 73) where the mobile device ("Card") generates a cryptogram in response to a COMPUTE CRYPTOGRAPHIC CHECKSUM, GENERATE AC, or GPO command. Ex[1026] §§ 3.4.2, 3.4.3, 5.2, 5.3; Ex[1027] § 2.4.1. The

A POSA would also have found it obvious to implement Collinge in the manner described above because these are standardized operations in secure payment transactions and are therefore applying a known technique (using EMV standard POS-card messaging) to yield predictable results (permitting EMV terminal to assemble an authorization request/payment message for transmission), and would ensure the system is compatible with existing POS infrastructure. Ex[1006] ¶ 101.

Paper 1 at 45-46

[1.C/1E-3] APDU Response

- The Petition explains how Collinge’s “first set of data” is used to generate a cryptogram and pass it to the POS via a “standardized EMV” APDU

Petition:

As explained in [1.E] and [1.E-1], the card (or device) being interrogated would respond to a command APDU (C-APDU) from the POS terminal with a response APDU. As explained in [1.A-2], the application on the mobile device interacts with the POS terminal, over a first communications channel (e.g., NFC). Ex[1011], [0002]-[0003], [0044].

As explained in [1.A-2], [1.E-1], and [1.E-2], Collinge discloses that the mobile device sends payment credential data, which a POSA would understand to include generated payment cryptogram and other “common payment credential information” such as account information (e.g., PAN)⁶ to the POS terminal in response to a POS communication. *See supra* §§ X.A.1.c, X.A.1.i(1), X.A.1.i(2); *see also* Ex[1011], [0044], [0055], [0070], [0077]; Ex[1009], [0061] (Table 1, Step 15), [0064]. A POSA would understand that this can be implemented in the manner directed by either Construction 1 or Construction 2. *Supra* § VIII; Ex[1006] ¶ 107.

Alternatively, under Construction 1, where the Term could mean one or more APDUs collectively containing the response cryptogram and an account identifier for the account, this is satisfied by Collinge’s reference to the PayPass card specification in the provisional application, incorporated by reference. Ex[1009], [0003], [0029]-[0032], [0040]. It is understood in the art that, during interrogation for a contactless transaction including a Mastercard Paypass transaction, the “Card” (user device) would provide, in R-APDUs, account information such as a Primary Account Number (PAN) in response to a READ RECORD command, and a response cryptogram (CVC3 or Application Cryptogram) in response to a COMPUTE CRYPTOGRAPHIC CHECKSUM or GENERATE AC command APDU. Ex[1048], [0017]-[0018]; Ex[1026], §§ 3.4.2, 3.4.3, 3.8.1–3.8.3, 5.2, 5.3; Ex[1006] ¶ 108.

[1.C] APDU Response

- Because PayPass is EMV-compliant and a POSITA would be familiar with EMV, a POSITA would know that the response from a card is communicated with an APDU, which is simply the standard message format.

3.2.3 Command and response format

In accordance with the OSI 7-layer model, the information transaction exchanged between the card and the terminal can be divided into three protocol sections:

- Application protocols for command and response data (layer 7). A step in an application protocol consists of sending a command application protocol data unit (C-APDU) from the terminal application to the card application. The latter processes it and sends back the response application protocol data unit (R-APDU) to the terminal application. A schematized picture of a C-APDU/R-APDU pair is given in Figure 3.5.

Radu, Ex[1037], 46-47

6.1 Command APDU Format

The command Application Protocol Data Unit (APDU) consists of a mandatory header of four bytes followed by a conditional body of variable length, as shown in Figure 1:

CLA	INS	P1	P2	Lc	Data	Le
← Mandatory Header ² →				← Conditional Body →		

Figure 1: Command APDU Structure

6.2 Response APDU Format

The response APDU format consists of a conditional body of variable length followed by a mandatory trailer of two bytes, as shown in Figure 2:

Data	SW1	SW2
← Body →	← Trailer →	

Figure 2: Response APDU Structure

The data field of a response APDU is an object structured as defined in Annex B. The detailed coding of the objects is provided with the commands described in subsequent sub-clauses.

EMV Book 3, Ex[1021] §§ 6.1-6.2 at 57-58

Paper 1 at 38, 43-44, Paper 26 at 11

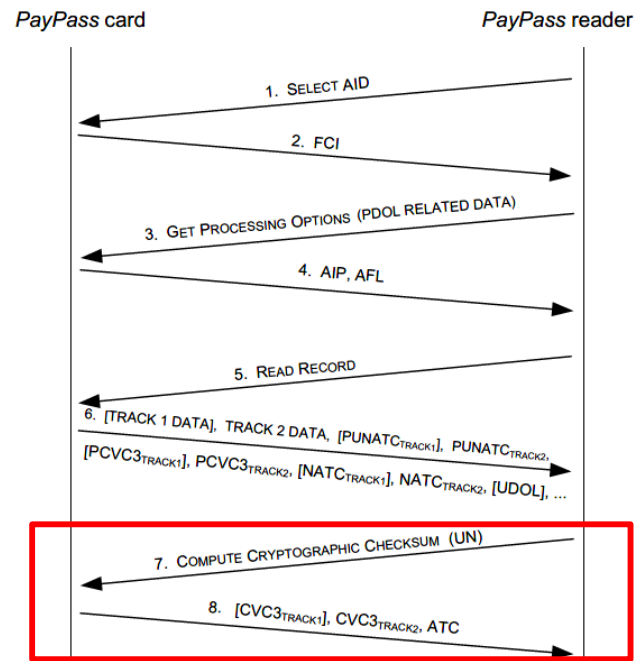
[1.C] APDU Response

- Collinge states that its technique is applicable to Paypass, and a Skilled Artisan would know any command and response (including the cryptogram) will be sent in an APDU

3.1 Overview of Command – Response APDU Flow

Figure 2 illustrates the Command – Response APDU flow of a *PayPass* – Mag Stripe transaction including the SELECT AID command.

Figure 2—Command – Response APDU Flow



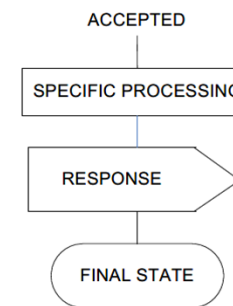
PayPass Spec, Ex[1058], §III.3.1 at 23

2.4 C-APDU Processing

2.4.1 Overview

Figure 2.2 illustrates the actions taken by the *PayPass* – M/Chip 4 application when a C-APDU is processed.

Figure 2.2—Processing a C-APDU



A C-APDU is processed if the C-APDU Recognition and Acceptance procedures have not resulted in the rejection of the C-APDU (see Section 2.3.3).

The processing that is specific to the C-APDU is specified in Section 2.4.3 for the COMPUTE CRYPTOGRAPHIC CHECKSUM command and in [M/CHIP4] for all other commands.

The R-APDU resulting from the processing of a C-APDU and the destination state of the application when the C-APDU is processed are specified in the section dedicated to the C-APDU.

PayPass Spec, Ex[1059], §2.4.1 at 25

Paper 1 at 38, 43-44, Paper 26 at 11

[1.E-2] Cryptogram (CVC3) Calculation

- Collinge explicitly recites a formula for calculation of cryptogram (CVC3) using the single-use key, KS_{UN}

2. Calculate T as follows:

$$T := DES3(KS_{UN})[M]$$

The two least significant bytes of T are the $CVC3_{TRACK1/2}$

Ex[1009], 29-30

- A POSITA understands that a cryptogram such as CVC3 or Application Cryptogram is a Message Authentication Code, generated with 3DES algorithm.

8.1 Application Cryptogram Generation

8.1.1 Data Selection

An Application Cryptogram consists of a Message Authentication Code (MAC) generated over data:

- referenced in the ICC's DOLs and transmitted from the terminal to the ICC in the GENERATE AC or other command, and
- accessed internally by the ICC.

EMV Book 2, Ex[1020], §8.1.1 at 100

A1.2 Message Authentication Code

The computation of an s-byte MAC ($4 \leq s \leq 8$) is according to ISO/IEC 9797-1 using a 64-bit block cipher ALG in CBC mode. More precisely, the computation of a MAC S over a message MSG consisting of an arbitrary number of bytes with a MAC Session Key K_S takes place in the following steps.

B1.1 Data Encryption Standard (DES)

The double-length key triple DES encipherment algorithm (see ISO/IEC 18033-3) is the approved cryptographic algorithm to be used in the encipherment and MAC mechanisms specified in Annex A1. The algorithm is based on the (single) DES algorithm standardised in ISO 16609.

