Smart Cards in Hostile Environments

Howard Gobioff
Carnegie Mellon Univ.
Pittsburgh, PA 15213
`hgobioff@cs.cmu.edu`

Sean Smith
Los Alamos Nat. Lab.
Los Alamos, NM 87545
`sean@lanl.gov`

J. D. Tygar
Carnegie Mellon Univ.
Pittsburgh, PA 15213
`tygar@cs.cmu.edu`

Bennet Yee
UC San Diego
La Jolla, CA 92093
`bsy@cs.ucsd.edu`

Abstract

One often hears the claim that smart cards are the solution to a number of security problems, including those arising in point-of-sale systems. In this paper, we characterize the minimal properties necessary for the secure smart card point-of-sale transactions. Many proposed systems fail to provide these properties: problems arise from failures to provide secure communication channels between the user and the smart card while operating in a potentially hostile environment (such as a point-of-sale application.) Moreover, we discuss several types of modifications that can be made to give smart cards additional input/output capacity with a user, and describe how this additional I/O can address the hostile environment problem. We give a notation for describing the effectiveness of smart cards under various environmental assumptions. We discuss several security equivalences among different scenarios for smart cards in hostile environments. Using our notation, these equivalences include:

- private input $\approx$ private output
- trusted input + one-bit trusted output $\approx$ trusted output + one-bit trusted input
- secure input $\approx$ secure output

This research was supported in part by the Advanced Research Projects Agency under contract F119628-93-C-0193, IBM, U.S. Department of Energy under Contract No. W-7405-ENG-36, and the US Postal Service. Howard Gobioff was supported in part by a National Science Foundation Graduate Fellowship. The views and conclusions in this document are those of the authors and do not necessarily represent the official policies or endorsements of the US Government, its agencies, or any of the research sponsors.

1 Introduction

Point-of-sale (POS) systems introduce a number of security problems. In a traditional credit card model, the customer reveals his credit card number to the merchant. This allows a corrupt merchant to improperly use the customer's credit card.

To solve this problem, computer scientists have proposed the use of *smart cards* that can act as intermediate brokers. Smart cards are small handheld computational devices that can perform cryptographic operations. One type of smart card model is a *stored value card* containing an account balance register. The smart card is considered tamper-resistant, in that it is not feasible for any person to modify the smart card account balance without going through an approved protocol. Many recent smart cards provide mechanisms that will cause any attempt to physically read data in the smart card to result in all data being zeroed (e.g., US Federal Information Processing Standard 140-1 [8].)

Similarly, the merchant's POS computer will contain a tamper-resistant register representing the merchant's account balance. When a customer makes a purchase, the smart card account balance is decremented by the amount of the purchase, and the merchant's POS account balance is incremented by the same amount. Later, smart cards and POS systems report their current account balances to a computer acting as a bank, and their accounts are accordingly modified. If the registers are truly tamper-proof, this approach appears to provide a safe way to exchange values off-line. This approach (with slight modifications) is taken in the proposed *MasterCard 2000* and *Visa Stored Value Card* systems [6, 7, 9].

A different set of approaches has been proposed by digital cash researchers [3, 4, 5]. "Electronic wallets" transfer an electronic token to the point-of-sale system. At a later time, the electronic token is "cashed in", for reconciliation, to the computer acting as a bank. Digital cash approaches typically provide anonymous transactions, and use fewer assumptions about tamper-resistance. In particular, Chaum [5] divides the smart card into an *observer*, a tamper-proof device trusted by the digital cash system, and the user's *representative*, in which the observer is embedded. The user has full control over the hardware embodying the representative, but has no internal access to the observer. The observer participates in Chaum's protocol and actively prevent double spending in such a way that the user need not trust the correctness of the observer with respect to leaking identity information; the observer may, however, cause denial of service.

However, both stored value cards and electronic wallets ignore one very feasible attack: since traditional smart cards do not contain any provision for directly displaying output or directly receiving input from the customer, they must depend on the merchant's POS system for I/O with the customer (this problem was observed in [2, 10, 11]). This introduces a significant vulnerability: for example, a corrupt merchant might try to charge the customer's smart card $1000 for the purchase of a gold watch while truthfully reporting on his POS display that the purchase is for a $10 watch battery. If the customer authorizes the smart card to transfer funds based on the displayed data, the merchant successfully defrauds the customer.

