USENIX
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

The following paper was originally published in the

USENIX Workshop on Smartcard Technology

Chicago, Illinois, USA, May 10–11, 1999

Risks and Potentials of Using EMV for Internet Payments

Els Van Herreweghen
IBM Zurich Research Laboratory

Uta Wille
Jelmoli Information Systems

For more information about the USENIX Association:
Phone: 1 510 528 8649 FAX: 1 510 548 5738
Email: office@usenix.org WWW: http://www.usenix.org

Risks and Potentials of using EMV for Internet Payments

Els Van Herreweghen
IBM Zurich Research Laboratory, Ruschlikon, Switzerland
Email: evh@zurich.ibm.com
Uta Wille*
Jelmoli Information Systems AG, Zurich, Switzerland
Email: wille_u@jelmoli.ch

Abstract

Existing payment smartcards developed for traditional point-of-sale transactions are being considered for use in Internet transactions. Such solutions have been suggested as alternatives to using payment protocols more specifically designed for Internet payments (such as SET [8]) but often lacking smartcard support. In this paper, we analyze EMV'96 [7], a representative example of an existing payment smartcard specification. We investigate which security requirements for an Internet payment system can and cannot be met when using EMV for Internet payments. We suggest possible modifications that can enhance the security of an Internet payment scheme based on EMV.

1. Introduction

With the growing use of the Internet for commercial transactions, there has been much effort in developing systems and protocols for securing payments on the Internet. A prominent example of such a protocol is the Secure Electronic Transaction (SET, [8]) protocol. It is, however, not designed with smartcard support in mind. Current implementations require the customer to make SET transactions from a fixed, trusted personal computer. A secure SET implementation on a smartcard for use with public (and untrusted) terminals would require the smartcard to store the user's account data and cryptographic keys as well as the SET client implementation, which is not feasible with current smartcard technology.

The lack of portability of Internet-specific systems such as SET has caused the payment industry to look at the possibility of using existing debit and credit payment smartcards for Internet payments. A standard in this area is the EMV'96 Specification [7], which describes the functionality required by such smartcard-based payment systems.

The objective of this paper is to discuss the potentials and restrictions of using EMV payment cards for debit and credit payments over the Internet. In Section 2 we formulate a set of general security requirements for smartcard-based debit and credit Internet payments. After summarizing EMV'96 security mechanisms in Section 3, we analyze in Section 4 the security properties of using EMV 'as is' for Internet payments, by checking the resulting protocols against the formulated requirements. Since the Internet scenario differs from the scenario assumed by EMV'96, these protocols show a number of vulnerabilities. In Section 5, we finally discuss mechanisms to increase the security of using EMV in the Internet scenario.

2. Model and Security Requirements for Smartcard Internet Payments

Our model of a generic Internet payment system (Figure 1) consists of a customer and a merchant who exchange money for goods or receipts, and at least one financial institution linking electronic payments to the transfer of "real money" [1].

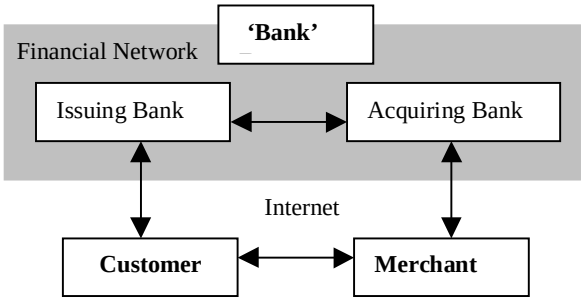

Figure 1. Payment Model.

Customer and merchant communicate over an open network (the Internet) with each other and with their banks (issuing bank and acquiring bank, respectively).

* This paper reports on work done at the IBM Zurich Research Laboratory, Ruschlikon, Switzerland.

During a transaction, actual connectivity may be limited to certain subsets of players. In a typical online purchase scenario, the customer only has a connection to the merchant, and communicates indirectly with his issuing bank (e.g., through an authorization message sent to the merchant and forwarded by the merchant to acquiring and issuing banks). The underlying communication model, however, does not influence the security requirements stated in the following.

Before formulating security requirements for a payment transaction, we need to make a number of assumptions about trust relations and liability distributions between the parties involved:

A1. Issuer and acquirer enjoy some degree of mutual trust and share an infrastructure for secure communication. This allows us to join their security requirements into "bank" requirements.
A2. Money transfers between accounts are traceable. This gives the user some assurance of refund in case of fraud by hackers or bank insiders.
A3. A contract determines the business, trust, responsibility, and liability relationships between the merchant and the bank. The contract especially defines valid payments by specifying the requirements to be fulfilled to provide the merchant with a payment guarantee.
A4. A contract determines the business, trust, responsibility, and liability relationships between the customer and the issuing bank. The contract defines what the bank considers proofs of payment by the customer and specifies the requirements for liability and disputability.
A5. The customer (user) can trust critical parts of his or her system to enable secure authorization of a transaction. In the specific case of the user's payment instrument being a smartcard authorizing the payment on the user's behalf, the user interacts with the card reader (or electronic wallet) by verifying output (e.g., transaction amount, merchant ID) on its display, and by entering data (e.g., PIN-code) on a keyboard or PIN-pad. The user can trust that:
    - The correct transaction data are displayed;
    - Secret data such as a PIN-code entered by the user is not exposed or intercepted.