EMV Book 2, Ex[1020], §8.1.1 at 141, 149

Paper 1 at 47, Paper 26 at 12

[1.E-2] Cryptogram (CVC3) Calculation

- Collinge improved upon PayPass-MagStripe by using a one-time cryptographic key KS_{UN} , rather than static key (e.g. $KDCVC3$), for generation of the application cryptogram/MAC (CVC3), while maintaining compatibility with the existing PayPass-MagStripe terminal infrastructure/payment messaging.
- Generation of $KDCVC3$ vs. KS_{UN} : both are 16-byte keys (Ex[1050] ¶35)

$KDCVC3$ is a 16-byte DES3 key derived from the *Issuer Master Key for CVC3 Generation* (IMK_{CVC3}) as follows:

1. Concatenate from left to right the PAN (without any 'F' padding) with the PAN sequence number⁷ (if the PAN sequence number is not available, then it is replaced by a '00' byte). If the result X is less than 16 digits long, pad it to the left with hexadecimal zeros in order to obtain an eight-byte number Y in numeric (n) format. If X is at least 16 digits long, then Y consists of the 16 rightmost digits of X in numeric (n) format.

2. Compute the two eight-byte numbers:

$Z_L := DES3(IMK_{CVC3})[Y]$

$Z_R := DES3(IMK_{CVC3})[Y \oplus ('FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF')]$

and define :

$KDCVC3 := (Z_L || Z_R)$.

PayPass Spec., Ex[1058] §III.3.8.1 at 108

KD_{UN}

Reuse Value Y as defined above

Generate KD_{UN} using:

$Z_L := DES3(IMK_{UN})[Y]$

$Z_R := DES3(IMK_{UN})[Y \oplus ('FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF')]$

$KD_{UN} := (Z_L || Z_R)$

Ex[1009], 27

KS_{UN}

Generate KS_{UN} using:

$U_L := DES3(KD_{UN})[(ATC || 'F0' || '00' || UN_{CLOUD})]$

$U_R := DES3(KD_{UN})[(ATC || '0F' || '00' || UN_{CLOUD})]$

$KS_{UN} := (U_L || U_R)$

Ex[1009], 28

Paper 1 at 47, Paper 26 at 12-13

[1.E-2] Cryptogram (CVC3) Calculation

- KS_{UN} can be used in place of a KD_{CVC3} to generate the CVC3 in exactly the same way, thereby maintaining compatibility (Ex[1050] ¶35)

3.8.2 Dynamic CVC3 Generation

The $CVC3_{TRACK1}$ is generated using DES3 encipherment as follows:

1. Concatenate the data listed in Table 44 in the order specified to obtain an 8 byte data block (D):

Table 44—Track 1 CVC3 Data Objects

Data Object	Length
$IVCVC3_{TRACK1}$	2 bytes
UN	4 bytes
ATC	2 bytes

2. Calculate O as follows:

$$O := DES3(KD_{CVC3})[D]$$

3. The two least significant bytes of O are the $CVC3_{TRACK1}$.

1. Concatenate the following data to obtain an 8 byte data block (M):

- $Cloud_CVC3_{TRACK1/2}$ (2 bytes)
- UN_{READER} (4 bytes)
- ATC (2 bytes)

2. Calculate T as follows:

$$T := DES3(KS_{UN})[M]$$

The two least significant bytes of T are the $CVC3_{TRACK1/2}$

Ex[1009] 29-30

PayPass Spec., Ex[1058] §III.3.8.2 at 109

Paper 26 at 12-13

[1.E-2] Cryptogram (CVC3) Calculation

- It is also common knowledge that MACs including CVC3s are calculated using triple DES algorithm, and the DES algorithm would not include the encryption key in its encrypted text. (Ex[1050] ¶36)

Message authentication is a technique which produces cryptographic check digits which are appended to the message. These digits are analogous to a parity check or cyclic redundancy check, except that in this case the check digits are cryptographically generated, using DES and a secret key. These digits, called the message authentication code or MAC, are generated by the originator, appended to the transmitted message, and then checked by the recipient, who also holds the same secret key used in the generation process. Should anyone attempt to modify the message between the time the MAC is generated and the time it is checked, he would be detected. Not knowing the secret key, he would be unable to generate the correct MAC for his modified message. Similarly, no one can successfully introduce a spurious message because he could not generate the proper MAC for this message.

Example: Let M be the plain text message M=0123456789ABCDEF, where, M is in hexadecimal (base 16) format. Rewriting M in binary format, we get the 64-bit block of text:

Example: Let K be the hexadecimal key K=133457799BBCDFF1. This gives us as the binary key (setting 1 = 0001, 3 = 0011, etc. and grouping together every eight bits, of which the last one in each group will be unused):

Meyer, Ex[1034], 17

Which in hexadecimal format is: 85E813540F0AB405.

This is the encrypted form of M = 0123456789ABCDEF: namely C = 85E813540F0AB405.

Does not contain
the key (K)

Radah, Ex[1039] 29-30

Paper 1 at 47, Paper 26 at 12-13, n.11

[1.E-3] R-APDU w/ Cryptogram and Account ID

- CVC3 and Application Cryptogram (AC) are well known in the art to be names of response cryptograms generated by the payment application for authorization of a transaction in an EMV-compliant transaction.

The processing is similar to that performed in the case of magnetic stripe cards, with the observation that the payment message sent to the acquirer contains the supplementary information gathered by the terminal from the card during the processing explained in Figure 6.1. After receiving this information, the issuer can better assess whether the card is genuine or not, since the Application Cryptogram produced by the card is “fresh” for every new transaction. The authorization response that is sent back by the IH

6.9.1 Authorization request and response with chip data

The terminal generates an authorization request message 1100. The data fields included in this message are those explained in Section 2.9 for the processing of transactions using magnetic stripe cards. In addition, the authorization request message includes the field ICC system-related data, corresponding to bit 55 in the bitmap (see Section 2.8.1) of the data elements present in the 1100 message, according to ISO 8583:1993. This field contains all the data objects from the point of service, which allows the IH to verify the correctness of the ARQC produced by the card. Among these data objects one can mention:

- Application Cryptogram (ARQC);

The skilled artisan will appreciate that the CVC3 (card verification code) is a cryptographic checksum also known as a MAC. Typically, with a normal magnetic stripe transaction, it is a static three digits. In EMV, dynamic cryptographic checksums are employed due to their greater

“Mechanism to allow the use of disposable cards on a system designed to accept cards conforming to the standards of the global payments industry”, Ex[1052] 14:63-65

Radu, Ex[1037], 54, 57

- In particular, Kernel behavior supporting a Mastercard brand or cards are defined in EMV Contactless Book C-2

This document defines the behaviour of the Kernel used in combination with cards supporting a MasterCard brand or cards having a Kernel Identifier indicating Kernel 2, as defined in [EMV Book B].

EMV Book C-2, Ex[1026], 23

Paper 26 at 14

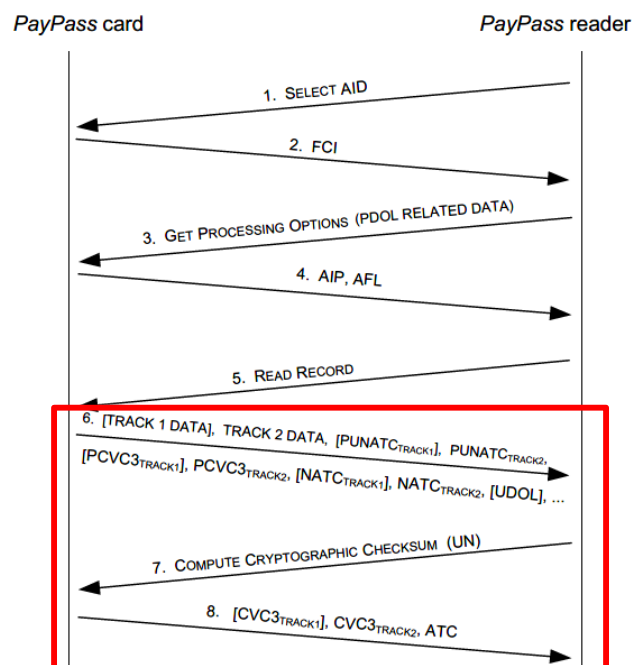
[1.E-3] R-APDU w/ Cryptogram and Account ID

- A POSITA reviewing Collinge would be familiar with the PayPass Specification, and know that PayPass is Mastercard's implementation of contactless EMV-compliant transactions, and/or find it obvious to implement Collinge to be EMV-compliant

3.1 Overview of Command – Response APDU Flow

Figure 2 illustrates the Command – Response APDU flow of a *PayPass* – Mag Stripe transaction including the SELECT AID command.

Figure 2—Command – Response APDU Flow



PayPass Spec, Ex[1058], §III.3.1 at 23

3.4.2 Mag-Stripe Mode

For a mag-stripe mode transaction, after the GET PROCESSING OPTIONS command, the Kernel continues with the following steps:

1. It reads the data records from the Card, containing *Track 1 Data* and *Track 2 Data*, together with instruction on how to populate the discretionary data.
2. It issues the COMPUTE CRYPTOGRAPHIC CHECKSUM command, including *Unpredictable Number (Numeric)* to the Card, requesting the Card to return a CVC3 cryptogram, calculated over *Unpredictable Number (Numeric)*.
3. It populates the *Track 2 Data* with the *Unpredictable Number (Numeric)*, the *Application Transaction Counter*, and *CVC3 (Track2)*.

