

A Study on ATM Security Measures
by Command Verification

(制御コマンド真正性検証を用いた
ATM セキュリティ対策に関する研究)

A dissertation

by

Hisao Ogata

Supervisor: Prof. Dr. Tsutomu Matsumoto

Graduate School of Environment and Information Science
Yokoyama National University

March, 2020

Abstract

Recently, criminals frequently carry out logical attacks on ATMs (Automated Teller Machines) systems to steal cash from ATMs in more than 30 countries. Since the number of ATMs is increasing worldwide with global economic growth, the logical attacks on increasing ATMs become serious social issues. In general, an ATM consists of a PC and peripheral devices, such as a card reader and a dispenser. The PC and a peripheral device are connected with a USB/RS-232C cable, and the PC and the host computer are connected via the financial institution's intranet. Major attack surfaces of those logical attacks are the intranet, the PC, and the USB/RS-232C cables. Eventually, unauthorized cash dispensing commands are sent to the dispenser to fraudulently withdraw cash from the ATM without generating a transaction in those attacks. In existing measures, it is necessary to maintain the integrity of the executable files on the PC. However, these measures could be bypassed or disabled by criminals since existing ATM operations require frequent physical/logical access to the inside of ATMs. Maintaining the integrity of the executable files by tight operational management raises the increasing costs of operational management. In particular, it is difficult to maintain integrity with limited human resources 24 hours 7 days in case a financial institution operates more than ten thousand ATMs.

Therefore, there are two objectives of this study. The first objective is to provide an effective ATM security measure without imposing a heavy burden on financial institutions and ATM operations while maintaining the stability of services as social infrastructure. The second objective is to establish a general application scheme of the proposed measure that can be applied to multiple ATM systems and transactions. This dissertation is organized as follows. Chapter 2 presents the existing ATM systems and services. Chapter 3 describes logical attacks on ATMs and existing measures. Chapter 4 proposes a measure called "Command Verification", the primary model of the measure and applied system examples. Chapter 5 describes issues of existing measures and applying Command Verification to one transaction sub-process in an ATM transaction. Command Verification and existing measures are compared in terms of the practical effects of the measures in existing ATM operations. Chapter 6 details the issue and the solution for applying it to two transaction sub-processes in an ATM transaction. In detail, we propose an implementation model analysis of Command Verification to solve the issue. Chapter 7 presents the issue and the solution for applying it to all transaction sub-processes in an ATM transaction. That is, we propose

an implementation design method of Command Verification to cope with the issue. Chapter 8 concludes this dissertation.

In chapter 4, we propose a measure called “Command Verification” to solve the issues of the existing security measures and show applied system examples. The primary idea of Command Verification is that peripheral devices themselves verify commands sent from the PC. We also propose a primary model of Command Verification in order to apply it to various ATM systems and transactions. Since peripheral devices usually do not have any information to verify a command, two peripheral devices are defined in the model; an information acquiring device and a verified command executing device. The information acquiring device extracts command verification information from input data of the acquiring device and securely transfers the information to the verified command executing device. The verified command executing device verifies a command from the PC with the received information.

In chapter 5, an application of Command Verification to one transaction sub-process in an ATM transaction is described. Practical effects of Command Verification and existing measures are compared in the application, namely, the cash handling sub-process in a cash withdrawal transaction with a smart card. Three conditions to effectively prevent unauthorized cash withdrawal in existing ATM operations are derived from analysis of existing ATM systems and operations. It was shown that Command Verification can meet the three conditions while the existing measures do not meet them.

In chapter 6, an application of Command Verification to two transaction sub-processes in an ATM transaction is described. That is an application to a transaction sub-process before/after communication between an ATM and the host computer in a deposit transaction with a smart card. There are multiple protected properties from multiple attack surfaces in the transaction sub-processes, and constraints to be satisfied which are coming from existing systems and operations. Since it is difficult to achieve properly implementable systems of Command Verification to meet the requirements, the implementation model analysis is introduced to compare the features of the models in a preliminary step to derive proper systems. Two recommended implementation models were derived from the model analysis, and two types of properly implementable systems were finally derived using the recommended models. The management cost of the properly implementable system has been reduced to less than one ten-thousandth.