We now list the requirements on a payment protocol in the above model. Requirements R1 to R7 apply to electronic payment protocols in general. Requirement R8 is related to controlling access to the customer's payment instrument and is treated with a special focus on the use of smartcards.

A number of requirements deal with proof of *authorization of the transaction by an authorizing party to a verifying party*. This is achieved by an authorization message containing a non-forgeable cryptographic proof of authentication by the authorizing party of critical transaction-related data, satisfying the following properties:

- The verifying party can verify authenticity and integrity of the critical data in the authorization message, and can verify that the data originated from the authorizing party;
- The message cannot be used to authorize another transaction (non-replayable); nor can it be used in any other way by an attacker to falsely authorize another transaction on behalf of the customer. The last requirement applies to schemes where secret authorization data (such as a PIN) is sent to the bank for verification. In such cases, this requirement translates into the requirement that this data be confidentiality-protected (encrypted) during transfer from card to bank.

As in [3], we furthermore distinguish between *weak* and *undeniable* proofs of authorization. A weak proof (e.g., shared-key based EMV Application Cryptogram, Section 3.3) cannot serve as a proof for third parties while an undeniable proof (based on a digital signature) provides non-repudiation and therefore can be used in the case of disputes. Based on these notions we formulate the following security requirements for a payment protocol:

R1. **Authorization customer to bank**. The bank posseses a payment authorization from the customer before debiting the customer's account.
R2. **Authorization merchant to bank**. The bank only authorizes a payment to a merchant if the corresponding transaction has been authorized by that merchant.
R3. **Payment guarantee for merchant** before delivery of goods. This is achieved by either of:
    i. *Authorization of the transaction by the bank*;
    ii. *Authorization of the transaction by the customer*, where the bank guarantees customer-approved transactions (see assumption A3).
R4. **Authentication and certification of merchant to customer**. The customer has a minimum of authenticated and certified information about the merchant s/he makes a payment to.
R5. **Payment receipt for customer**. After completion of the payment, the customer possesses a proof that the payment was successful. This can either be:

i. Explicit *payment receipt from the merchant*;
ii. *Payment receipt from the bank*.

It is sometimes assumed that a receipt can be replaced by a statement of account [3].

R6. **Atomicity of payments.** No party can benefit from an interrupted protocol run.

R7. **Privacy, anonymity**. The customer may require privacy of order and payment information and possibly anonymity (from eavesdroppers and eventually from merchants and/or banks).

R8. **Cardholder authorization**. The customer's payment system should be protected against unauthorized use. In the case of smartcard-based payments, unauthorized use of the card should be protected against (e.g. through use of a PIN). As mentioned in A5, the customer also needs to trust at least the terminal (or electronic wallet) s/he's using in conjunction with the smartcard.

3. Security Mechanisms Provided by EMV

This section gives an overview of EMV'96 security mechanisms securing transaction flows. Mechanisms such as card and terminal risk management are not discussed here. For a detailed description of security mechanisms provided by EMV'96 we refer to [7].

Figure 2 shows the general EMV POS scenario of an IC (Integrated Circuit) terminal interacting with an IC card, with the human user presenting the card, and with the bank. (The actual EMV functionality for authorizing transactions resides with the issuing bank. Here we make abstraction of the distinction between the issuing bank and the merchant's acquiring bank.)

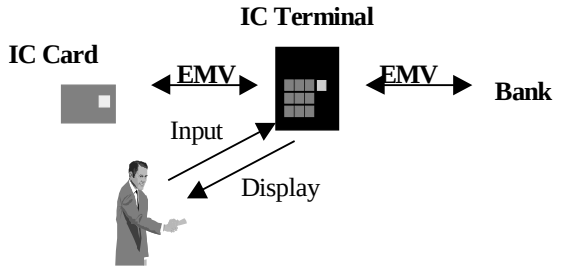

Figure 2. The EMV POS Scenario

- Terminal-card interaction consists of EMV commands issued by the terminal and card responses;
- Interaction between terminal and bank consists of the exchange of authorization requests and responses, often over a telephone connection;
- Interaction between terminal and human user consists of output to the user via the terminal display, and input by the user authorizing the transaction (such as a PIN-code).