3.8.2 Mobile Mag-Stripe Mode Transactions

For the support of mobile mag-stripe mode transactions, the Kernel has two mobile specific data objects:

1. The *Mobile Support Indicator*, indicating that the Kernel supports mobile and that a particular transaction requires CVM
2. A *Reader Contactless Transaction Limit (On-device CVM)* for phones (as opposed to a *Reader Contactless Transaction Limit (No On-device CVM)* for cards)

5.2 COMPUTE CRYPTOGRAPHIC CHECKSUM

5.2.1 Definition and Scope

The COMPUTE CRYPTOGRAPHIC CHECKSUM command initiates the computation of the dynamic CVC3 on the Card.

5.2.2 Command Message

The COMPUTE CRYPTOGRAPHIC CHECKSUM command message is coded according to Table 5.3.

Paper 26 at 14-15

EMV Book C-2, Ex[1026], §§3.4.2, 3.8.2, 5.2

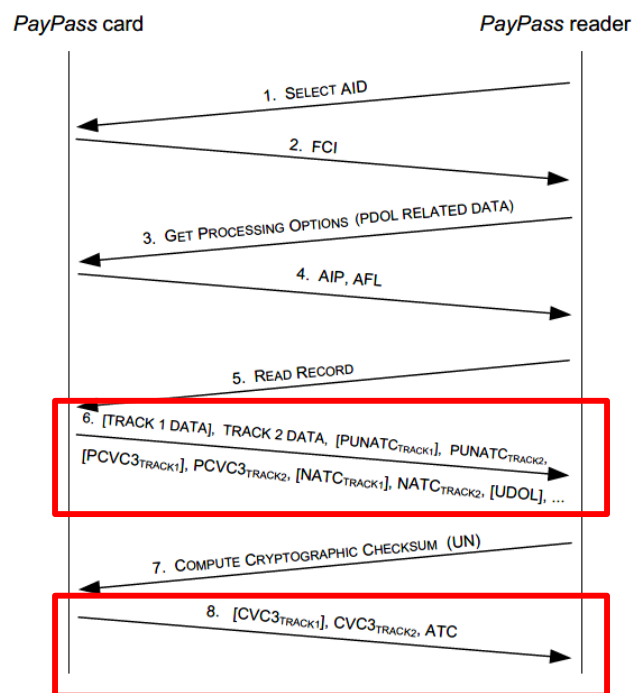
[1.E-3] R-APDU w/ Cryptogram and Account ID

- In a Mastercard transaction, the cryptogram and account number are returned in different R-APDUs, satisfying the construction where “at least one response [APDU]” i.e. more than one ADPU “contain[] the response cryptogram and an account identifier for the account.”

3.1 Overview of Command – Response APDU Flow

Figure 2 illustrates the Command – Response APDU flow of a *PayPass* – Mag Stripe transaction including the SELECT AID command.

Figure 2—Command – Response APDU Flow



PayPass Spec, Ex[1058], §III.3.1 at 23

3.4.2 Mag-Stripe Mode

For a mag-stripe mode transaction, after the GET PROCESSING OPTIONS command, the Kernel continues with the following steps:

- It reads the data records from the Card, containing *Track 1 Data* and *Track 2 Data*, together with instruction on how to populate the discretionary data.
- It issues the COMPUTE CRYPTOGRAPHIC CHECKSUM command, including *Unpredictable Number (Numeric)* to the Card, requesting the Card to return a CVC3 cryptogram, calculated over *Unpredictable Number (Numeric)*.
- It populates the *Track 2 Data* with the *Unpredictable Number (Numeric)*, the *Application Transaction Counter*, and *CVC3 (Track2)*.

3.8.2 Mobile Mag-Stripe Mode Transactions

For the support of mobile mag-stripe mode transactions, the Kernel has two mobile specific data objects:

- The *Mobile Support Indicator*, indicating that the Kernel supports mobile and that a particular transaction requires CVM
- A *Reader Contactless Transaction Limit (On-device CVM)* for phones (as opposed to a *Reader Contactless Transaction Limit (No On-device CVM)* for cards)

5.2 COMPUTE CRYPTOGRAPHIC CHECKSUM

5.2.1 Definition and Scope

The COMPUTE CRYPTOGRAPHIC CHECKSUM command initiates the computation of the dynamic CVC3 on the Card.

5.2.2 Command Message

The COMPUTE CRYPTOGRAPHIC CHECKSUM command message is coded according to Table 5.3.

Paper 1 at 49-50, Paper 26 at 14-15

EMV Book C-2, Ex[1026], §§3.4.2, 3.8.2, 5.2

[1.E-3] R-APDU w/ Cryptogram and Account ID

- Alternatively, in a VISA EMV transaction, the “card” may return an AC in the same APDU message with the Track 2 Equivalent Data with the PAN, in response to the GPO command. Ex[1027] §§2.4.1, 5.2.2.3, 5.4.3.2, Table E-6.

In the GPO response, the kernel is expected to receive data elements from the card that are appropriate to the conditions indicated in the Terminal Transaction Qualifiers:

- For EMV mode – a cryptogram with supporting/additional data, Track 2 Equivalent Data, and for offline approved transactions, a signature and pointer to supporting/additional data (AFL). Signatures that would cause the response data to exceed its data size limit are not included, but are instead provided in a record which is indicated as a record entry in the AFL.

EMV Book C-3, Ex[1027] §2.4.1 at 24

Table E-6: Data Content for Track 2 Equivalent Data

Tag	Value	Format	Length	Presence
'57'	Track 2 Equivalent Data		var. up to 19	M
	Application Primary Account Number (PAN)	cn	var. up to cn 19	M
	field separator = 'D'		1	M
	Expiry Date	n	4	M
	Service Code	n	2	M

EMV Book C-3, Ex[1027] Table E-6 at 159 (Track 2 data includes PAN)

If the reader is EMV mode-enabled
and the reader is not mag-stripe mode-enabled,
then the kernel shall continue processing and for all subsequent kernel processing with different behaviour based on path, the kernel shall proceed with EMV Mode Path Processing, as described in section 5.4.3.

Else, if the card indicates mag-stripe mode is not supported (AIP byte 2 bit 8 is 0b), the kernel shall continue processing and for all subsequent kernel processing with different behaviour based on path, the kernel shall proceed with EMV Mode Path Processing, described in section 5.4.3.

5.4.3 EMV Mode Path Processing – Determine Card Transaction Disposition

The kernel shall perform the procedure defined in this section for EMV mode transactions.

Requirements – EMV Mode – Determine Card Transaction Disposition

5.4.3.1 (Cryptogram Information Data) [5.69]

If the card does not return the Cryptogram Information Data (CID),
then the kernel shall:

- Construct the CID and initialize it with a value of '00'.
- Set CID bits 8-7 to the value of Issuer Application Data byte 5 bits 6-5³ using identical bit settings.

5.4.3.2 (Cryptogram Type Transaction Disposition) [5.70]

The kernel examines the Cryptogram Information Data (CID) to determine the cryptogram type (TC, ARQC, or AAC).

EMV Book C-3, Ex[1027] §5.2.2.3 at 38, § 5.4.3.2 at 62 (instructing continued processing of the response to the GPO command at § 5.4.3, including examination of cryptogram at § 5.4.3.2, consistent with §2.4.1's disclosure that a cryptogram is returned in the same response to GPO)

Paper 1 at 49-50, Paper 26 at 14-15

[1.E-3] R-APDU w/ Cryptogram and Account ID

- Collinge is compatible with EMV because its single-use key, KS_{UN} , is a 16-byte value (Ex[1050] ¶35), and thus can be used directly as the 16-byte Application Cryptogram Session Key for EMV mode transactions.

8.1.2 Application Cryptogram Algorithm

The method for Application Cryptogram generation takes as input a unique 16-byte ICC Application Cryptogram Master Key MK_{AC} and the data selected as described in section 8.1.1, and computes the 8-byte Application Cryptogram in the following two steps:

1. Use the session key derivation function specified in Annex A1.3 to derive a 16-byte Application Cryptogram Session Key SK_{AC} from the ICC Application Cryptogram Master Key MK_{AC} and the 2-byte Application Transaction Counter (ATC) of the ICC.
2. Generate the 8-byte Application Cryptogram by applying the MAC algorithm specified in Annex A1.2 to the data selected and using the 16-byte Application Cryptogram Session Key derived in the previous step.

A1.3.1 Common Session Key Derivation Option

The common session key derivation option generates a unique session key for each transaction performed by the application. It does this by enciphering an 8-byte diversification value with the 16-byte ICC Master Key (MK) to produce a 16-byte ICC Session Key (SK).