In chapter 7, an application of Command Verification to all transaction sub-processes in an ATM transaction is explained. Namely, application to all transaction sub-processes in a cash withdrawal transaction with a magnetic stripe card, in which there are few existing security mechanisms. When Command Verification is applied to the transactions, there are many implementable systems due to the poor existing security mechanisms. It is difficult to derive proper systems among the many implementable systems, since the proper systems should meet many conditions; preventing a wide range of logical attacks, harmonizing with existing ATM operations, and minimizing modification costs of peripheral devices. We propose a systematic implementation design method of Command Verification to derive proper systems, which consists of three steps and guidance. Three proper systems out of the 135 implementable systems were selected by applying the design method to magnetic stripe card transactions. That is, the number of candidate systems to be examined in detail was reduced to one forty-fifth.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Tsutomu Matsumoto, for his tremendous advice, insights, and supports for our research. Without his guidance and persistent help, this dissertation would not have been possible. I would like to express my gratitude to Professor Tatsunori Mori, Professor Tsutomu Matsumoto, Professor Junji Shikata, Associate Professor Katsunari Yoshioka, and Lecturer Shinichi Shirakawa for serving on my dissertation committee. Their invaluable comments and feedback were extremely helpful to improve this dissertation.

I would like to greatly thank the representative director, Dr. Tsukasa Ogino of the general incorporated association Connected Consumer Device Security Council, and Mr. Toshiya Hamasaki of Hitachi-Omron Terminal Solutions, Corp. for their aid in starting and supporting our research with Prof. Matsumoto's laboratory. With regard to the discussion and the trial production of the prototype system to demonstrate the operation of the proposed measure, I am also grateful to Ms. Yukie Taniyama, Mr. Eiji Mizuno, Mr. Masahiro Yoshii, Mr. Hiroyuki Tanoue, Mr. Yoshinaga Horii of Hitachi-Omron Terminal Solutions, Corp.

I gratefully acknowledge the work of the past and present members of the Matsumoto Laboratory. Their suggestions gave me useful advice and feedback. I also appreciate the help from the secretaries of Prof. Mtsumoto's Laboratory, Ms. Mio Narimatsu, Ms. Tomoko Ishidate, and Ms. Emiko Kawamura.

Finally, I would like to especially thank my parents for their unconditional support and constant encouragement. I would like to express my deepest gratitude to my wife, Tomoko Ogata for her unconditional support and continuous encouragement.

Table of Contents

Abstract	i
Acknowledgements.....	iv
Table of Contents	v
List of Figures	viii
List of Tables	x
Important Terms and Abbreviation	xi
Chapter 1 Introduction.....	1
1.1 Background.....	1
1.2 Contribution.....	2
1.3 Organization	3
Chapter 2 ATM System and Services	5
2.1 ATM System Structure	5
2.2 ATM Services	7
2.3 ATM Transactions	7
Chapter 3 Logical Attacks and Existing Measures.....	11
3.1 Logical attacks on ATMs and entry points.....	11
3.2 Typical Logical attacks on ATMs	13
3.2.1 Jackpotting.....	13
3.2.2 Black boxing	15
3.2.3 Man in the Middle	16
3.3 Existing Guidance	17
3.4 Keeping Security in Global Supply Chain	23
Chapter 4 Command Verification by Controlled Devices	26
4.1 Concept of Command Verification.....	26
4.2 Application examples of Command Verification	29
Chapter 5 Application of Command Verification to One Transaction Sub-process..	40
5.1 Introduction	40
5.2 Issues of Existing ATM Systems and Operations.....	41
5.2.1 Existing Cash Withdrawal Transaction with a smart card	41
5.2.2 Issues of ATM Systems and Operations	42
5.2.3 Conditions to Effectively Prevent Jackpotting.....	45

5.3	Application of Command Verification to transaction sub-process handling cash	46
5.3.1	Implementation Idea of Command Verification.....	46
5.3.2	Implementation of Command Verification	48
5.3.3	Evaluation of the Measure.....	54
5.4	Discussion	55
Chapter 6	Application of Command Verification to Two Transaction Sub-processes ..	57
6.1	Introduction	57
6.2	Issues of Existing ATM Systems and Operations.....	58
6.2.1	Existing Deposit Transaction	58
6.2.2	Issues of Existing Security Measures	59
6.3	Application of Command Verification.....	62
6.3.1	Implementation Model Analysis.....	62
6.3.2	Conditions to Prevent Logical Attacks.....	65
6.4	Implementation	66
6.4.1	Implementation Outline.....	66
6.4.2	Detailed Data Flow of Implementation	68
6.4.3	Architecture of the Proposed Peripheral Devices	70
6.4.4	Evaluation of Command Verification through Implementation.....	71
6.5	Discussion	74
Chapter 7	Application of Command Verification to All Transaction Sub-processes	76
7.1	Introduction	76
7.2	Issues of Command Verification.....	77
7.2.1	An ATM System and Magnetic Stripe Card Transaction.....	77
7.2.2	Issues of Existing Security Measures	78
7.2.3	Conditions to Implement Command Verification	80
7.3	Design Method to Implement Command Verification.....	81
7.3	81
7.3.1	Implementation Models	81
7.3.2	Outline of Implementation Design Method.....	83
7.4	Implementation	84
7.4.1	Implementation for Magnetic Stripe Card Transaction.....	84
7.4.2	Detailed Data Flow of the Proper Systems	90
7.4.3	Evaluation of the Design Method.....	93

