2000

Using EMV Smartcards for Internet Payments

Els Van Herreweghen
IBM Zurich Research Laboratory, evh@zurich.ibm.com

Uta Wille
Jelmoli Information Systems, AG, wille_u@jelmoli.ch

Using EMV Smartcards for Internet Payments

Els Van Herreweghen
IBM Zurich Research Laboratory, 8803 Rüschlikon, Switzerland
Email: evh@zurich.ibm.com

Uta Wille*
Jelmoli Information Systems AG, Zurich, Switzerland
Email: wille_u@jelmoli.ch

***Abstract*—Existing smartcards developed for point-of-sale payments are being considered for use in Internet transactions. Such use provides an alternative to designing new smartcard solutions supporting protocols more specifically designed for Internet payments, such as SET ([9]). In this paper, we analyze EMV'96 [7], a representative example of an existing payment smartcard specification. We investigate the security of possible Internet payment systems based on EMV, and suggest modifications that can enhance the security of an Internet payment scheme based on EMV.**

I. Introduction

With the growth of electronic commerce, much effort has been put into developing secure Internet payment systems and protocols. A prominent example is SET (Secure Electronic Transaction, [9]). The original SET specifications do not target smartcard support, and SET implementations typically restrict the user to making payments from a dedicated personal computer. The lack of portability of Internet-specific systems such as SET has caused the payment industry to explore the use of 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.

This paper discusses security issues related to using EMV cards for debit and credit Internet payments. In Section II, we formulate security requirements for general smartcard-based debit and credit payments over the Internet. After summarizing the EMV'96 security mechanisms in Section III, we analyze in Section IV the security properties of using EMV 'as is' for Internet payments, by checking the resulting protocols against the formulated requirements. As the Internet scenario differs from the scenario assumed by EMV'96, these protocols show a number of vulnerabilities. In Section V, we propose mechanisms to increase the security of using EMV in the Internet scenario. Section VI, finally, discusses related work.

II. 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 exchanging money for goods or receipts as well as of at least one financial institution linking electronic payments to the transfer of "real money" [1].

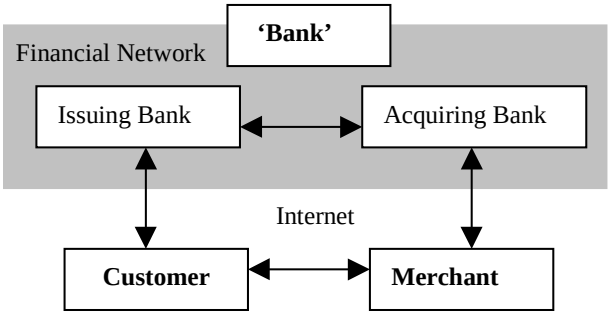

Fig. 1. Internet payment model.

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

During a transaction, actual connectivity may be limited to subsets of players. In a typical online purchase scenario, the customer has a connection only 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 the acquiring and issuing banks). The communication model, however, does not influence the security requirements R1 to R8 stated below.

Before formulating these security requirements, 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 describe only one set of "bank" requirements.

---

* This work was done while at the IBM Zurich Research Laboratory, Rüschlikon, Switzerland.

This paper is an updated version of [12].

A2. Contracts between banks, customer and merchant ensure that money transfers between accounts are traceable. This gives user and merchant 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. It especially defines valid payments by specifying the requirements to be fulfilled so as 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. It 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. If the user's payment instrument is a smartcard authorizing 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 being displayed;
- secret data such as a PIN code entered by the user is not exposed or intercepted.

A5, admittedly, is very difficult to realize. Without this assumption, however, secure user authorization can never be achieved. Its inclusion as an assumption enables us to reason about protocol requirements needed for secure authorization. We should keep in mind, however, that without the trusted card readers needed to realize A5, no smartcard-based payment system can claim non-repudiable or fully secure user authorization.

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 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 to falsely authorize another transaction on behalf of the customer. The latter applies to schemes in which secret authorization data (e.g., a PIN) is sent to the bank. In such cases, this requirement translates into the requirement that this data be confidentiality-protected (encrypted) during transfer from card to bank.