The 8-byte diversification value is represented as

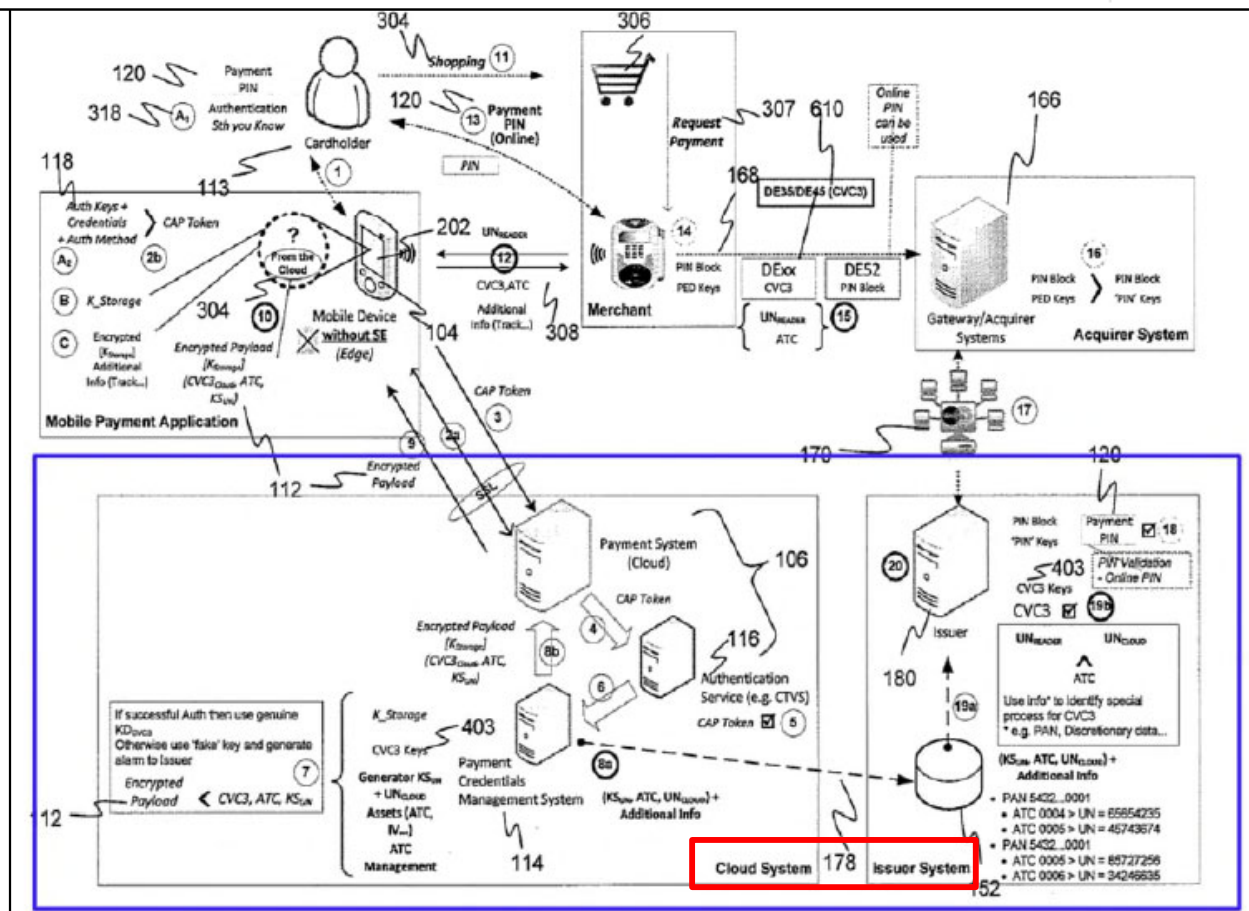
$$R = R_0 || R_1 || R_2 || R_3 || R_4 || R_5 || R_6 || R_7.$$

Ex[1020] §A1.3.1, §8.1.2.

[10/27] Remote Computer System

- Petitioner's mapping of remote computer system includes functions of authorization, which includes the issuer system connected to the "cloud system"

[0046] In one embodiment, system 100 may include a connection 178 between the issuer 180 and a payment credentials management system 114.

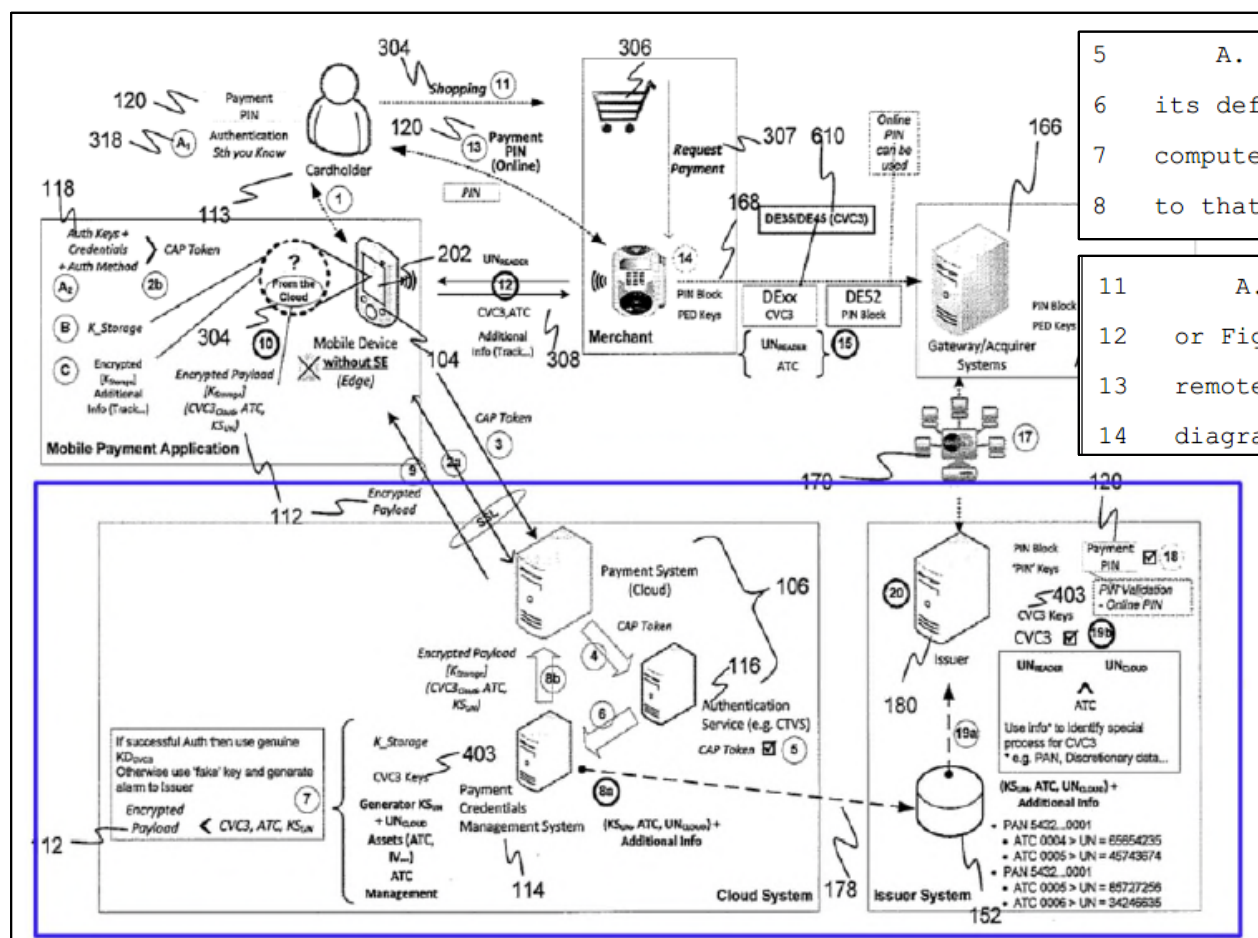


Paper 1 at 38-41,
58-59;
Paper 26 at 17

Ex[1009] [0046], Fig. 7

[10/27] Remote Computer System

- Dr. Fiske’s analysis is not limited to the cloud system— he merely acknowledged that the functionality expressly recited in claim 1 was performed in Collinge’s remote **SE** system, which is only a part of the remote computer system in the blue-boxed portion, and that system can encompass multiple computers.



5 A. The word "system" is actually quite loose in
6 its definition. It could encompass multiple
7 computers, or at some level of detail you might refer
8 to that collection of computers as a single system.