EMV uses both asymmetric (public-key) and symmetric (shared-key) security mechanisms:

- Asymmetric security mechanisms authenticate the card as a valid card to the terminal;
- Symmetric security mechanisms generate and verify transaction cryptograms (essentially Message Authentication Codes, MACs) based on a key *k* shared between card and (issuer) bank.

The full set of security mechanisms is shown in Figure 3 which is taken from a transaction flow example in [7]. For reasons of simplicity, we make abstraction of most options and variants of the security mechanisms and focus on showing the maximum security features that can be achieved by an EMV compliant transaction.

3.1. Public-key based authentication of IC card to IC terminal

The first four messages exchanged implement the *Dynamic Data Authentication* (DDA) authenticating the card to the terminal using a public-key based challenge-response protocol. The READ_RECORD command returns the necessary Certification Authority (CA) identifier and public-key certificates needed by the terminal to authenticate the card's public key in CERT_C. CERT_C is certified by the issuer and can be verified using the issuer's public key in CERT_I, which in turn is certified by the CA and can be verified using the CA's public key known to the terminal. The actual challenge-response authentication is then executed by the terminal issuing an INTERNAL_AUTHENTICATE command containing authentication-related data (ARD), and the card responding with a signature over this data using its private signature key.

For cards without digital signature capability, EMV also provides the *Static Data Authentication* mechanism using static card data signed by the Issuer.

3.2. Cardholder Verification

EMV supports online (PIN sent to and verified by the bank) and offline (PIN verified by the card) PIN verification; the exact method supported by the card is read by the terminal with the initial READ_RECORD. Offline PIN verification is executed by the terminal issuing the VERIFY command, containing the PIN data entered by the user; the card's response indicates success or failure. The response is not cryptographically authenticated.

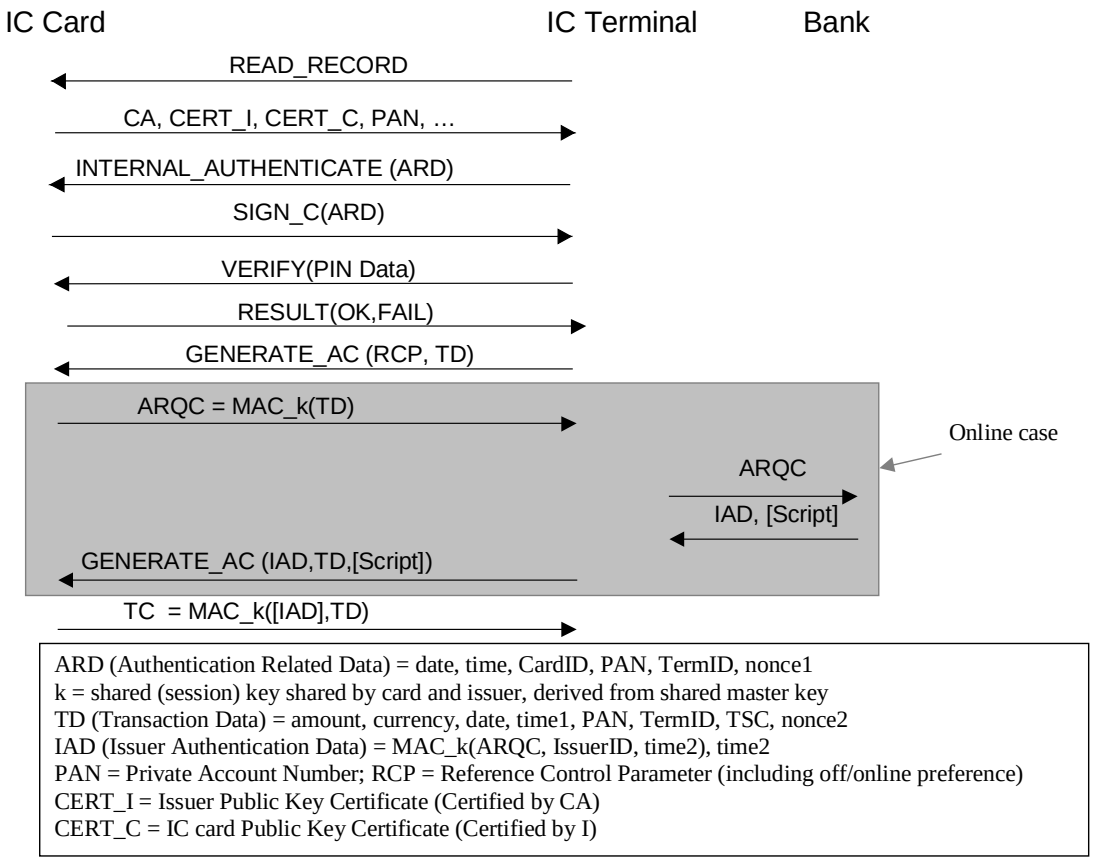

Figure 3. Model EMV Transaction Flow

3.3. Shared-key based application cryptograms and off- or online processing

The GENERATE_AC command, including Transaction Data (TD), triggers the card to produce a cryptogram that can be verified by the issuer. If both card and terminal agree on completing the transaction offline (based on both entities' risk management policies) the card returns a TC (Transaction Certificate) approving the transaction. If either card or terminal want to continue online, the card produces an ARQC (Authorization Request Cryptogram), which the terminal passes on to the bank in an *online authorization request*. If verification is successful, the bank returns an *authorization response* message containing Issuer Authentication Data (IAD) and possibly a command script to be delivered to the card. The terminal then issues the second GENERATE_AC command including the IAD and the command script.