Furthermore, as in [2] and [3], we distinguish between *weak* and *undeniable* proofs of authorization. A weak proof (such as a shared-key-based EMV Application Cryptogram, Sect. III.C) cannot serve as a proof for third parties whereas an undeniable proof (based on a digital signature) provides nonrepudiation and therefore can be used in case of a dispute. Based on these notions we formulate the following security requirements for a payment protocol:

R1. **Authorization customer to bank**. The bank possesses 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.** This is achieved by either
   i. *authorization of the transaction by the bank*, or
   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 authenticated and certified critical information about the merchant.

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. *an explicit payment receipt from the merchant* or
   ii. *a payment receipt from the bank*.
   It is sometimes assumed that a receipt can be replaced by a statement of account [2,3].

R6. **Atomicity.** No party benefits from an interrupted protocol run.

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

R8. **Cardholder authorization**. The customer's payment system is protected against unauthorized use. In the case of smartcard payments, unauthorized use of the card is prevented (e.g., through use of a PIN).

The above requirements are illustrated in Section IV, where we discuss vulnerabilities that result if some of the requirements are not met.

III. Security Mechanisms Provided by EMV

This section gives an overview of the EMV'96 mechanisms for 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 the reader to [7].

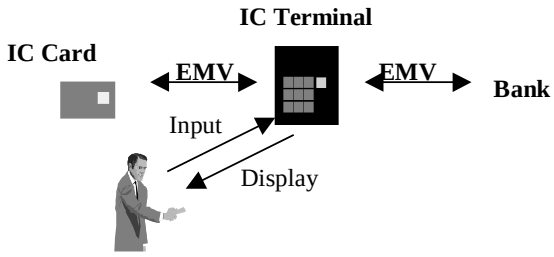

Fig. 2. The EMV POS (Point-Of-Sale) scenario.

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 do not distinguish between the issuing bank and the merchant's acquiring bank.)

- 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. The full set of security mechanisms, as shown in Figure 3, is taken from a transaction flow example in [7]. We do not discuss options and variants but focus on the maximum security features achievable in an EMV-compliant transaction.

A. Public-key-based Authentication of IC Card to IC Terminal

The first four messages exchanged implement Dynamic Data Authentication (DDA) of the card to the terminal using a public-key-based challenge-response protocol. The READ_RECORD command returns the necessary Certification (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. CERT_I, in its turn, is certified using the CA's public key known to the terminal. The INTERNAL_AUTHENTICATE command triggers the actual card authentication; the card responds with a signature over the authentication-related data (ARD).