Note that the systemic threat that is being addressed by this paper differs dramatically from those being addressed by the observer model in digital cash systems. Here, we are concerned with the possibility of corruption of the POS terminal, so that the information displayed to the user — as part of an authorization request — shows one price, while the smart card is shown another. This variant of the Trojan Horse attack is impossible to solve without some way for the user to learn the true transaction value as seen by the smart card. In the observer model, Chaum assumes that the representative possesses secure I/O for communication with the user, a property not true of traditional smart cards.

This paper explores a number of variations in smart card designs that address this problem. We give an informal notation to describe equivalences of various smart card mechanisms to provide protection to interactions between the user and smart card where the smart card is accessed in a potentially hostile environment. These equivalences show that mechanisms that achieve certain security properties can be simulated by alternative mechanisms.

Further, we describe some potential designs for smart cards with additional I/O channels direct to the user. For example, these hypothetical smart cards contain LEDs that display values to be directly read by the smart card owner, or contain buttons to directly input material from the smart card owner. In this paper, we describe requirements for these I/O-enhanced smart cards and consequences of their theoretical security properties. However, we do not attempt here to discuss the physical construction, economics, or feasibility of various alternatives among these I/O-enhanced smart cards.

2 Our Model

We describe interactions between the smart card and the customer by separating the description of input and output. The security properties of both input and output can be described by the presence or absence of two attributes: privacy and trust.

Privacy means that the content of a communication cannot be observed by anyone who is neither the sender nor receiver. In our context, privacy refers to a customer – smart card communication being protected from observation by the merchant. If a communications channel is not private, we say it is *public.*

Trust means that a recipient has confidence in the origin and freshness of a communication. If the customer receives a trusted communication from the smart card, then he is confident that the communication originated from the smart card in the immediate past and is being received intact (for example, a multi-part message is being received unmodified by an adversary). If the customer has a trusted input channel to the smart card, the smart card can treat communications on that channel as fresh, in proper order, and having originated from the possessor of the smart card. If a communications channel is not trusted, we say it is *untrusted.*

If a communication is both trusted and private, we say it is *secure.*

2.1 Notation

We give informal definitions of our notation; this gives us a convenient shorthand for discussing security properties.

A $\succ$ B This expression means that any protection mechanisms provided by a smart card with B can also be provided by a smart card with A. For example, "trusted input $\succ$ no input" because a smart card that has no input can be simulated by a smart card that does have trusted input.

A $\approx$ B This expression means that both A $\succ$ B and B $\succ$ A.

A + B This expression refers to a class of communication security properties provided by smart cards that have both A and B. For example, "trusted input + trusted output $\succ$ trusted input", because any smart card that has trusted input only can be simulated by a smart card that has both trusted input and trusted output.

3 Methods

This section includes arguments showing the following equivalences:

- private input $\approx$ private output
- trusted input $+$ one-bit trusted output $\approx$ trusted output $+$ one-bit trusted input
- secure input $\approx$ secure output

3.1 Achieving these Properties

The most straightforward way for a customer to establish trusted and private communication channels with a smart card is to insert the smart card into a reader/writer device trusted by the customer. These devices directly provide *trusted* input and output, and with the proper physical shields (e.g., to prevent shoulder-surfing), also provide *private*, and hence *secure*, input and output.

In POS applications, the merchant controls the reader/writer. Indeed, from the customer's point of view, the merchant's smart card reader qualifies as a potentially hostile environment. Hence, we need to consider techniques to achieve trusted and private communication with smart cards in hostile environments.

One approach is to put a keypad and display directly on the smart card. Such peripherals increase the cost of the smart card (and may violate assumptions about the smart card's physical security), but provide trusted communication for the customer; and, with physical shielding, can provide privacy as well. If keypads and displays on the smart card are infeasible, the customer could carry a trusted portable smart card reader [2]. In this sort of system, after the transaction parameters are transferred to the smart card, the customer would transfer the card from the merchant's terminal into the portable reader. Only if the customer approves the transaction would he move the card back to the merchant's terminal. However, using portable readers this way may be unacceptably awkward for many POS applications; certainly, if the customers are willing to carry portable smart card I/O devices, we might as well omit the smart card and have the trusted I/O devices communicate directly with the merchant terminal via some higher bandwidth channel than a smart card (e.g., infra-red link or a cable).

Another approach routes communications through the merchant's reader/writer, and protects those communications using information security techniques. For example, if the customer and the smart card shared knowledge of a large codebook, they could use this codebook to send messages to each other that intermediaries could neither understand nor forge. Alternatively, if the customer can perform cryptography in his head (such as digital signatures, or RSA or DES encryption) and can enter data using numeric keypads very quickly, then the customer and smart card could simply pass encrypted and/or signed messages to each other, achieving trust and, if encryption is used, privacy. However, doing operations with large codebooks from memory and performing RSA and DES encryptions in one's head appears to be beyond the ability of most normal human beings.