ARQC, TC and IAD are authenticated using MACs (Message Authentication Codes). These are generated by 64-bit block ciphers using a session key *k* derived from a master key shared by the card and the issuer. The issuer can verify both ARQC and TC; in the online case the card verifies the IAD in the second GENERATE_AC command and thereby authenticates the issuer's response. The terminal triggers the generation and verification of these cryptograms but cannot verify them.

4. EMV Payments in the Internet Scenario

In the remainder of this paper, we analyze if and how EMV cards can be used for secure Internet payments.

The scenario (Figure 4) depicts a customer using his or her EMV card for online purchases from a PC that has a card acceptance device (reader) attached to it. The merchant still acts as the EMV terminal, issuing and receiving EMV commands and responses, but now communicates with customer and bank over the Internet.

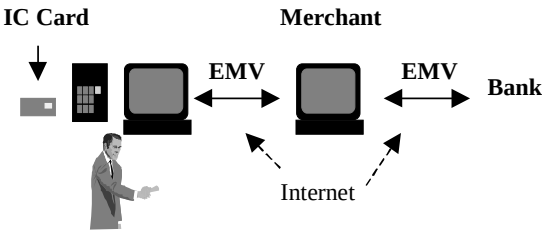

Figure 4. The EMV Internet Scenario

PIN verification deserves some special attention. While, in the POS scenario, the terminal secures the transaction by making sure the PIN is verified correctly (by card or bank), PIN verification in an Internet setting can and should no longer be controlled by the merchant:

1. *Online PIN verification* now requires the PIN to be sent from card to merchant to bank over insecure Internet connections. Even when encrypting (e.g., using SSL [5]) communication, the PIN appears in clear in the merchant's software, which is too high an exposure.
2. Even *offline PIN verification* (using VERIFY) can no longer be controlled by the merchant:
   - requiring VERIFY (including the PIN) to be issued by the merchant assumes the PIN first to be sent to the merchant over an Internet connection (and unnecessarily expose it);
   - furthermore, the merchant doesn't gain any security from the VERIFY result since it is not authenticated, and when received over the Internet, doesn't guarantee to the merchant that this was the PIN verification result produced by the card (or, stronger, that the card ever executed the VERIFY command!) .

Consequently we recommend (and assume in the following discussion) in the Internet scenario:

- Only the offline PIN authentication mechanism (VERIFY by the card) to be used;
- The VERIFY command to be issued locally (at the cardholder terminal); and
- The card to actually enforce cardholder verification by only issuing ARQC/TC after a successful VERIFY. This is currently not an explicit condition in the EMV specifications; but since in our scenario, neither merchant nor bank can enforce cardholder verification, we now have to make this an explicit condition.

We now map the online and offline transaction flows of Figure 3 to the Internet scenario of Figure 4, resulting in the 'online and offline EMV Internet flows' as depicted in Figure 5. In the following paragraph we analyze the security of these two scenarios by checking them against the security requirements defined in Section 1. Table 1 also summarizes the results of this analysis.

1. *Authorization customer to bank*. In both scenarios the transaction is weakly authorized to the bank by the customer who generates an ARQC and a TC using a key shared with the issuer.
2. *Authorization merchant to bank*. The merchant does not explicitly authorize the transaction in the above protocols; there is only an implicit authorization by asking the bank for an authorization (by sending ARQC) or clearing of the payment (by sending TC).
3. *Payment guarantee for merchant*. In both protocols the merchant does not obtain a sufficient guarantee for the payment which is desirable before delivering goods. The merchant neither receives an authorization of the transaction by the bank nor by the customer because it cannot verify any of TC, ARQC, or IAD.
4. *Authentication and certification of merchant to customer*. EMV does not provide mechanisms to authenticate the terminal and to certify the merchant. The merchant's terminal identifier TermID is included in both ARQC and IAD such that the customer does have a guarantee that only a merchant with the TermID in the ARQC can claim the payment. However, the customer has no way of linking the TermID to the merchant s/he thinks the payment is made to, such that merchant impersonation attacks cannot be excluded.
5. *Payment receipt for customer*. The evaluation of protocols with regard to this requirement depends on the definition of valid payments and the contract between the customer and the issuer (see A3 and A4). In the offline scenario the customer does not get any authenticated proof of the payment. In the online case, one might consider the IAD as a payment receipt from the bank. This assumes, in turn, that the bank's authorization response completes the payment and that the merchant can consider the ARQC together with the IAD as a guarantee for payment. This is unlikely since the merchant can neither verify the ARQC nor the IAD.