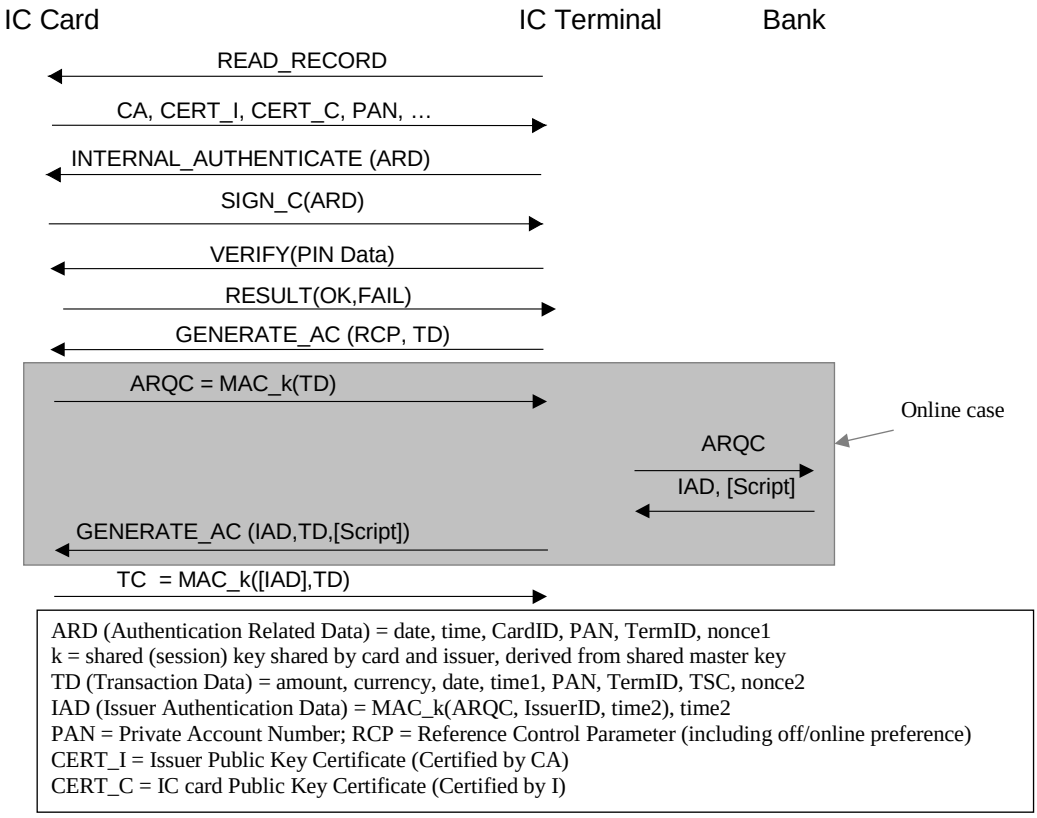

Fig. 3. Model EMV transaction flows.

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

B. Cardholder Verification

EMV supports online (PIN is sent to and verified by the bank) and offline (PIN is verified by the card) PIN verification; the exact method supported by the card is returned in the READ_RECORD response. 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.

C. 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 their 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.

IV. 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 in Figure 4 shows a customer using an EMV card for online purchases from a personal computer equipped with a card acceptance device (reader). The merchant still acts as the EMV terminal, issuing and receiving EMV commands and responses, but communicates with customer and bank over the Internet.

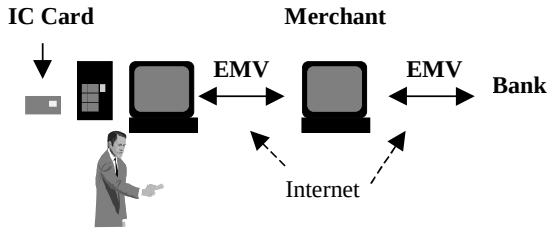

Fig. 4. The EMV Internet scenario.

PIN verification deserves some special attention. While in the POS scenario the terminal secures the transaction by making sure the PIN is verified correctly, PIN verification in an Internet setting 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 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. Firstly, requiring VERIFY (including the PIN) to be issued by the merchant assumes that the PIN first be sent to the merchant over an Internet connection (and thus unnecessarily expose it). Secondly, the result of VERIFY is not authenticated. Thus, when received over the Internet, there is no guarantee for the merchant that this result was produced by the card.

Thus, for the Internet scenario, we recommend (and assume in the following discussion) that

- only the offline PIN authentication mechanism (VERIFY by the card) be used;
- the VERIFY command be issued locally (at the cardholder terminal), and
- the card application itself enforce cardholder verification by issuing ARQC/TC only after a successful VERIFY (this is currently not an explicit condition in the EMV specifications).

We now map the online and offline transaction flows of Figure 3 to the Internet setting of Figure 4, resulting in two EMV Internet scenarios (with and without online authentication) as shown in Figure 5. In the following paragraph we analyze their security by checking them against the requirements in Section II. Unless otherwise mentioned, the discussion of a requirement is valid for both scenarios. Table 1 summarizes the results.

1. *Authorization customer to bank.* The transaction is weakly authorized by shared-key-based cryptograms (ARQC or TC).