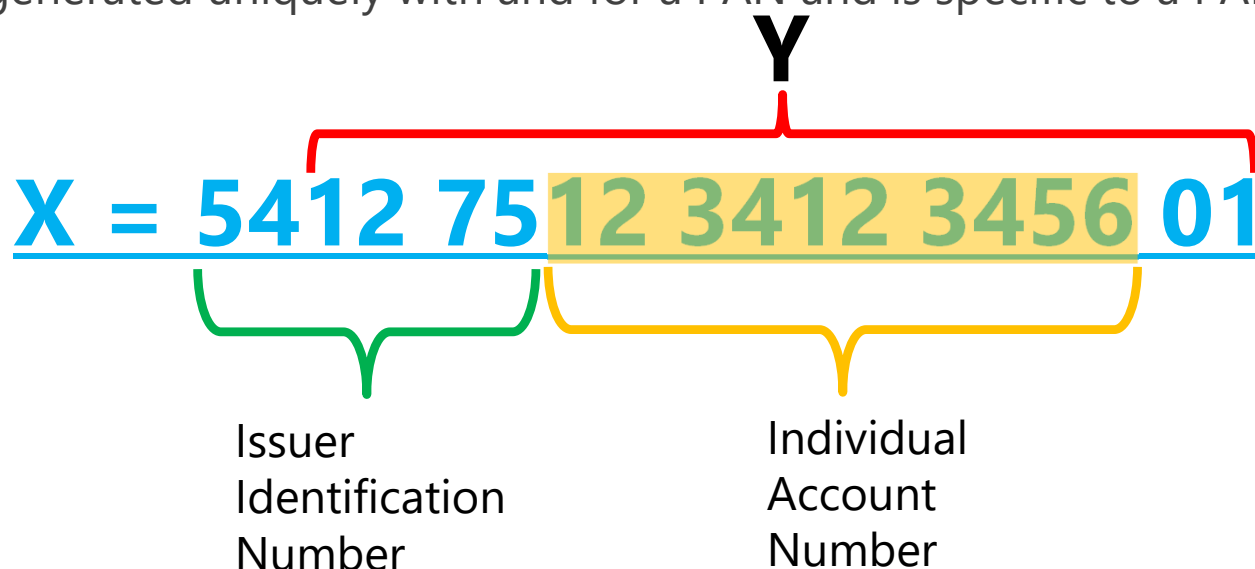
11 A. I think so. I can't see in either Figure 6
12 or Figure 7 an item identified with a 100. But the
13 remote-SE system is clearly labeled 110 in both
14 diagrams.

Ex[1049], 25:25-26:8, 73:3-75:9

Ex[1009] Fig. 7

[10.B]/[27.B] Remote Computer System

- See [1.C] above:
 - Collinge Provisionals do not teach that retrieval of KSUN is done using only the ATC—it is generated also with KDUN, which is in turn generated from PAN (values X and Y)
 - KSUN is generated uniquely with and for a PAN and is specific to a PAN.



KD_{UN}

Reuse Value Y as defined above

Generate KD_{UN} using:

$Z_L := \text{DES3}(\text{IMK}_{\text{UN}})[Y]$

$Z_R := \text{DES3}(\text{IMK}_{\text{UN}})[Y \oplus (\text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'} \parallel \text{'FF'})]$

$\text{KD}_{\text{UN}} := (Z_L \parallel Z_R)$

Ex[1009], 27

KS_{UN}

Generate KS_{UN} using:

$U_L := \text{DES3}(\text{KD}_{\text{UN}})[(\text{ATC} \parallel \text{'F0'} \parallel \text{'00'} \parallel \text{UN}_{\text{CLOUD}})]$

$U_R := \text{DES3}(\text{KD}_{\text{UN}})[(\text{ATC} \parallel \text{'0F'} \parallel \text{'00'} \parallel \text{UN}_{\text{CLOUD}})]$

$\text{KS}_{\text{UN}} := (U_L \parallel U_R)$

Ex[1009], 28

[10.C]/[27.C] Validating CVC3

- Collinge teaches that the technique for validating a cryptogram/message authentication code, such as CVC3, was “apparent” to a POSITA, and this is corroborated by the cited payment-textbook as the well-established practice for transaction authorization.

database 310. At stage 19b, the issuer processing server 308 may validate the payment CVC3 value using at least the UN_{READER} , UN_{CLOUD} , and application transaction counter. Methods for validating a CVC3 will be apparent to persons having skill in the relevant art. At stage 20, the issuer 108 may submit a response based on the validation, and the method may proceed accordingly as in traditional payment transactions.

Collinge, Ex[1011], [0087]

6.9.1 Authorization request and response with chip data

The terminal generates an authorization request message 1100. The data fields included in this message are those explained in Section 2.9 for the processing of transactions using magnetic stripe cards. In addition, the authorization request message includes the field ICC system-related data, corresponding to bit 55 in the bitmap (see Section 2.8.1) of the data elements present in the 1100 message, according to ISO 8583:1993. This field contains all the data objects from the point of service, which allows the IH to verify the correctness of the ARQC produced by the card. Among these data objects one can mention:

- Application Cryptogram (ARQC);

Radu, Ex[1037], 57

After receiving the authorization request message 1100, the IH first verifies that the card at the point of service is genuine. To this end, the cryptographic module of the IH must verify the correctness of the ARQC, according to the following steps:

- Use the key indicator present in the IAD to retrieve the correct version of the IMK_{AC} .
- Use the account identification data to derive the key MK_{AC} corresponding to the card participating in the transaction.
- Derive the actual session key SSK_{AC} from MK_{AC} , using the ATC as diversifier information in the keys-tree.
- Use the algorithm version number in the IAD to call the same algorithm for the computation of the MAC as that used in the card.
- Use SSK_{AC} to produce a MAC on the same data objects that were considered by the card in the transaction claim.
- Compare the computed MAC against the ARQC received in the authorization request message 1100. If the two values are equal, the ARQC is accepted as valid evidence that the card at the point of service is authentic and genuine.

Radu, Ex[1037], 58

[8.A] Provisioning Data incl. Account ID

- This element is obvious because Collinge discloses receiving from a remote-SE all necessary “payment credentials” to create a transaction request message, which would include “account information (e.g. PAN)”

[0009] Another method for generating and provisioning payment credentials to a mobile device lacking a secure element includes: storing, in a database, at least a storage key, a plurality of dynamic card validation code keys, and an application transaction counter associated with a mobile application program; provisioning, to the mobile device, at least the storage key, an authentication component, and static payment credentials, wherein the static payment credentials are associated with a payment account; receiving, from a mobile

Ex[1011], [0009]

- Also, it would be obvious to send an account identifier along with the key, so that the mobile device knows which account’s key it is receiving, which a POSITA would understand would be useful when designing a card capable of being used with multiple accounts or a credential wallet usable with multiple accounts. Ex[1050] ¶48.

Paper 26 at 19-20

[8.B] Post-Transaction Provisioning

- Contrary to PO's assertions (POR, 40), KS_{UN} in Collinge is not generated by the mobile device and is instead generated at the remote computer system.

[0083] At stage 7, the payment credentials management system 704 may generate an encrypted payload. As part of the encrypted payload, the payment credentials management system 704 may generate a session key unpredictable number (KS_{UN}), a cloud unpredictable number (UN_{CLOUD}), and may identify and/or store a plurality of dynamic card validation code (CVC3) keys and the $K_{STORAGE}$. Methods and systems for the generation of unpredictable numbers will be apparent to persons having skilled in the relevant art. The encrypted payload may include at least one CVC3 key, the KS_{UN} , and the application transaction counter and may be generated using a derived CVC3 key (KD_{CVC3}). In one embodiment, the

Ex[1011], [0083]

- Therefore, logically and obviously, when the mobile device uses up the one-time key KS_{UN} after an interrogation (i.e. transaction), it should request a new KS_{UN} for the next transaction, and in response, the remote computer system would generate and send a new set of credential data for that account.

[4/15/20/32] Compliance w/ Card Specification

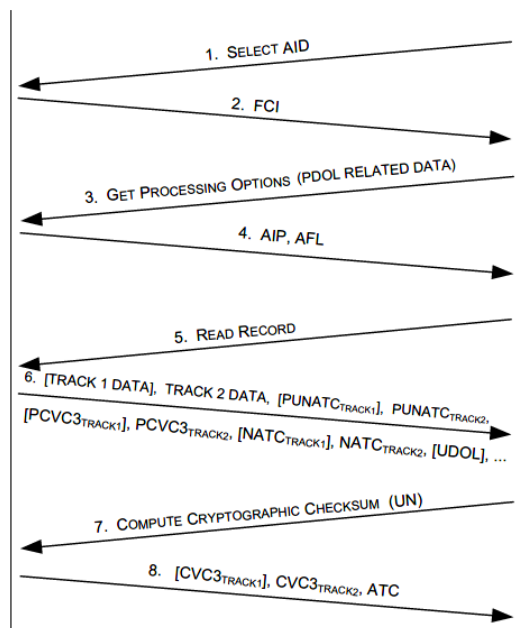
- The Collinge Provisionals state that the invention enables PayPass transactions, where interrogation is standardized, including by the PayPass Specification and as evidenced by the '171 Patent itself.

3.1 Overview of Command – Response APDU Flow

Figure 2 illustrates the Command – Response APDU flow of a PayPass – Mag Stripe transaction including the SELECT AID command.

Figure 2—Command – Response APDU Flow

PayPass card PayPass reader



PayPass Spec, Ex[1058], §III.3.1 at 23

Examples of cryptograms at the time of this filing are defined by the following card specifications in table 4.