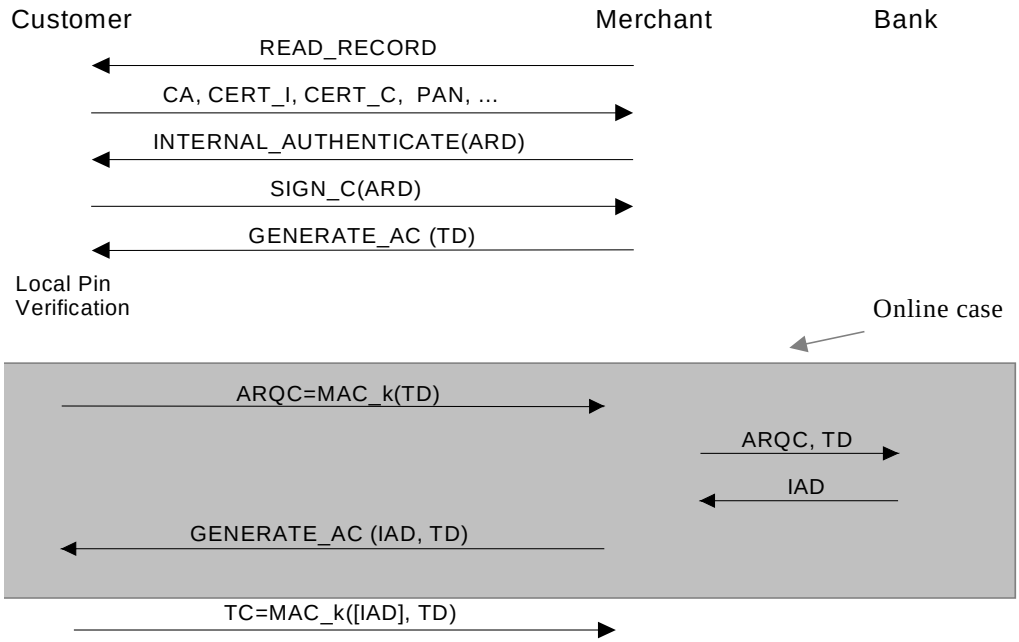

Figure 5. On-and offline EMV Internet Scenario

	Online	Offline
Part I. : GENERAL		
BANK		
1. authorization customer to bank	Y (weak)	Y (weak)
2. authorization merchant to bank	N	N
MERCHANT		
3. payment guarantee for merchant		
• authorization bank to merchant	N	N
• authorization customer to merchant	N	N
CUSTOMER		
4. merchant authentication + certification	N	N
5. payment receipt for customer		
• from the merchant	N	N
• from the bank	N	N
ALL PARTIES		
6. atomicity of payments	Y	Y
7. privacy, anonymity	N	N
Part II. SMARTCARD-SPECIFIC		
8. cardholder authorization	Y (if card-enforced VERIFY)	Y (if card-enforced VERIFY)

Table 1. Security Analysis of Online and Offline EMV Internet Scenarios
(Y = Requirement Satisfied; N= Requirement Not Satisfied)

6. *Atomicity of payments*. Atomicity of payments is provided in both protocols based on assumption A2 that money transfers between accounts are traceable.
7. *Privacy, anonymity* are not supported: EMV does not encrypt transaction data on the card-terminal channel, and the constant customer identification is present in the data read from the card. Privacy and anonymity issues are not further analyzed in the following.
8. *Cardholder authorization*. Based on assumption A5, this is achieved when using card-enforced PIN verification as recommended earlier in this section. Note that local PIN Verification in Figure 5 is shown to occur just before the card generates the ARQC; however it may be performed in an earlier stage (as long as the user is made aware of and agrees with the transaction data at the moment s/he enters the PIN).

The results of the analysis in Table 1 show a majority of unsatisfied requirements (N) without a distinction between offline and online scenarios. This is a result of the EMV specifications being developed for the POS scenario where the EMV terminal is under some control by the merchant (sometimes also bank), the purchase is performed face-to-face and merchant and bank communicate over secure connections. We shortly list and illustrate these assumptions.