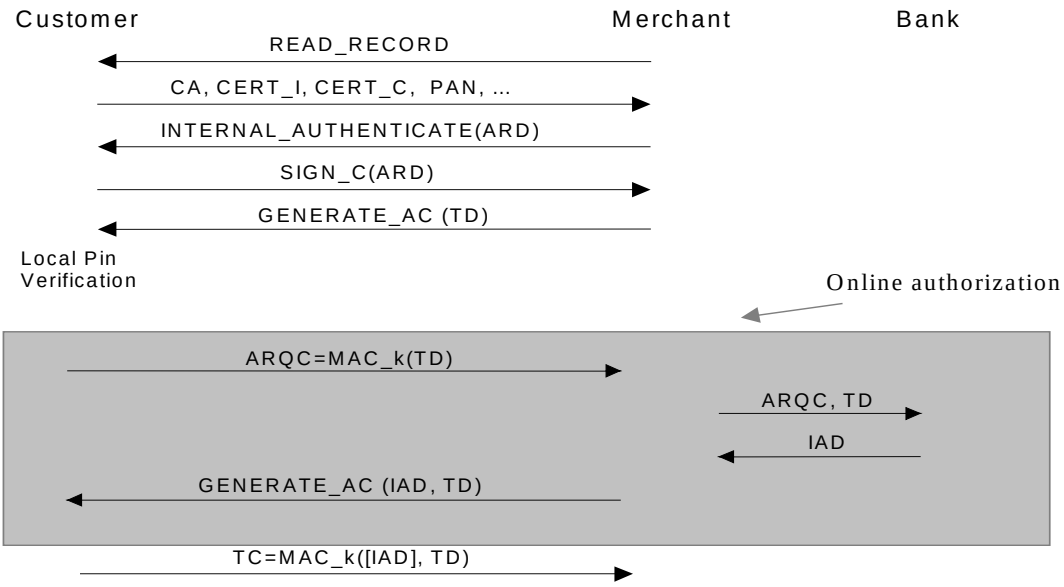

Fig. 5. EMV Internet scenarios with and without online authorization.

2. *Authorization merchant to bank*. The merchant does not explicitly authorize the transaction.
3. *Payment guarantee for merchant*. The merchant receives no authorization of the transaction by either the bank or by the customer because the merchant cannot verify any of TC, ARQC, or IAD.
4. *Authentication and certification of merchant to customer*. EMV provides no mechanisms to authenticate the terminal and certify the merchant.
5. *Payment receipt for customer*. In the scenario without online authorization, the customer receives no proof of payment. In the other scenario, one might consider the IAD as a payment receipt (see A3 and A4). This, again, assumes that the bank's response completes the payment and that the merchant can consider the ARQC together with the IAD as a guarantee for payment. This is unlikely because the merchant can verify neither ARQC nor IAD.
6. *Atomicity of payments*. Atomicity is provided, based on assumption A2 that money transfers between accounts are traceable.
7. *Privacy, anonymity* are not supported: EMV does not encrypt transaction data (such as customer identification) on the card-to-terminal channel. Privacy and anonymity will not be further discussed.
8. *Cardholder authorization*. Based on assumption A5, this is achieved when using card-enforced PIN verification as recommended above (see also Figure 5).

The summary in Table 1 shows a majority of unsatisfied requirements (N) without distinction between scenarios. This is a result of the EMV specifications being developed for the POS scenario, in which the terminal is under some control by the merchant (and/or bank), the merchant can verify the physical presence of the card, and merchant and bank communicate over secure connections. A more detailed analysis of the POS assumptions and their influence on the EMV design can be found in [12].

Before discussing mechanisms that increase the security of EMV-Internet scenarios, we illustrate some threats resulting from the "N"s in the table.

A. No Payment Guarantee for the Merchant

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

- *No bank-to-merchant authorization*: As 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 that the transaction was successful; alternatively, valid transaction data or a valid IAD can be modified during the transaction without the merchant becoming aware of it, 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 (although Dynamic Data Authentication would at least require the fraudster to have the card) or the cardholder can repudiate a payment he or she actually made. Even if a valid TC was issued by the card, it can be modified on the way to the merchant.