7.4.4	Architecture of the Proposing Peripheral Devices	94
7.5	Discussion	95
Chapter 8	Conclusion	97
	Bibliography	100
	List of Papers	106

List of Figures

Figure 1.1 Structure of this dissertation	4
Figure 2.1 ATM system structure	5
Figure 2.2 An logical structure of an ATM system	6
Figure 2.3 ATM transaction flow	8
Figure 2.4 Data flow of a cash withdrawal transaction with a smart card.....	9
Figure 3.1 Physical attacks and Logical attacks on ATMs	11
Figure 3.2 A typical attack steps of “Jackpotting”	13
Figure 3.3 A typical attack steps of “Black Boxing”	15
Figure 3.4 A typical attack steps of “Man in the Middle”	16
Figure 3.5 An overview of the existing guidance	18
Figure 3.6 ATM supply chain model.....	24
Figure 4.1 A model of control system with physical action.....	26
Figure 4.2 Objective of existing measures and Command Verification	27
Figure 4.3 Issues of existing measures.....	28
Figure 4.4 Primary model of “Command Verification”	29
Figure 4.5 Existing cash withdrawal transaction with a smart card	30
Figure 4.6 Application example to one transaction sub-process	31
Figure 4.7 Existing deposit transaction with a smart card	33
Figure 4.8 Mechanical structure of cash handling module	33
Figure 4.9 Application to two transaction sub-processes.....	35
Figure 4.10 Existing cash withdrawal transaction with a magnetic stripe card... 37	
Figure 4.11 Application to all transaction sub-processes.....	38
Figure 5.1 Existing cash withdrawal transaction with a smart card	42
Figure 5.2 An outline of the existing guidance	43
Figure 5.3 Implementation idea of Command Verification	47
Figure 5.4 Implementation of Command Verification to one transaction sub-process	49
Figure 5.5 Installation of Certificate Authority’s certificate.....	50
Figure 5.6 Signature verification with ECDSA.....	51
Figure 5.7 Key exchange with ECDH	51
Figure 5.8 Key derivation for encrypted communication	52
Figure 5.9 Comparison of existing devices and proposing devices.....	53
Figure 6.1 Data flow of existing deposit transaction.....	59

Figure 6.2 An outline of the existing guidance	61
Figure 6.3 Implementation models of Command Verification	63
Figure 6.4 Implementation outline of Command Verification.....	67
Figure 6.5 Implementation example of Command Verification	69
Figure 6.6 Comparison of existing devices and proposing devices.....	71
Figure 7.1 Data flow example of existing magnetic stripe card transaction.....	78
Figure 7.2 Implementation models of Command Verification	81
Figure 7.3 Data flow ensuring consistency among transaction sub-processes	87
Figure 7.4 Implementation example of card reader communicating with the host computer.....	88
Figure 7.5 Implementation examples of Command Verification	91
Figure 7.6 Comparison of existing devices and proposing devices.....	95

List of Tables

Table 2.1 Examples of ATM services	7
Table 2.2 ATM transaction types and transaction sub-processes	8
Table 3.1 Description of typical physical attacks and logical attacks on ATMs.....	12
Table 3.2 The numbers of guidance and recommendations that eventually protect the target	19
Table 3.3 Eventual protected targets of the guidance and recommendations in the first line.....	20
Table 3.4 Eventual protected targets of the guidance and recommendations in the second line	21
Table 3.5 Eventual protected targets of the guidance and recommendations in the third line	22
Table 3.6 Market priorities.....	25
Table 4.1 Model features.....	26
Table 5.1 Comparison of the existing requirements and the proposed measure....	55
Table 6.1 Logical attacks for unauthorized deposit.....	60
Table 6.2 Comparison of implementation models of Command Verification	63
Table 6.3 Correspondence between four conditions and prevented logical attacks	65
Table 6.4 Annual numbers of potential unauthorized access to the files to protect	73
Table 7.1 Logical attacks to steal cash from ATMs	79
Table 7.2 Comparison of implementation models for magnetic stripe card transactions	82
Table 7.3 Targeted property and logical attacks.....	85
Table 7.4 Information to verify command and information acquiring device	86
Table 7.5 Summary of applied implementation models.....	89