3.2 Equivalence

If a customer has direct secure input and output communication paths, he can safely communicate with his smart card and achieve safe POS applications. However, safe communications are possible over indirect, untrusted paths if the customer has a shared secret with the smart card, such as a one-time pad. Below we establish a set of equivalences and implications for various types of customer/smart card communication.

Private input $\succ$ private output. If the customer has an input channel to the smart card that, should it actually reach the smart card, can be presumed private, the customer can turn an untrusted public output channel into an untrusted private output channel by giving the smart card a one-time pad to encrypt its output. (While methods using one-time pads may seem a bit far-fetched, in Section 4 below, we discuss a practical example.)

Secure input $\succ$ secure output. Similarly, suppose the customer has a secure channel to the smart card. The customer can transform an untrusted public output channel into a secure channel by entering a one-time pad. Output from the smart card to the customer is encrypted with the one-time pad, including a data checksum to detect data integrity loss.

Trusted input + one bit of trusted output $\succ$ trusted output. The customer can feed the value displayed by untrusted output back to the smart card. The smart card then uses its one bit of trusted output to signal that it received the value and agrees with it.

Symmetry of input and output. In a hostile environment, a symmetry exists between the customer and the smart card. (The principal differences are that the smart card has more memory and more computational ability.) Input from the customer's point of view is equivalent to output from the smart card's point of view, and *vice versa.* Suppose a rule exists of the form

$$X_1 \text{ input } + Y_1 \text{ output } \succ X_2 \text{ input } + Y_2 \text{ output}$$

Then by this symmetry, we also have:

$$Y_1 \text{ input } + X_1 \text{ output } \succ Y_2 \text{ input } + X_2 \text{ output}$$

Applying this rule to the above equivalences, we obtain the following:

Private output $\succ$ private input. If the customer receives private output from the smart card, he can generate private input to the smart card. If the smart card presents a one-time pad to the customer through the private output, the customer can encrypt his desired input to the smart card (see Section 4 for an example).

Secure output $\succ$ secure input. Similarly, we can use a one-time pad to guarantee the privacy of our communication. With a private channel, the smart card can present the customer with an authentication challenge. The customer provides an appropriate response and subsequent communication are encrypted by the one-time pad.

Trusted output with one bit of trusted input $\succ$ trusted input. If the customer has trusted output from the smart card and one bit of trusted input, the customer can generate trusted input. (Note: if the customer can yank his smart card out of the merchant's reader/writer, then he has at least one bit of trusted input!) The customer provides his input to the smart card and the smart card echoes back the information to the customer. If the smart card echoes the wrong information, the customer uses the one bit of trusted input to inform the smart card of the communication failure. (This method uses an implicit assumption that the possessor of the smart card is an authorized user. By itself, this method by does not provide protection against smart card theft.)

4 Achieving Secure Transactions

We discuss some possible requirements for an I/O-enhanced smart cards that would give a variety of security configurations. We make no attempt to discuss the technical or economic feasibility of I/O-enhanced smart cards; this paper is concerned with exploring various equivalences of security properties among different types I/O-enhanced smart cards.

Here are minimum requirements to accomplish a POS transaction: The customer must communicate to the smart card enough information to indicate the amount of the transaction. It is also necessary that the smart card know the merchant's identity so that it can verify it (in order to protect against Trojan horse attacks by untrusted POS terminals) — the merchant identity information is important to avoid problems later with unrolling transactions, e.g., in order to return defective or otherwise unsuitable merchandise. The customer need not personally provide the requisite information to the smart card. The merchant may provide the information directly to the smart card which will then verify it with the user through trusted input and output. The smart card does not require user authentication, which is why "trusted" means to possessor of card, and with trusted I/O privacy is not required for transaction authorization. If either trusted input or output is unavailable, then, as we see below, we may require additional privacy conditions.

From the customer's perspective, the only absolute requirement is to provide proper information to the smart card. The minimal required mechanism is trusted input. As we have seen, trusted input can be implemented through a variety of combinations of input and output properties.

The merchant must be able to tell his POS system the amount of the expected transaction and know when the transaction completes. This requirement is satisfied if the merchant has trusted input to the POS system, which is trivial if the merchant controls the environment, and one bit of trusted output to indicate transaction completion.