TABLE 1.
SECURITY ANALYSIS EMV INTERNET SCENARIOS WITH AND WITHOUT ONLINE AUTHORIZATION
(Y = REQUIREMENT SATISFIED; N= REQUIREMENT NOT SATISFIED)

	Online authorization	Without online authorization
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 (card-enforced VERIFY)	Y (card-enforced VERIFY)

B. No Merchant Authorization

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 of it, or a dishonest customer may intercept and modify the transaction data on the merchant-to-bank channel. In the former scenario, the customer does not receive the goods ordered and has to claim a refund, while in the latter the merchant does not receive the expected payment for goods possibly delivered.

C. No Merchant-to-customer Authentication/Certification

For debit or credit payments the damage for the customer caused by lack of merchant authentication is limited: the customer can only lose money to a legitimate merchant. The absence of a merchant-to-customer (M-C) authentication mostly reinforces the danger posed 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.

D. No Receipt for the Customer

This is critical mainly if the customer buys goods at rapidly changing conditions (such as shares). It can cause a loss of goods, opportunities, or money for the customer if the merchant denies certain conditions.

V. MECHANISMS TO ADD SECURITY WHEN USING EMV OVER THE INTERNET

Let us now discuss various mechanisms that can add security to the above scenarios. We first analyze the merits of using a transport-layer mechanism such as SSL ([5]) to secure the communication channels used, a solution that imposes no changes on the EMV infrastructure. Given the limited improvements using this approach achieves, we then recommend some modifications to the EMV infrastructure.

A. Securing Communication Channels

SSL can provide authentication of communicating parties, and integrity and/or confidentiality of the ensuing dialogue. If the parties involved adequately secure their systems, this provides protection against outsider attacks. However, as will be discussed in the following paragraphs, SSL cannot protect against dishonest insiders.

SSL secures a data stream rather than authenticating individual messages. This data stream could carry data generated by applications other than EMV, and is secured using a shared temporary session key that is meaningful only to both communicating parties. Thus, SSL 'authenticated' messages or data streams can never have any authenticating value to a third party. 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 honesty. Thus, SSL may add a weak authoriza-

tion value to EMV messages exchanged between a bank and a merchant who trust each other; the same can probably not be said about messages exchanged between customer and merchant.

SSL, then, can add some protection from outsider attacks, but does not provide the authorization of EMV messages necessary to protect against dishonest insiders (or honest insiders using insecure systems). In the next subsections, we suggest two modifications to EMV that help solve these problems.

B. Signed Authorization Response

In the online authorization scenario, the (issuing) bank could sign the authorization response with its private signature key:

SIGN_I (Y/N, Transaction Data, IAD)

The merchant can verify this signature and now has an undeniable payment guarantee, which solves the main vulnerabilities shown in Sect. IV.A. The merchant can also detect modifications of Transaction Data (TD), which weakens the threats incurred by a missing authorization of the merchant to the bank (see Sect. IV.B).

If the same issuer key is used for signing the authorization response as for certifying cards, this extension is very straightforward to implement, because the certificate CERT_I is already present on the card. However, re-using this issuer certification key to sign transaction messages considerably increases its exposure. This is critical because the public key is stored on many cards and therefore difficult to replace if compromised.

Using a different issuer public key (and certificate) is quite costly because it either has to be stored on the card or sent by the issuer to the merchant as part of the authorization response. An alternative solution, combining security and low overhead, consists of the acquirer (as opposed to the issuer) signing the authorization response. As 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.

C. Transaction Certificate (TC)

The Transaction Certificate TC could be signed with the card's private key:

TC = SIGN_C (TD, [IAD])

A public-key-based TC is verifiable by the merchant and can be considered a payment guarantee (Sect. IV.A), depending on the contract terms between merchant and acquirer. Also, the merchant can detect a modification of transaction data, precluding some of the threats related to a missing merchant-to-bank authorization (Sect. IV.B).

A signed TC seems to be a natural extension to EMV, given that DDA-capable cards already have signature capability. However, to support this extension, message formats for cryptogram generation need to be changed, which may have a major impact on the entire EMV infrastructure. Despite these necessary changes, we recommend this extension because without it, it is difficult to provide security in the absence of online authorization.