Important Terms and Abbreviation

(1) Important Terms

authenticity	Authenticity in the sense of cryptography; authentication plus integrity, namely, you can establish that the command/message originated from a trusted entity, which implies integrity.
validity	A term that encompasses authenticity and timeliness
valid	Adjective of “validity”
command validity	
verification	Authentication and timeliness verification of a control command
command	
verification	A shortened form of “command validity verification”
authorized	Having official permission to do something by the host computer, the responsible financial institution, or the responsible vendors

(2) Abbreviation

API	Application Programming Interface
ATM	Automated Teller Machine
BIOS	Basic Input / Output System
CEN/XFS	Comité Européen de Normalisation / eXtensions for Financial Services
DTL	Data Transfer Library
EMV	EuroPay, MasterCard International and Visa International
EPB	Enciphered PIN Block
EPP	Encrypting PIN Pad
FI	Financial Institution
HDD	Hard Disk Drive
HSM	Hardware Security Module
ISO	International Organization for Standardization

MAC	Message Authentication Code
OS	Operating System
PAN	Primary Account Number
PC	Personal Computer
PCI	Payment Card Industry
PCI DSS	Payment Card Industry Data Security Standard
PCI PTS POI	Payment Card Industry PIN Transaction Security Point of Interaction
PIN	Personal Identification Number
WAN	Wide Area Network

Chapter 1 Introduction

1.1 Background

Attacks to ATMs (Automated Teller Machines) used to be only physical attacks such as card skimming to steal cardholder data and physical crash of ATM bodies to steal cash. Recently, criminals frequently utilize logical attacks to steal cash from ATMs in more than 30 countries, resulting in more than ten million US Dollar cash damage, a week to a month and a half ATM service suspension, and serious social issues. Although cashless payments are growing worldwide, cash circulation is gaining in emerging countries as economic growth, and the number of ATMs is also increasing globally. Thus, strengthening ATM security measures is an urgent issue.

In general, an ATM consists of a PC and peripheral devices, such as a card reader and a dispenser handling cash. The PC and a peripheral device are connected with a USB/RS-232C cable, and the PC and the host computer are connected via the financial institution's intranet. A dispenser is usually installed in a safe of an ATM to physically protect cash. Major attack surfaces of the logical attacks described above are the financial institution's intranet [1] [2] [3], the PC [2] [4] [5] [6] [7], and the USB/RS-232C cables [1] [3] [8] [9]. Eventually, unauthorized cash dispensing commands are sent to the dispenser to cash-out from the ATM without conducting a transaction with the host computer in those attacks. The main measures of existing security measures [1] [3] [10] [11] [12] [13] are cryptographic communication between ATMs and the host computer, cryptographic communication between the PC and a peripheral device in an ATM, and anti-malware for the PC to secure the integrity of executable files in the PC.

However, these measures could be bypassed or disabled by criminals since frequent physical/logical access inside ATMs in existing ATM operations. For example, once a few days to a week periodical cash replenishment and collection for cash services, and once a quarter periodical software/contents updating. Securing the integrity of executable files by tight operational management raises the increasing costs of operational management. In particular, it is difficult to secure integrity by limited human resources 24 hours 7 days in case a financial institution operates more than ten thousand ATMs.

1.2 Contribution

This dissertation contributes two points to solve the issues of the existing measures described above. The first point is to propose an ATM security measure effectively and efficiently working in existing ATM operations without overburdening financial institutions. The second point is to provide a general scheme to apply the proposed measure to various ATM systems and transactions. Regarding the first point, we propose a measure called “Command Verification” that controlled peripheral devices themselves verify commands sent from the PC before executing the commands to access the property. A primary model of Command Verification is also proposed so that Command Verification could be applied universally to various systems. Since peripheral devices usually do not have any information to verify a command, two peripheral devices are defined in the model; an information acquiring device and a verified command executing device. The information acquiring device extracts command verification information from input data of the acquiring device and securely transfers the information to the verified command executing device. The verified command executing device verifies a command from the PC with the received information.

To evaluate Command Verification, practical effects of Command Verification and existing measures in existing ATM operations were compared using the application to the cash handling sub-process in a cash withdrawal transaction with a smart card. In general, an ATM transaction consists of four transaction sub-processes: generating a transaction request message, sending the transaction request message to the host computer, receiving a response message from the host computer, and handling cash according to the response message. Three conditions to effectively prevent unauthorized cash withdrawal in existing ATM operations are derived from analysis of existing ATM systems and operations. It was shown that Command Verification can meet the three conditions while the existing measures do not meet them.