4.1 Examples

Private output $\succ$ private input. How can we get use private output to simulate private input? For example, if a smart card has an integral display unit but no direct input capabilities, customers can privately communicate with the smart card. One way is for the smart card to present a random sequence of digits on its display unit [1]. The customer then sends a sequence of increment, decrement, and next-digit commands to the smart card via the public, unsecure communication channel. These commands alter the smart card's initial random value until the smart card displays the input value desired by the customer. This is effectively a special form of a one-time pad. This approach can be used for both entering a password and transaction amounts. (Note that this method is vulnerable to an adversary that can simultaneously shoulder-surf the display and tap the input stream.)

Private input $\succ$ private output. Conversely, how can we use private input to simulate private output? Consider a smart card containing an integral numeric keypad that but lacking human readable output capabilities. This smart card is inserted in a POS terminal providing (unsecure) output for the smart card. If the customer takes precautions to prevent observers from observing information typed into the smart card, the customer can provide the smart card with a one-time pad. The smart card could then encrypt data using the one-time pad. Since only the smart card and the customer are in possession of the one-time pad, they are the only parties able to read the message.

Trusted input with one bit of trusted output $\succ$ general trusted output. A trusted input channel, such as a numeric keypad, allows the smart card to present trusted output over an untrusted communications channel. The customer feeds the output from the untrusted path back to the smart card via the trusted input channel. If the smart card receives an unexpected value from the customer, it uses a single bit of trusted output, such as an integral LED, to signal the problem to the customer. (This method does not address the customer authentication problem for stolen smart cards.)

Trusted output with one bit of trusted input $\succ$ general trusted input. If a smart card is capable of presenting trusted output to the customer (for example, through an integral display) and the customer can reply with one bit of trusted input (such as removing the smart card from the reader) then the customer and smart card can achieve a trusted input channel. The customer communicates to the smart card via an untrusted input channel and the smart card then echoes back the input message through the trusted output channel. If the customer disagrees with the message displayed by the smart card, the customer communicates this via the single bit of trusted input.

5 Conclusion

We believe that the corrupt point-of-sale terminal problem to be a major challenge for using smart cards in electronic commerce. We have begun a discussion of potential solutions by discussing equivalences among varying types of I/O-enhanced smart cards and the types of protection they provide. We hope that the smart card community will investigate the economic and technical feasibility of providing trusted input and output mechanisms (or their equivalent) for smart cards.

We also believe that these mechanisms could also find applications outside of POS transactions. For example, consider the key management case: Imagine that a user has a portable device (such as a smart card) with a private key (for asymmetric) for electronically signing documents. How can the user make sure that his or her device only signs the document that he or she approved?

Our informal calculus of equivalences is meant to be suggestive instead of a formal reasoning method for smart card security. However, we believe that this notation could be formalized, and that the process of making it mathematically rigorous may illuminate further issues in the use of smart cards in hostile environments.

References

[1] M. Abadi, M. Burrows, C. Kaufman, and B. Lampson. Authentication and delegation with smart-cards. Technical Report 67, DEC Systems Research Center, October 1990.

[2] Jean-Paul Boly, Antoon Bosselaers, Ronald Cramer, Rolf Michelsen, Stig Mjølsnes, Frank Muller, Torben Pedersen, Birgit Pfitzmann, Peter de Rooij, Berry Schoenmakers, Matthias Schunter, Luc Vallée, and Michael Waidner. The esprit project cafe — high security digital payment systems. In Dieter Gollmann, editor, *Computer Security — ESORICS 94, Third European Symposium on Research in Computer Security*, number 875 in Lecture Notes in Computer Science, Brighton, United Kingdom, November 1994. Springer-Verlag. ISBN 3-540-58618-0.

[3] Stefan Brands. An efficient off-line electronic cash system based on the representation problem. Technical Report CS-R9323, Centrum voor Wiskunde en Informatica, 1993.

[4] E. Brickell, P. Gemmell, and D. Kravitz. Trustee-based tracing extensions to anonymous cash and the making of anonymous change. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 457–466, 1995.

[5] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, October 1985.

[6] Europay International, MasterCard, and Visa. Integrated circuit card specification for payment systems, October 1994.

[7] Mastercard launches development of smart card platform. Press Release, July 20, 1994.

[8] U. S. National Institute of Standards and Technology. Federal information processing standards publication 140-1: Security requirements for cryptographic modules, January 1994.

[9] Visa to develop and test prototype chip technology. Press Release, November 8, 1994.

[10] Bennet Yee and Doug Tygar. Secure coprocessors in electronic commerce applications. In *Proceedings of The First USENIX Workshop on Electronic Commerce*, New York, New York, July 1995.

[11] Bennet S. Yee. *Using Secure Coprocessors*. PhD thesis, Carnegie Mellon University, 1994.