D. Merchant Authentication

The changes proposed in Sects. V.B (online authorization only) or V.C (especially important in the absence of online authorization) can greatly improve the security of the corresponding EMV-Internet scenarios. The remaining vulnerabilities are 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 secret keys to be stored in merchant terminals. However, 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. Therefore such a modification can only improve overall security. It 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 for the customer that, without online authorization, is the only means for the customer to receive a receipt (other than an after-the-fact account statement). Alternatively, signature verification could be done in the trusted card reader (or, possibly, in the PC software).

VI. Related Work

The principle of using existing payment smartcards to secure Internet transactions was applied in pilot projects such as e-COMM [4] and C-SET [6] in France. Both integrate shared-key-based Transaction Certificates from existing EMV-like banking cards within SET or SET-like protocols.

In line with these pioneering efforts, recent enhancements to the EMV'96 and SET specifications allow integration of the two payment systems. In the following, we discuss these enhancements, and relate them to the analysis made in this paper.

The EMV'96 Chip Electronic Commerce Specification [8] describes how to integrate key EMV applications, such as online card authentication and cardholder verification, into SET. According to [8], SET provides confidentiality, integrity, interoperability, and merchant authentication, while EMV provides card authentication and cardholder verification. This is done by including EMV cryptograms, and possibly EMV PINs, into the SET payment messages. The SET extensions necessary to accommodate this inte-

gration are described in the SET Common Chip extension [10] and SET Online PIN Extensions [11].

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. Similarly, the EMV and SET extensions mentioned above describe not just one but a large set of possible combined EMV-SET solutions, depending on the EMV smartcard application and the version of SET used. As such, we cannot analyze the EMV and SET extensions against the security requirements in this paper. We hope, however, that the analysis and examples provided in this paper can help design specific solutions based on the EMV and SET specifications.

VII. 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 that can increase security in the Internet setting.

The most challenging is the EMV scenario without online authorization, where only the use of a public-key-based Transaction Certificate provides appropriate security to the merchant.

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 that can only be solved by extending the EMV infrastructure with terminal-to-card (or, 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 limit the risk of outsider attacks.

Acknowledgments

We thank Michael Waidner and the anonymous referees for their helpful comments and suggestions.

References

[1] N. Asokan, P. Janson, M. Steiner and M. Waidner, "The State of the Art in Electronic Payment Systems," in *IEEE Computer* 30(9) 28-35, Sept. 1997.

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

[3] M. Bellare, J. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, E. Van Herreweghen, and M. Waidner, "Design, implementation and deployment of the iKP secure electronic payment system," IEEE J. Sel. Areas in Commun. 18(4), April 2000 issue, in press.

[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] Banksys and Groupement de Cartes Bancaires, "Interoperable C-SET: Overview, Business Description and Protocol." Available from http://www.banksys.be/eng/index3p5.html.

[7] Europay, Mastercard, Visa, "EMV'96 Integrated Circuit Card Specification for Payment Systems, Integrated Circuit Card Terminal Specification for Payment Systems and Integrated Circuit Card Application Specification for Payment Systems," Version 3.1.1, May 1998. Now available from http://www.emvco.com/specifications.cfm

[8] Europay, Mastercard, Visa, "EMV'96 Chip Electronic Commerce," Version 1.0, Dec. 1999. Available from http://www.emvco.com/specifications.cfm.

[9] 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.setco.org/download.html.

[10] SETCo, "Common Chip Extension SET™ 1.0 (prepared by Europay, Mastercard, Visa)," Sept. 1999. Available from http://www.setco.org/download.html.

[11] SETCo, "Online PIN Extensions to SET™ 1.0," May 1999. Available from http://www.setco.org/download.html.

[12] E. Van Herreweghen and U. Wille, "Risks and Potentials of Using EMV for Internet Payments," in *USENIX Workshop on Smartcard Technology*, Chicago, May 1999, pages 163-173.