Concerning the second point, we propose two methods to apply Command Verification to various systems and transactions. An application to two transaction sub-processes is described to explain the first method, namely, application to a transaction sub-process before/after communication between an ATM and the host computer in a deposit transaction with a smart card. There are two requirements to be satisfied in the application. One is to effectively protect multiple properties from multiple attack surfaces in the two transaction sub-processes. The other is to meet

constraints to be harmonized with existing systems and operations. Since it is difficult to achieve properly implementable systems of Command Verification to meet the requirements, the implementation model analysis is introduced to compare the features of the models in a preliminary step to achieve the proper systems. Two recommended models were derived from the model analysis, and two types of properly implementable systems were finally derived using the recommended models. System management costs were reduced to less than one ten-thousandth of the existing measures with the properly implementable systems.

Regarding the second method, we explain it in an application to all transaction sub-processes, namely, the application to a cash withdrawal transaction with a magnetic stripe card, in which there are few existing security mechanisms. When Command Verification is applied to the magnetic stripe card transactions, there are many implementable systems due to the poor existing security mechanisms. It is difficult to derive proper systems among the many implementable systems, since the proper systems should meet three conditions; preventing a wide range of logical attacks, harmonizing with existing ATM operations and minimizing modification costs of peripheral devices. We propose a systematic implementation design method of Command Verification to derive proper systems, which consists of three steps and guidance. Three proper systems out of the 135 implementable systems were selected by applying the design method to magnetic stripe card transactions.

1.3 Organization

The remainder of this dissertation is organized as follows. Chapter 2 presents the existing ATM systems and services. Chapter 3 describes logical attacks on ATMs and existing measures. Chapter 4 proposes a measure called “Command Verification”, the primary model of the measure, and applied system examples. Chapter 5 describes the issues of existing measures and an application of Command Verification to one transaction sub-process in an ATM transaction. Command Verification and existing measures are compared in terms of the practical effects of the measures in existing ATM operations. Chapter 6 details the issue and the solution for an application of Command Verification to two transaction sub-processes in an ATM transaction. In detail, we propose an implementation model analysis of Command Verification to solve the issue. Chapter 7 presents the issue and the solution for an application of Command Verification to all transaction sub-processes in an ATM transaction. That is, we propose

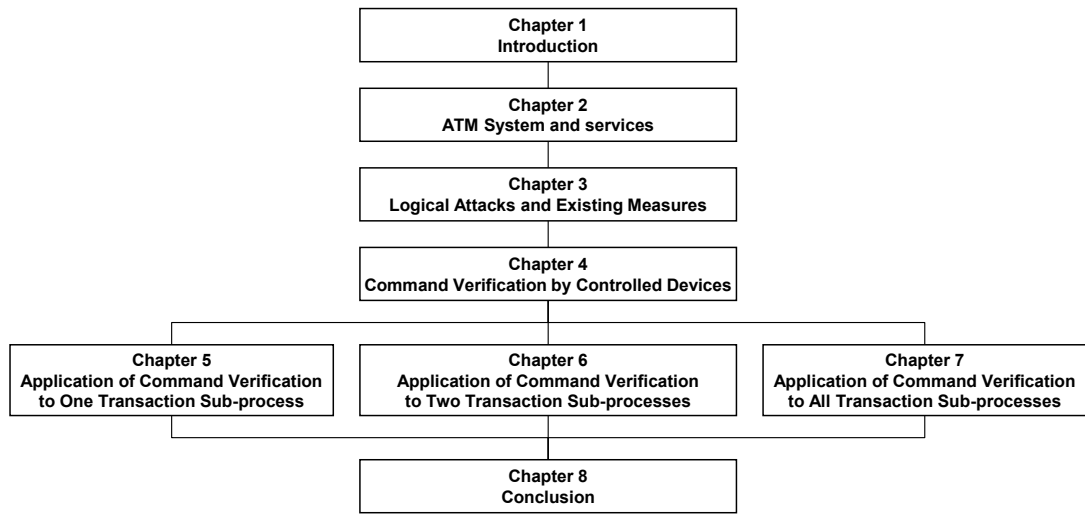


Figure 1.1 Structure of this dissertation

an implementation design method of Command Verification to cope with the issue. Chapter 8 concludes this dissertation.

Chapter 2 ATM System and Services

In this chapter, an ATM system and services are described. In detail, an overview of an ATM system structure, examples of typical ATM services, transaction sub-processes in an ATM transaction, and vulnerable points in an ATM system.