TABLE 4

Card specification	Cryptogram Designator or Name
MASTERCARD PAYPASS MSD	CVC3
VISA Contactless MSD	dCVV
DISCOVER ZIP MSD	dCVV
EMV, VISA QVSDC, VISA VSDC, MASTERCARD MCHIP	AC, TC, AAC, ARQC

Depending on the interrogation sequence selected by the point-of-sale, it may not always be possible to implement a system set forth in FIG. 17. Many point-of-sale interrogation sequences carried out at the time of this filing are based on card specifications such as:

- MASTERCARD, PAYPASS, MChip
- AMERICAN EXPRESS, EXPRESS PAY
- DISCOVER ZIP
- Visa QVSDC, VSDC
- Contactless EMV

'171 Patent, Ex[1001] at 25:50-63, 31:55-60

Paper 26 at 21

[6/22] Use of 3DES Algorithm

- The Collinge Provisionals explicitly state that the cryptogram is generated using 3DES algorithm

2. Calculate T as follows:

$$T := \text{DES3}(\text{KS}_{\text{UN}})[M]$$

The two least significant bytes of T are the $\text{CVC3}_{\text{TRACK1/2}}$

Ex[1009], [0071] at 29-30

Paper 1 at 56, Paper 26 at 21-22

[16/33] Use of an Issuer Master Key (IMK)

- Use of Issuer Master Keys to derive keys used to generate MAC cryptograms is well known and standardized:

code (CVC3) keys and the $K_{STORAGE}$. Methods and systems for the generation of unpredictable numbers will be apparent to persons having skilled in the relevant art. The encrypted

Collinge, Ex[1011] [0083]

After receiving the authorization request message 1100, the IH first verifies that the card at the point of service is genuine. To this end, the cryptographic module of the IH must verify the correctness of the ARQC, according to the following steps:

- Use the key indicator present in the IAD to retrieve the correct version of the IMK_{AC} .
- Use the account identification data to derive the key MK_{AC} corresponding to the card participating in the transaction.
- Derive the actual session key SSK_{AC} from MK_{AC} , using the ATC as diversifier information in the keys-tree.

Radu, Ex[1037], 58

8.1.2 Application Cryptogram Algorithm

The method for Application Cryptogram generation takes as input a unique 16-byte ICC Application Cryptogram Master Key MK_{AC} and the data selected as described in section 8.1.1, and computes the 8-byte Application Cryptogram in the following two steps:

1. Use the session key derivation function specified in Annex A1.3 to derive a 16-byte Application Cryptogram Session Key SK_{AC} from the ICC Application Cryptogram Master Key MK_{AC} and the 2-byte Application Transaction Counter (ATC) of the ICC.
2. Generate the 8-byte Application Cryptogram by applying the MAC algorithm specified in Annex A1.2 to the data selected and using the 16-byte Application Cryptogram Session Key derived in the previous step.

A1.3.1 Common Session Key Derivation Option

The common session key derivation option generates a unique session key for each transaction performed by the application. It does this by enciphering an 8-byte diversification value with the 16-byte ICC Master Key (MK) to produce a 16-byte ICC Session Key (SK).

A1.4 Master Key Derivation

This annex specifies two optional methods for the derivation by the issuer of a 16-byte ICC Master Key used for Application Cryptogram generation, issuer authentication, and secure messaging.

Note that neither method is mandatory. Issuers may decide to adopt an alternative method for this function.

These methods take as input the PAN and PAN Sequence Number, plus a 16-byte Issuer Master Key IMK, and produce the 16-byte ICC Master Key MK in the following way:

Ex[1020] §§8.1.2, A1.3.1, A1.4

[16/33] Use of an Issuer Master Key (IMK)

- Dr. Fiske did not say that Collinge departed from EMV's key derivation approach of using issuer master keys in session key derivation. Cf. POR, 48-49. He merely noted that Collinge uses an additional piece of data (UNCLOUD) in that process, which is not prohibited by the EMV standard.

12 Q. So how would one be able to predict or
13 calculate the session key KSUN without the knowledge
14 of the issuer master key?
15 A. You would not be able to predict the value of
16 KSUN unless you knew the value of the issuer master
17 key.
18 Q. Okay, that's what I thought maybe you were
19 trying to say. I think your previous answer, maybe
20 you left the word "not" out.
21 A. Okay.
22 Q. That might --
23 A. Let me just explain my understanding of the
24 process of generating KSUN.
25 So as with EMV, you start with an issuer

1 master key. You derive a card-specific key by using
2 the issuer master key and operating over the PAN and
3 the PAN sequence number, and that's defined in EMV.
4 What is then defined in EMV is a mechanism
5 where you take the card key and you operate over the
6 application transaction counter, and that generates
7 the session key. And that mechanism is in EMV, and
8 it's also in Fiske.
9 What Collinge does is, instead of just
10 iterating -- or encrypting the --
11 No, instead of deriving the session key
12 using just the application transaction counter, they
13 also use what they call UNCLOUD, which is the result
14 of the random number generation. So that's why they
15 call it a session key unpredictable number, to
16 differentiate it from a normal EMV session key.
17 That's how I would understand it.

Ex[1049] 77:12-78:17

Paper 26 at 22

GROUNDS 3/4 - FISKE 05

[8/9] Provisioning Data incl. Account ID

User device **502** is provisioned with a payment application associated with issuing bank **500**. The user of device **502** can interact with a terminal **504** at a PoS via user device **502** in order to make a purchase from a merchant. The PoS terminal **504** can communicate with the acquiring bank **506**, and the transaction is later settled between the issuing bank **500** and the acquiring bank **506**, once the proper transfer of funds has been arranged.

Ex[1012] at 11:34-44 (discussing provisioning the user device with data sufficient to make a purchase);

In response to receiving the READ RECORD commands, or other commands (such as the aforementioned GPO), the ICC sends all of the records that are identified in the AFL to the terminal. The records relevant to offline data authentication include a Certification Authority (CA) Public Key Index, an Issuer Public Key Certificate, the Primary Account Number (PAN) and Signed Static Application Data or an ICC Public Key Certificate depending on the offline data authentication method being employed.

Ex[1012] at 3:44-46 (discussing PAN sent to terminal from the mobile application);

The terminal then checks the Issuer Identifier recovered from the Issuer Public Key Certificate against first 3-8 digits of the Primary Account Number read from the ICC.

Ex[1012] at 4:49-41 (discussing the mobile application sending the account number to the terminal during a transaction).

- It is obvious to a POSITA to include PAN in the credential data provisioned to the mobile device, based on Fiske's multiple references to how this PAN would be included with the cryptogram by the card when requesting payment authorization.

[8/9] Provisioning Data incl. Account ID

- Like Collinge, the non-permanent cryptographic key in Fiske (ICC Session Key) is specific to an account and only usable for that account.

45 According to embodiments of the invention, the payment
keys, which are necessary for the use of the payment
application, are provided subsequently to the installation of
the payment application, under the control of the issuing
bank. Equipping the user device with the payment keys in
50 this manner has the effect of activating the payment appli-
cation, thereby associating it with an account at the issuing
bank, and enabling it to conduct EMV payment transactions.
The payment keys may be stored in an encrypted state in a
memory portion associated with the standard application
55 environment such as in read only memory 604 or an alter-
native persistent memory, using, for example, the Advanced
Encryption Standard (AES). The key used to encrypt and

In addition, the payment application is configured such
that the ICC Master Key is not held locally on the user
device, and instead held by a remote entity such as the
issuing bank. The user device is provisioned with an ICC 20
Session Key, generated by the issuing bank on the basis of
the ICC Master Key. The ICC Session Key can be generated
by the issuing bank in accordance with, for example, the
method described above in relation to FIG. 4.

Ex[1012], 12:45-52, 12:50-57, 13:17-24.

- Therefore, as explained, it would be obvious to send an account identifier along with the key, so that the mobile device knows which account's key it is receiving, which a POSITA would understand would be useful when designing a card capable of being used with multiple accounts or a credential wallet usable with multiple accounts. Ex[1050] ¶¶ 48, 60.

