# Smart Cards in Hostile Environments

Howard Gobioff Sean Smith J. D. Tygar

September 14, 1995
CMU-CS-95-188

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

**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. This paper argues that many proposed smart card systems still lack effective security for point-of-sale applications. We consider the point-of-sale terminal as a potentially hostile environment to the smart card. Moreover, we discuss several types of modifications that can be made to smart cards to improve their security and address this problem. We prove a set of equivalences among a number of these modifications:

- private input = private output
- trusted input + one-bit trusted output = trusted output + one-bit trusted input
- secure input = 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. This paper is registered as Los Alamos Unclassified Report LA-UR-95-2224. 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.

**Keywords:** POS, smart card,security

# 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-proof in that it is not feasible for any person to modify the smart card account balance without going through an approved protocol. Any attempt to physically read any data in the smart card should result in all data being zeroed (e.g., US Federal Information Processing Standard 140-1 [7].)

Similarly, the merchant's POS computer will contain a tamper-proof 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 appears to provide a safe way to exchange values off-line. This approach (with slight modifications) is taken in the new proposed *MasterCard 2000* and *Visa Stored Value Card* systems [5, 6, 8].

A different set of approaches has been proposed by digital cash researchers [2, 3, 4]. (These approaches are sometimes called *electronic wallets* or *electronic purses.*) 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-proofing.

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[9, 10]). This introduces a significant vulnerability: for example, a corrupt merchant might try to charge the customer $1000 while reporting on his POS display that the price was $10. This paper explores a number of variations in smart card design that address the problem. We present a set of theoretically equivalent methods to solve the problem, and give examples of potential implementations.

# 2 Our Model

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

*Privacy* means that the content of a communication cannot be observed by anyone who is neither the sender nor receiver. In particular, 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 in the transmitted order). 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.*

# 3 Methods

This section includes arguments showing the following equivalences:

- private input = private output
- trusted input + one-bit trusted output = trusted output + one-bit trusted input
- secure input = 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. In this sort of system, whenever a display was required the smart card would halt; the customer would transfer the smart card from the merchant's terminal into the portable reader. If the customer approves the transaction, he would move the card back to the merchant's terminal. Using portable readers this way may be unacceptably awkward for many POS applications.

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, such as digital signatures, or RSA or DES encryption, in his head, 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, and performing RSA and DES encryptions in one's head appears to be beyond the ability of 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 implies 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 the *Achieving Transactions* section below, we discuss a practical example.)

**Secure input implies secure output.** Similarly, suppose the customer has an 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.

**Trusted input with one bit of trusted output implies 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

$$X_1 \text{ input plus } Y_1 \text{ output implies } X_2 \text{ input plus } Y_2 \text{ output}$$

Then by this symmetry, we also have:

$$Y_1 \text{ input plus } X_1 \text{ output implies } Y_2 \text{ input plus } X_2 \text{ output}$$

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

**Private output implies 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.

**Secure output implies 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 implies 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 Transactions

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 this way, we protect against Trojan horse attacks by untrusted POS terminals. The customer need not personally provide this information. The merchant may give the information to the smart card, which presents the information to the customer through trusted output, who uses trusted input to approve the transaction. If trusted input and output to the smart card is possible, then privacy is not required. If either trusted input or output is unavailable, then, as we see below, we may have privacy requirements.

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 input from private output.** For example, if a smart card has an integral display unit but no direct input capabilities, customers can privately communicated with the smart card. One way (given in [1]) is for the smart card to present a random sequence of digits on its display unit. 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 output from private input.** 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 implies 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 implies 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

The corrupt point-of-sale terminal problem is a major challenge to using smart cards for electronic commerce. We argue that smart cards should include trusted input and output mechanisms, or their equivalent. We believe that these mechanisms could also find applications outside of POS transactions.

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

[3] 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.

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

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

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

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

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

[9] Bennet Yee and J. D. Tygar. Secure coprocessors in electronic commerce applications. In *First USENIX Workshop on Electronic Commerce*, July 1995. To appear.

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