2.1 ATM System Structure

An overview of an ATM system is depicted in Figure 2.1. In general, an ATM consists of a PC and peripheral devices, such as a card reader and a dispenser which handles cash. The PC and a peripheral device are connected with a USB/RS-232C cable, and the PC and the host computer are connected via the financial institution's intranet. The PC is also connected with a software updating server to download software/contents for ATM maintenance. The touch screen shows ATM service menus and transaction results and is used to select a menu and input some parameters required for a transaction. The card reader accepts a smart card and a magnetic stripe card, and reads/writes data

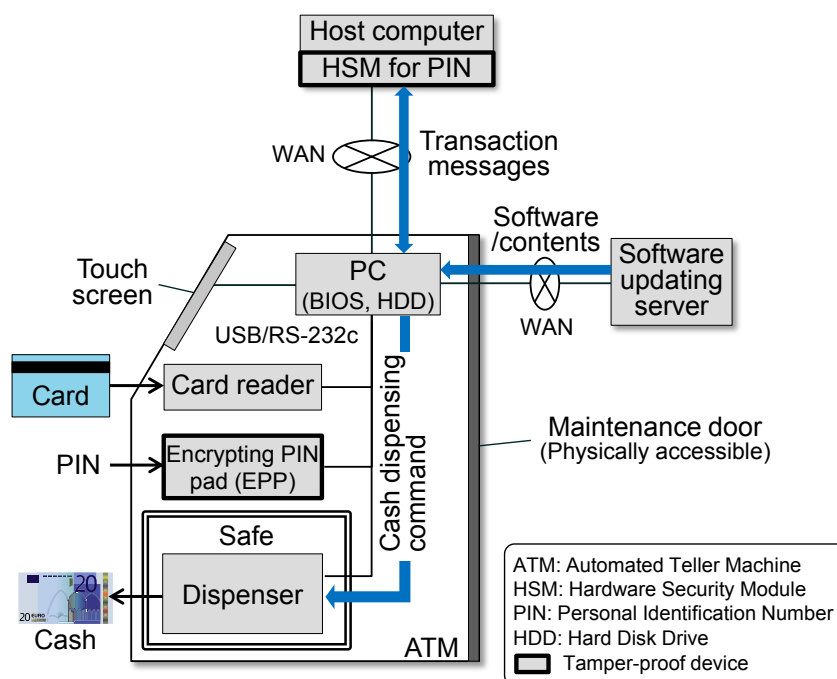


Figure 2.1 ATM system structure

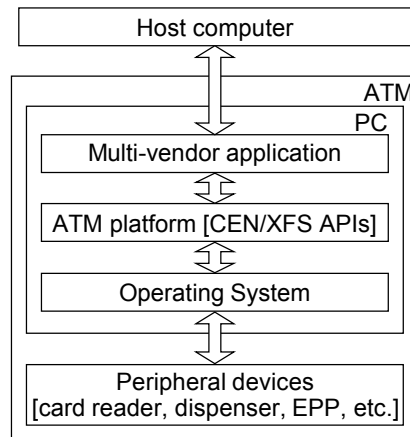


Figure 2.2 A logical structure of an ATM system

from/to the card. The Encrypting PIN Pad (EPP) is a tamper-proof peripheral device used for an ATM user's Personal Identification Number (PIN) entry to show proof of identity. The EPP itself outputs an encrypted PIN called Enciphered PIN Block (EPB), which is cryptographically protected in conformity with the Payment Card Industry (PCI) requirements [14] [15] and ISO 9564 [16] [17] [18]. The EPB is transferred to the Hardware Security Module [19] in the host computer, which is also a tamper-proof device, and the PIN is extracted from the EPB and authenticated in the HSM. A dispenser is usually installed in a safe of an ATM to physically protect cash. Tight access control is required to access the inside of the safe. In the case of so-called cash recycling ATMs, a cash handling module is installed in an ATM instead of a dispenser. The cash handling module dispenses and deposits cash. The maintenance door is to access the inside of the ATM and the door is usually closed with a physical key.

As shown in Figure 2.2, the PC is logically constituted with three layers: multi-vendor application, ATM platform to control the peripheral devices, and Windows^{® 1} OS. The ATM platform provides international standardized interfaces to a multi-vendor application: Comité Européen de Normalisation / eXtensions for Financial Services (CEN/XFS) APIs [20]. The ATM platform architecture was established in the 90s and the primary concepts are interoperability and compatibility. Since the APIs' specifications are open to the public and the APIs are not cryptographically protected, the APIs are frequently abused for unauthorized cash-out.