[10-13/27-30] Standardized Authorization

- These claims are directed to well-established, standardized authorization process, all of which were long known in the art, as Fiske disclosed and evidenced and corroborated by textbook materials
 - “authorization data comprises the response cryptogram and the account identifier”

6.9.1 Authorization request and response with chip data

The terminal generates an authorization request message 1100. The data fields included in this message are those explained in Section 2.9 for the processing of transactions using magnetic stripe cards. In addition, the authorization request message includes the field ICC system-related data, corresponding to bit 55 in the bitmap (see Section 2.8.1) of the data elements present in the 1100 message, according to ISO 8583:1993. This field contains all the data objects from the point of service, which allows the IH to verify the correctness of the ARQC produced by the card. Among these data objects one can mention:

- Application Cryptogram (ARQC);
- Account identification data of the card (PAN and PAN Sequence Number);
- ATC and the unpredictable number;

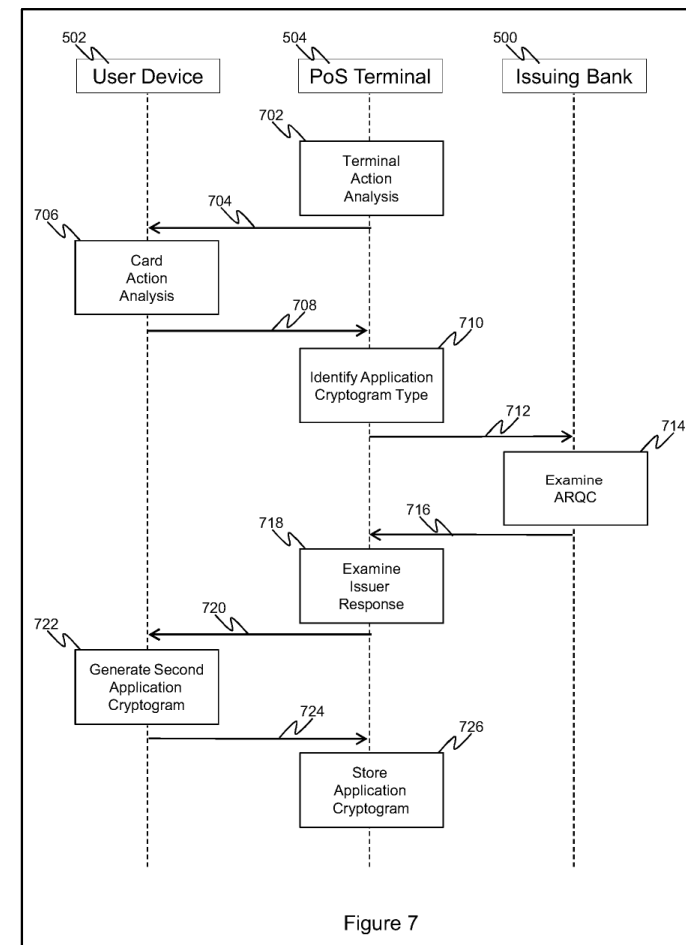
Radu, Ex[1037], 57

[10-13/27-30] Standardized Authorization

- These claims are directed to well-established, standardized authorization process, all of which were long known in the art, as Fiske disclosed and evidenced and corroborated by textbook materials
 - “determining a...key associated with the account”

The issuer then examines the ARQC in step 714 in order to make a decision on whether to authorize the transaction. Optionally, the issuer may identify which Session Key was used to generate the ARQC, determine the amount of time that has passed since that ICC Session Key was provisioned to the user device, and make the decision on whether to authorize the transaction on the basis of whether that amount of time exceeds the aforementioned threshold amount.

Ex[1012], 16:13-14;Fig. 7



[10-13/27-30] Standardized Authorization

- These claims are directed to well-established, standardized authorization process, all of which were long known in the art, as Fiske disclosed and evidenced and corroborated by textbook materials
 - “a calculated cryptogram”

After receiving the authorization request message 1100, the IH first verifies that the card at the point of service is genuine. To this end, the cryptographic module of the IH must verify the correctness of the ARQC, according to the following steps:

- Use the key indicator present in the IAD to retrieve the correct version of the IMK_{AC} .
- Use the account identification data to derive the key MK_{AC} corresponding to the card participating in the transaction.
- Derive the actual session key SSK_{AC} from MK_{AC} , using the ATC as diversifier information in the keys-tree.
- Use the algorithm version number in the IAD to call the same algorithm for the computation of the MAC as that used in the card.
- Use SSK_{AC} to produce a MAC on the same data objects that were considered by the card in the transaction claim.
- Compare the computed MAC against the ARQC received in the authorization request message 1100. If the two values are equal, the ARQC is accepted as valid evidence that the card at the point of service is authentic and genuine.

Radu, Ex[1037], 58

[11/28/12/29] Use of Unpredictable Data

- Again, use of unpredictable data for generation of cryptogram and authorization is well-established, standardized authorization process, long known in the art, as evidenced and corroborated by textbook materials

8.1.1 Data Selection

An Application Cryptogram consists of a Message Authentication Code (MAC) generated over data:

- referenced in the ICC's DOLs and transmitted from the terminal to the ICC in the GENERATE AC or other command, and
- accessed internally by the ICC.

The recommended minimum set of data elements to be included in Application Cryptogram generation is specified in Table 26.

Value	Source
Amount, Authorised (Numeric)	Terminal
Amount, Other (Numeric)	Terminal
Terminal Country Code	Terminal
Terminal Verification Results	Terminal
Transaction Currency Code	Terminal
Transaction Date	Terminal
Transaction Type	Terminal
Unpredictable Number	Terminal
Application Interchange Profile	ICC
Application Transaction Counter	ICC

EMV Book 2, Ex[1020] §8.1.1 at 100

6.9.1 Authorization request and response with chip data

verify the correctness of the ARQC produced by the card. Among these data objects one can mention:

- Application Cryptogram (ARQC);
- Account identification data of the card (PAN and PAN Sequence Number);
- ATC and the unpredictable number;

- Use the algorithm version number in the IAD to call the same algorithm for the computation of the MAC as that used in the card.
- Use SSK_{AC} to produce a MAC on the same data objects that were considered by the card in the transaction claim.
- Compare the computed MAC against the ARQC received in the authorization request message 1100. If the two values are equal, the ARQC is accepted as valid evidence that the card at the point of service is authentic and genuine.

Radu, Ex[1037], 57-58

- A POSITA understands that the "response cryptogram" is the cryptogram generated by the mobile device (card) in response to a COMPUTE CRYPTOGRAPHIC CHECKSUM, a GENERATE AC, or a GPO command. See [supra](#) p. 54, [supra](#) p. 55

[13/30] Inclusion of Transaction Data

- The entire point of the authorization in any “payment system” (the title of Fiske’s invention) is for the account issuer to authorize a transaction, which means the issuer must receive the transaction data, such as the amount user is attempting to spend, and in which currency, to determine whether to permit the payment.

8.1.1 Data Selection

An Application Cryptogram consists of a Message Authentication Code (MAC) generated over data:

- referenced in the ICC’s DOLs and transmitted from the terminal to the ICC in the GENERATE AC or other command, and
- accessed internally by the ICC.

The recommended minimum set of data elements to be included in Application Cryptogram generation is specified in Table 26.

Value	Source
Amount, Authorised (Numeric)	Terminal
Amount, Other (Numeric)	Terminal
Terminal Country Code	Terminal
Terminal Verification Results	Terminal
Transaction Currency Code	Terminal
Transaction Date	Terminal
Transaction Type	Terminal
Unpredictable Number	Terminal
Application Interchange Profile	ICC
Application Transaction Counter	ICC

EMV Book 2, §8.1.1 at 100

8.1.2 Application Cryptogram Algorithm

The method for Application Cryptogram generation takes as input a unique 16-byte ICC Application Cryptogram Master Key MK_{AC} and the data selected as described in section 8.1.1, and computes the 8-byte Application Cryptogram in the following two steps:

- Use the session key derivation function specified in Annex A1.3 to derive a 16-byte Application Cryptogram Session Key SK_{AC} from the ICC Application Cryptogram Master Key MK_{AC} and the 2-byte Application Transaction Counter (ATC) of the ICC.
- Generate the 8-byte Application Cryptogram by applying the MAC algorithm specified in Annex A1.2 to the data selected and using the 16-byte Application Cryptogram Session Key derived in the previous step.

EMV Book 2, §8.1.2 at 101

6.9.1 Authorization request and response with chip data

verify the correctness of the ARQC produced by the card. Among these data objects one can mention:

- All the data objects necessary to reproduce the transaction claim.

Radu, Ex[1037], 57-58

[1.E-2] AC Generation

- A POSITA would know, even without knowing how the Application Cryptogram (AC) is generated, that the encryption key will not be included in the resulting cryptogram; otherwise anyone with the cryptogram could use the included key to create a fraudulent payment request. Ex[1050] ¶169.
- EMV uses 3DES as approved algorithm for AC generation. See also [supra](#) p.51.

A1.2 Message Authentication Code

The computation of an s-byte MAC ($4 \leq s \leq 8$) is according to ISO/IEC 9797-1 using a 64-bit block cipher ALG in CBC mode. More precisely, the computation of a MAC S over a message MSG consisting of an arbitrary number of bytes with a MAC Session Key K_S takes place in the following steps.

B1.1 Data Encryption Standard (DES)

The double-length key triple DES encipherment algorithm (see ISO/IEC 18033-3) is the approved cryptographic algorithm to be used in the encipherment and MAC mechanisms specified in Annex A1. The algorithm is based on the (single) DES algorithm standardised in ISO 16609.