1. A (physically) immediate and secure channel is assumed between card and terminal:
   - No tools for secure messaging between card and terminal are provided;
   - The result of PIN verification cannot be authenticated by the terminal;
   - Terminal applications rely upon correctness and integrity of other data returned by the card (e.g., static data authentication, risk management data).
2. A secure channel is assumed between terminal and bank:
   - no tools for secure messaging between terminal and bank are provided (e.g., online authorization messages are not authenticated between them).
3. The merchant is assumed to trust the bank:
   - the terminal cannot verify Application Cryptograms (ARQC, TC) or IAD.
4. The bank is assumed to trust the terminal to deliver messages to the card:
   - some security mechanisms rely upon the delivery of issuer messages to the card by the terminal (e.g. command scripts in the authorization response).
5. The terminal is assumed not to be counterfeitable and not to be illegally manipulatable:
   - there is no explicit mechanism to authenticate the terminal to the card;
   - the terminal does not explicitly authorize/sign any part of the transaction;
   - the card does not obtain a payment receipt from the terminal.
6. The purchase is assumed to be performed face-to-face:
   - merchant authentication is not in the scope of EMV;
   - a guarantee for the delivery of goods is out of the scope of EMV;
   - a description of goods is not part of the transaction data.
7. It is assumed that the physical presence of the same card is verifiable during the transaction:
   - different parts of the transaction protocol are not explicitly linked;
   - there is no mechanism for the terminal to verify that the same card is used for different parts of the protocol (e.g. DDA, ARQC generation, TC generation).

Before discussing mechanisms which can increase the security of using EMV over the Internet, we summarize the most important vulnerabilities resulting from the above "N"s in the table. This is done to further illustrate possible risks and threats, and to point out their relative importance.

4.1. No payment guarantee for the merchant

This is probably the most serious problem: without a payment guarantee the merchant may lose money when delivering goods which are not paid for afterwards.

- *No bank to merchant authorization*: Since the merchant cannot verify the bank's authorization response (IAD), an attacker could impersonate the bank to the merchant with an invalid IAD, convincing the merchant the transaction was successful; alternatively, valid transaction data or a valid IAD can be modified during the transaction without the merchant being aware; or, the bank might repudiate the authorization afterwards.
- *No customer to merchant authorization*: This is especially critical in the offline case because the

merchant has to accept a payment without being able to verify the TC. Anyone can make a payment on the cardholder's behalf (though DDA would at least require the fraudster to have the card) or the cardholder can repudiate a payment s/he actually made. Even if a valid TC was issued by the card, it can be modified on the way to the merchant .

4.2. No merchant authorization

The absence of an explicit authorization of the transaction by the merchant means that an attacker may impersonate a real merchant to both customer and bank, and conduct a successful transaction on behalf of the real merchant who might not even be aware. This vulnerability can be exploited also by dishonest customers and merchants: a merchant can repudiate a transaction afterwards, claiming the above attack scenario has occurred; a dishonest customer may exploit the lack of merchant authorization by intercepting and modifying the transaction data on the merchant to bank channel. In the attack scenario as well as the dishonest merchant scenario, the customer does not get the ordered goods and has to claim refund while in the last scenario the merchant does not receive the expected payment for possibly delivered goods.

4.3. No merchant-to-customer authentication and certification

For debit or credit payments the danger for the customer caused by lack of merchant authentication is limited: the customer can only lose money to a legitimate merchant if we assume that the bank only clears payments for legitimate merchants. The absence of a merchant-to-customer (M-C) authentication mostly reinforces the danger caused by the absence of a merchant-to-bank (M-B) authorization, in the sense that a fully complete, normal and legitimate payment to M can take place without M being involved in any stage of the EMV protocol. On the contrary, a reasonable protection can be achieved if at least one of the two, M-C authentication or M-B authorization, is provided. Then either the customer or the bank can verify whether M is the merchant corresponding to the TermID in the ARQC or TC. If there is M-B authorization (at least during clearing), but no M-C authentication then it only remains critical that the customer might communicate with a different merchant than intended.

4.4. No receipt for the customer

Not receiving a payment receipt is mainly critical for the customer if s/he buys goods to conditions which change rapidly (e.g. stocks or shares). Especially in the offline protocol, the customer does not have any proof of having bought something to specified conditions before the actual clearing is performed and s/he has received his or her statement of account. This can cause a loss of goods, opportunities, or money to the customer if the merchant denies certain conditions.

Note that within EMV it is impossible to simultaneously provide the merchant with a payment guarantee and the customer with a receipt because one message always has to be sent last. A simultaneous payment guarantee and customer receipt could be provided if the protocol were embedded in some *fair-exchange protocol* (such as [2]), which is out of the scope of EMV.

5. Mechanisms to Add Security when Using EMV over the Internet

We now discuss the benefits of different mechanisms which can secure EMV when used over the Internet. The protocol vulnerabilities of 'bare' EMV over the Internet relate to the absence of authorization of certain messages, and to the absence of authentication and certification of the merchant to the customer. We first analyze the merits of using a transport-layer mechanism such as SSL to secure the communication channels used, a solution which doesn't impose any changes on the EMV infrastructure. Given the limited improvements achieved with this approach, we then recommend some modifications to the EMV infrastructure that might allow a more secure use of EMV for Internet payments.