¹ Windows is either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Table 2.1 Examples of ATM services

#	ATM services	Service contents
1	Cash Withdrawal	To withdraw cash up to a daily limit from a bank account
2	Deposit	To deposit cash to a bank account.
3	PIN change	To change Personal Identification Number (PIN)
4	Balance Enquiry	To check the current available balance in a bank account
5	Card to Card Transfer	To send cash from a bank account to an other account
6	Ministatement	To check the latest several transactions in a bank account
7	Credit Card Payment	To pay bill of a credit card
8	Bill Payment	To pay utility bills
9	Life Insurance Payment	To pay life insurance
10	Trust Donation	Make a donation to a favorite charity
11	Mobile Top-up	To recharge a mobile prepaid connection

Cited from <https://www.sbi.co.in/portal/web/personal-banking/atm-services>

2.2 ATM Services

Examples of typical services provided with overseas ATMs are shown in Table 2.1. No. 1 to 5 are common with ATMs in Japan. Since passbooks are not usually issued in overseas financial Institutions, ministatement is used to check the last several transaction records. As direct debit to a bank account is not popular for credit card payment, bill payment, and life insurance payment in overseas countries, account holders check the bills and pay with an ATM. ATMs accept a donation for charities, and to recharge a mobile prepaid connection.

2.3 ATM Transactions

ATM transactions are categorized into two types in accordance with card types, i.e. a smart card and a magnetic stripe card. There are three types of transactions for each card type: cash withdrawal, deposit, and remittance. As depicted in Figure 2.3, an ATM transaction consists of four transaction sub-processes; generating a transaction request

Table 2.2 ATM transaction types and transaction sub-processes

#	Card type	Transaction type	Transaction sub-process			Described chapter
			S1 Generate transaction request message	S2, S3 Communicate with the host computer	S4 Handle cash according to response message	
1	Smart card	Cash withdrawal			✓	Chapter 5
2		Deposit	✓		✓	Chapter 6
3		Remittance				-
4	Magnetic stripe card	Cash withdrawal	✓	✓	✓	Chapter 7
5		Deposit				-
6		Remittance				-

Transaction sub-process whose authenticity is assured with the existing security functions (EMV specifications) of a smart card and host computer

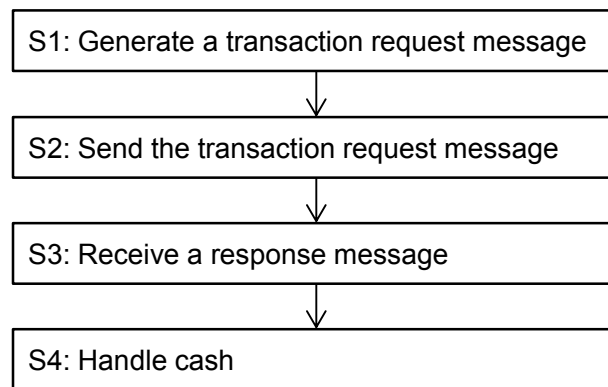


Figure 2.3 ATM transaction flow

message (S1), sending the transaction request message to the host computer (S2), receiving a response message from the host computer (S3), and handling cash according to the response message (S4). In the case of a cash withdrawal transaction, a process flow of the transaction sub-processes is described below. An ATM accepts a PIN and generates a transaction request message with the card data on a card, and other transaction parameter inputted with the touch screen (S1). The ATM sends the request message with the encrypted PIN to the host computer (S2). The host computer verifies the encrypted PIN, checks the account balance, and decides whether the transaction is authorized or not. And then, the host computer sends a response message to the ATM (S3). The ATM handles cash to dispense the cash according to the response message (S4). Although messages transferred between an ATM and the host computer are not cryptographically protected in magnetic stripe card transactions, the messages are cryptographically protected in smart card transactions. That is, the messages are