Triple DES encipherment involves enciphering an 8-byte plaintext block in an 8-byte ciphertext block with a double-length (16-byte) secret key $K = (K_L \parallel K_R)$ as follows:

$$Y = \text{DES}_3(K)[X] = \text{DES}(K_L)[\text{DES}^{-1}(K_R)[\text{DES}(K_L)[X]]]$$

Decipherment takes place as follows:

$$X = \text{DES}^{-1}(K_L)[\text{DES}(K_R)[\text{DES}^{-1}(K_L)[Y]]]$$

Single DES is only approved for usage with the version of the MAC mechanism specified in Annex A1 using Algorithm 3 of ISO/IEC 9797-1 (triple DES applied to the last block).

EMV Book 2, Ex[1020] §§A1.2, B1.1.

[1.E-3] Use of APDU

- As explained for grounds 1 & 2, use of APDU in the communications between the card and the terminal is standardized practice. See [supra](#) p.43, [supra](#) p.46

6.5 Commands

This section describes the following APDU command-response pairs:

- APPLICATION BLOCK (post-issuance command)
- APPLICATION UNBLOCK (post-issuance command)
- CARD BLOCK (post-issuance command)
- EXTERNAL AUTHENTICATE
- GENERATE APPLICATION CRYPTOGRAM
- GET CHALLENGE
- GET DATA
- GET PROCESSING OPTIONS
- INTERNAL AUTHENTICATE
- PIN CHANGE/UNBLOCK (post-issuance command)
- READ RECORD
- VERIFY

The post-issuance commands shall only be sent using script processing (see section 10.10) and secure messaging as specified in Book 2.

EMV Book 3, Ex[1021] §6.5

Table 2.2—Reader Functionality

Functionality	Explanation
Communication with the Card	This includes the protocol for the contactless interface as defined by [EMV Book D] and the exchange of APDUs as defined in [ISO/IEC 7816-4].

EMV Book C-2, Ex[1026] Table 2.2

3.2.3 Command and response format

In accordance with the OSI 7-layer model, the information transaction exchanged between the card and the terminal can be divided into three protocol sections:

- Application protocols for command and response data (layer 7). A step in an application protocol consists of sending a command application protocol data unit (C-APDU) from the terminal application to the card application. The latter processes it and sends back the response application protocol data unit (R-APDU) to the terminal application. A schematized picture of a C-APDU/R-APDU pair is given in Figure 3.5.

If the ICC performs a combined DDA/AC generation, the terminal can verify off-line that the card is genuine by verifying the correctness of the Signed Dynamic Application Data returned in the R-APDU of the GENERATE AC command.

Radu, Ex[1037], 46-47, 55-56

[1.E-3] Use of APDU

- Petitioner erroneously claimed that the APDU format defined in ISO 7816-4 does not apply to contactless. (POR at 66.) That is wrong and contradicted by the '171 Patent itself.
- EMV Contactless is built upon EMV Contact and generally share the C-ADPU and R-ADPU pairs:

5 C-APDU Commands

This chapter defines the commands and responses supported by the Kernel:

- 5.1 Introduction
- 5.2 COMPUTE CRYPTOGRAPHIC CHECKSUM
- 5.3 GENERATE AC
- 5.4 GET DATA
- 5.5 GET PROCESSING OPTIONS
- 5.6 PUT DATA
- 5.7 READ RECORD
- 5.8 RECOVER AC

EMV Book C-2 (contactless), Ex[1026] §5

6.5 Commands

This section describes the following APDU command-response pairs:

- APPLICATION BLOCK (post-issuance command)
- APPLICATION UNBLOCK (post-issuance command)
- CARD BLOCK (post-issuance command)
- EXTERNAL AUTHENTICATE
- GENERATE APPLICATION CRYPTOGRAM
- GET CHALLENGE
- GET DATA
- GET PROCESSING OPTIONS
- INTERNAL AUTHENTICATE
- PIN CHANGE/UNBLOCK (post-issuance command)
- READ RECORD
- VERIFY

The post-issuance commands shall only be sent using script processing (see section 10.10) and secure messaging as specified in Book 2.

EMV Book 3, Ex[1021] §6.5

[1.E-3] Use of APDU

- ISO/IEC 14443 adopts ISO/IEC 7816-4 for APDU message structure.

The protocol as defined in this part of ISO/IEC 14443 is capable of transferring the application protocol data units as defined in ISO/IEC 7816-4. Thus, application protocol data units may be mapped as defined in ISO/IEC 7816-4 and application selection may be used as defined ISO/IEC 7816-5.

ISO/IEC 14443 is intended to allow operation of proximity cards in the presence of other contactless cards conforming to ISO/IEC 10536 and ISO/IEC 15693 and Near Field Communication (NFC) devices conforming to ISO/IEC 18092 and ISO/IEC 21481.

ISO/IEC 14443, Ex[1063],6

1.1 Underlying Standards

This specification is based on the ISO/IEC 7816 and ISO/IEC 14443 series of standards and should be read in conjunction with those standards. However, if any provision or definition in this specification differs from those standards, the provision or definition herein shall take precedence.

EMV (contactless)
Book B, Ex[1024], 9

- This is confirmed by '171 Patent itself.

Ex[1001], col. 6

	ISO/IEC 7816 (contact)	ISO/IEC 14443 (contact less)	+	OSI- Layer
Application commands (APDUs)	7816-4			7
Transport protocols, Activation, Bit Transmission, Powersupply	7816-3	14443-4	Transport- protocol	4
		14443-3	Anticollision, Activation	2
		14443-2	Powersupply, Bit Transmission	1
Contacts	7816-2			
Phys. properties	7816-1	14443-1	Phys. properties	

Paper 26 at 27-28

[1.E-3] EMV Mode Visa Transactions

- See [supra](#) p.55.
- A POSITA would know to implement Fiske (a patent assigned to Visa) to conduct an EMV mode Visa transaction because Fiske explicitly teaches that it is for implementation of EMV transactions, which include Visa transactions—indeed, EMV standards devote an entire book (C-3) for contactless Visa EMV transactions.

(12) **United States Patent
Fiske**

(54) **PAYMENT SYSTEM**

(71) Applicant: **Visa Europe Limited**, London (GB)

(72) Inventor: **Stuart Fiske**, Guildford (GB)

(73) Assignee: **VISA EUROPE LIMITED**, London (GB)

55 Hence, it is an objective of the present invention to provide improved methods for providing payment card functionality on a user device, while minimizing any inconvenience to the user of the device and without requiring modification of the EMV infrastructure, or existing PoS hardware.

Fiske, Ex[1012], 2:54-59

[1.E-3] EMV Mode Visa Transactions

- Patent Owner points to EX1027 at 84 (Section 6.1) to argue that “there is no GENERATE AC command.” (POR 71.)
- But that section relates to a processing flow where the payment card is required to be presented twice to the terminal in order to perform “update processing,” not a standard basic transaction.

6 Issuer Update Processing Reader Requirements

Whilst the standard new transaction processing only utilizes one card presentment, it is possible to request a second presentment in support of issuer update processing. This requires both card and reader to support a second presentment of a card after an issuer response to an online request. If so, and Issuer Update Data has been

Ex[1027], 83-84

6.1 Issuer Update Processing Commands

To facilitate Issuer Update Processing, support for the EXTERNAL AUTHENTICATE command and issuer script processing is required.

Requirements – Issuer Update Processing Commands

6.1.1.1 (EXTERNAL AUTHENTICATE Command) [5.88]

The kernel shall support the EXTERNAL AUTHENTICATE command, as defined in section D.3.

6.1.1.2 (Issuer Scripts) [5.89]

The kernel shall support Issuer Scripts according to [EMV 4.2 Book 3], section 10.10, and [EMV 4.2 Book 4], section 6.3.9, except for the following:

- Unlike [EMV 4.2], Issuer Script Templates with Tag '71' or Tag '72' are processed and issued by the kernel in the same manner. There is no GENERATE AC command, and Issuer Script Templates with Tag '71' and '72' are both processed during Issuer Update Processing.
- No processing is performed on the Transaction Status Information (TSI) or Terminal Verification Results (TVR).

[4/5/20/21/6/15/22/32] Specification Compliance

- As explained above, a POSITA reading Fiske would understand it was meant to be implemented using standard EMV practices—whether a portion of the standard is “mandatory” is irrelevant to the inquiry of obviousness.

Hence, it is an objective of the present invention to
55 provide improved methods for providing payment card
functionality on a user device, while minimizing any incon-
venience to the user of the device and without requiring
modification of the EMV infrastructure, or existing PoS
hardware.

Fiske, Ex[1012], 2:54-59

AUSTIN

BRUSSELS

DALLAS

DUBAI

HOUSTON

LONDON

NEW YORK

PALO ALTO

RIYADH

SAN FRANCISCO

SINGAPORE

WASHINGTON

bakerbotts.com

©Baker Botts L.L.P., 2024. Unauthorized use and/or duplication of this material without express and written permission from Baker Botts L.L.P. is strictly prohibited. Excerpts and links may be used, provided that full and clear credit is given with appropriate and specific direction to the original content.