5.1. Securing communication channels

To the extent that SSL can provide initial authentication between communicating parties and integrity and/or confidentiality protection of the ensuing dialogue, it can provide a reasonable degree of protection against attacks by outsiders, under the condition that all parties involved adequately secure their systems. However, as discussed in the following paragraphs, SSL cannot provide the necessary authorizations we discussed before that are needed to protect parties against dishonest insiders.

SSL cannot authenticate individual EMV messages, rather it integrity-protects a data stream, which in addition could carry data generated by applications other than EMV. It secures the data using a shared session key which is temporary and cannot be tied to a

specific party, except by its communication partner, and then only during the existence of the connection. Obviously, SSL 'authenticated' messages or data streams can never have any authenticating value to a third party, regardless of trust assumptions of this third party. (One could of course argue whether this is the case for EMV ARQCs or TCs. But given the assumed tamperproofness of the cards, and possibly certified security of a bank's systems, EMV ARQCs or TCs may be considered by a third party as non-repudiable evidence.)

The authorizing value they have to the receiving party during the connection depends entirely on the receiver's trust in the sender's system and the sender's honesty. In a model where banks and merchants trust each other, this may suffice to add a weak authorization value to EMV messages exchanged between them; less clear is the authorization value for messages exchanged between customer and merchant. Specifically, in the offline scenario, it cannot provide a customer-authorized payment guarantee for the merchant.

Summarizing, we can say that SSL, under certain conditions, can add reasonable security against outsider attacks, but does not provide the authorization of EMV messages necessary to protect against dishonest insiders (or against honest insiders using insecure systems). In the next subsections, we suggest two modifications to EMV which can help towards solving these problems.

5.2. Signed authorization response

In the online scenario, an undeniable payment guarantee for the merchant may be provided by the (issuing) bank signing the authorization response message with its private signature key. The authorization response message becomes

SIGN_I (Y/N, Transaction Data, IAD)

where SIGN_I() stands for a signature with message recovery using the (issuer) bank's private signature key. This message can then be verified by the merchant (who already has obtained the issuer's public key during DDA) and can be submitted to the issuer again for final clearing.

The advantages of this extension are:

- This signature provides the merchant with an undeniable payment guarantee. Lack of a payment guarantee for the merchant was a major vulnerability in the above 'bare EMV' protocols (see 4.1).
- The signature prevents the Transaction Data (TD) from being modified during a protocol run without the merchant noticing it. Since the TD is included in the signature the merchant can refuse to deliver the goods if the TD is not correct. This simultaneously weakens the threats incurred by a missing authorization of the merchant to the bank (see 4.2).
- Since the merchant now has a payment guarantee before passing the IAD to the customer, the second GENERATE_AC command may now be considered as a payment receipt for the customer – assuming at least that the customer gets the IAD from the merchant (and not directly from the bank!) (see 4.4).
- No further keys have to be distributed in addition to the ones already needed for DDA.
- The extension is possible with current cards which support DDA and therefore have stored the issuer's certificate CERT_I. Only slight modifications of the terminal specifications are required to accommodate the increased length of the data fields of the authorization response message.

A disadvantage of the approach as described above is that the issuer's private key – intended only to certify card public keys - is used more often and for other purposes, increasing its exposure. This is critical because the corresponding public key is stored on many cards and therefore hard to replace in case of a compromise. Therefore we recommend to use a separate key for signing authorization responses. Using a second issuer public key (and certificate CERT2_I) for this purpose is quite costly since it has either to be stored on (and read from) the card, or sent by the issuer to the merchant as part of the authorization response. A solution which combines security and low overhead can be provided by the acquirer signing the authorization response (as opposed to the issuer). Since the merchant has a long-term relationship with the acquirer, it can be assumed that the acquirer's public key is stored permanently by the merchant.

5.3. Public-key Transaction Certificate (TC)

Another proposed change to the EMV specifications is the use of a public-key signature also for TC generation. The TC becomes

TC = SIGN_C (TD, [IAD])

signed using the card's private key. A public key-based TC is verifiable by the merchant and can be considered as a payment guarantee depending on contract terms between merchant and acquirer (which

may require certain risk management measures by the merchant). A public-key signed TC seems to be a natural extension given that DDA-capable cards already have the signature generation capability. However, in order to support this extension, message formats for cryptogram generation need to be changed, which may have a major impact on the whole EMV infrastructure and poses challenges related to backward compatibility.

Security gains that can be achieved by using a public-key based TC are:

- The merchant receives an undeniable authorization of the transaction by the customer and thus possibly (depending on contracts) a payment guarantee (see 4.1) also without online authorization;
- The transaction data cannot be modified without the merchant noticing it. This denies some of the threats incurred by a missing merchant-to-bank authorization (see 4.2).
- The TC can now also be considered an undeniable authorization customer-to-bank, as opposed to the weak authorization using the shared-key mechanism (see R1).

Despite the necessary changes to support a public key TC, we strongly recommend this extension to EMV since it is absolutely crucial to provide security in the offline case and therefore is a must when considering the use of EMV as a purse (e-cash) payment system in the Internet.

5.4. Merchant authentication

Changes proposed in 5.2 (online payments only) or 5.3 (especially important for offline payments) can greatly improve the security of the respective EMV-Internet scenarios. Vulnerabilities remain, primarily related to the lack of authentication and authorization of the merchant to both customer and bank. Closing these holes in a rigorous way by providing merchant authentication in EMV largely impacts the EMV infrastructure which currently does not allow for the storage of secret keys in merchant terminals. However, to the extent that the keys stored need not be system-wide symmetric keys but rather the merchant's own private signature key for authentication to bank and/or customer, we believe that such a modification can only strengthen overall security. Such a change first of all allows the merchant to sign the authorization request message, providing secure authorization by the requesting merchant. It also allows merchants to authenticate to the card and to deliver a signed payment receipt to the customer which in the offline case is the only means for the customer to get a receipt (other than an after-the-fact account statement). Since cards with signature verification capability are not likely to be used soon, the signature verification could be done in the trusted card reader (or eventually, in the PC software).

6. Related Work

The principle of using existing payment smart cards to secure Internet transactions has been applied in recent projects such as the e-COMM [4] and C-SET [6] projects in France. Both integrate shared-key based Transaction Certificates from existing EMV-like banking cards within SET or SET-like protocols. In this paper, rather than proposing a specific solution, we have tried to give a comprehensive and systematic overview of the security features and limits of a variety of related solutions, and hope it can be applied in the evaluation or design of similar systems.

7. Conclusion

The use of EMV 'as is' over the Internet has major (and unacceptable) security shortcomings. Securing the communication channels between the different parties (customer, merchant, bank) using secure communication protocols can prevent mainly outsider attacks. However it does not solve the inherent lack of authentication in the EMV protocol. Therefore we propose a number of EMV extensions which can increase security in the Internet setting.

The most challenging is the EMV offline scenario, where only the use of a public-key based Transaction Certificate provides appropriate security to the merchant. This scenario is particularly important if EMV'96 is used for purse (e-cash) applications.

Online EMV authorization in an Internet setting, though currently insecure because of merchant as well as bank impersonation attacks, can be made more secure by digitally signing authorization requests and responses. Lack of initial authentication and certification of the merchant to the customer is a vulnerability only to be solved by extending the EMV infrastructure with terminal-to-card (alternatively, terminal-to-reader or terminal-to-user's PC) dynamic authentication. In the absence of terminal authentication, software-based mechanisms (e.g. SSL server-to-client authentication) can be put in place to thwart the biggest risks of outsider attacks.

8. Acknowledgment

The authors thank Michael Waidner for his helpful comments and suggestions.

9. References

[1] N. Asokan, P. Janson, M. Steiner and M. Waidner, “The State of the Art in Electronic Payment Systems”, in *IEEE Computer*, September 1997.

[2] N. Asokan, V. Shoup and M. Waidner, “Asynchronous Protocols for Optimistic Fair Exchange”, *in 1998 IEEE Symposium on Research in Security and Privacy*, Oakland, May 1998, pages 86-99.

[3] M. Bellare, J.A. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, M. Waidner, “iKP - A Family of Secure Electronic Payment Protocols”, in *First USENIX Workshop on Electronic Commerce*, July 1995, pages 89-106.

[4] E-Comm, “The e-COMM Solution”, 1998. http://www.e-comm.fr/anglais/solution.html.

[5] T. Elgamal and K. Hickman, "The SSL Protocol (version 3)", Netscape Communications, Internet Draft, June 1995.

[6] Europay, “Secure Trading over the Internet thanks to C-SET.” http://www.europayfrance.fr/us/commerce/secure.html.

[7] Europay, Mastercard, Visa, “EMV’96 Integrated Circuit Card Specification for Payment Systems, Integrated Circuit Card Terminal Specifications for Payment Systems and Integrated Circuit Card Application Specification for Payment Systems”, Version 3.1.1, May 1998.

[8] Mastercard and Visa, “SET Secure Electronic Transactions Protocol, version 1.0. Book One: Business Specifications, Book Two: Technical Specification, Book Three: Formal Protocol Definition”, May 1997. Available from http://www.mastercard.com/set/#down.

[9] B. Pfitzmann and M. Waidner, “Properties of payment systems - general definition sketch and classification”, Research Report RZ 2823, IBM Research, May 1996.