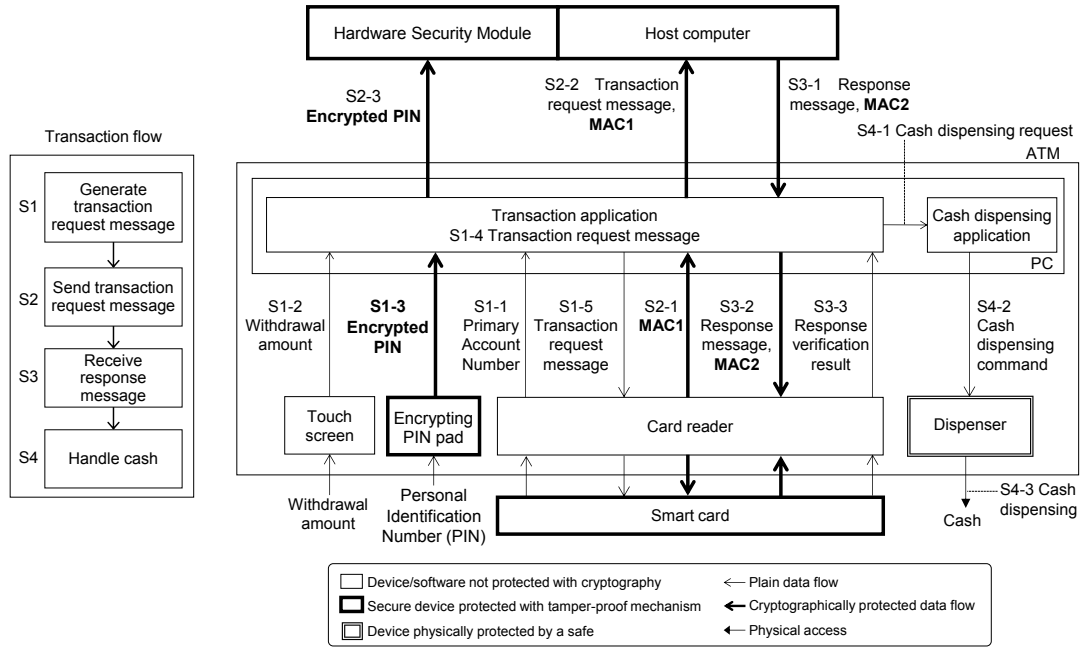


Figure 2.4 Data flow of a cash withdrawal transaction with a smart card

protected with the smart card and the host computer in conformity with the EMV^{® 2} (EuroPay, MasterCard International and Visa International) specifications [21] [22]. A message authentication code is attached to each message, and a cryptographic key for a message authentication code is shared between the smart card and the host computer in conformity with the EMV specifications [21]. The key is also linked with the Primary Account Number (PAN) on the smart card, which is preliminarily assigned to the card by the financial institution to identify the user.

A data flow example of a cash withdrawal transaction with a smart card is illustrated in Figure 2.4. It is supposed that the multi-vendor application includes “transaction application” processing transaction messages and “cash dispensing application” controlling the dispenser. The cryptographic keys to protect a PIN and the messages are supposed to be preliminarily shared. The detailed processes of the four transaction sub-processes are described as follows:

S1: Generating a transaction request message

The transaction application receives an S1-1 Primary Account Number (PAN) on a smart card from the card reader, an S1-2 encrypted PIN from the Encrypting PIN

² EMV is a registered trademark in the U.S. and other countries and an unregistered trademark elsewhere. The EMV trademark is owned by EMVCo.

pad, and an S1-3 withdrawal amount from the touch screen. And then, the application generates a transaction request message (hereinafter called “request message”) from the PAN and the withdrawal amount.

S2: Send the transaction request message

The transaction application sends the S2-1 request message to the smart card. The smart card generates a MAC to the message called “MAC1” and sends the S2-2 MAC1 back to the transaction application. The transaction application sends the S2-3 request message, MAC1, and the encrypted PIN to the host computer. The hardware security module authenticates the user using the PIN extracted from the encrypted PIN, and the host computer verifies the transaction request message with MAC1 and checks the user’s account balance or the credit to decide whether the transaction is authorized or not.

S3: Receive a request message

Then the host computer creates a response message and a message authentication code to the message called “MAC2”, and then sends the S3-1 response message and MAC2 back to the transaction application, and the application forwards S3-2 them to the smart card. The smart card verifies the received message with “MAC2” and returns the S3-3 response verification result to the transaction application. The value of the verification result varies in accordance with the host computer’s decision. It is noted that the response verification result is plain data as the smart card and the transaction application do not share any cryptographic keys.

S4: Handle cash

The transaction application provides the dispensing application with the S4-1 cash dispensing request including the withdrawal amount in accordance with the response verification result. The dispensing application specifies the bill denomination according to the user’s selection and sends an S4-2 cash dispensing command to the dispenser to dispense S4-3 cash.

As explained above, the transaction sub-processes, S1 and S2 are protected with MAC1 and MAC2 in an existing smart card transaction. However, the processes S1 and S4 are not protected in the existing transactions. Furthermore, in case of magnetic stripe card transactions, neither S1 nor S2 is protected.

Chapter 3 Logical Attacks and Existing Measures

3.1 Logical attacks on ATMs and entry points

Typical physical attacks and logical attacks on ATMs are presented in Figure 3.1 and Table 3.1 [1] [23] [24] [25] [26] [27]. In the logical attacks that we focus on, the primary idea to steal cash from ATMs is to send unauthorized cash dispensing commands to the dispensers without the support of a transaction with the host computer. “Man in the Middle” is an attack on the Wide Area Network (WAN) of the financial institution. For example, the attack alters a response message transferred in the WAN to an unauthorized message to cash-out from the ATM. “Jackpotting” is an attack on the PC

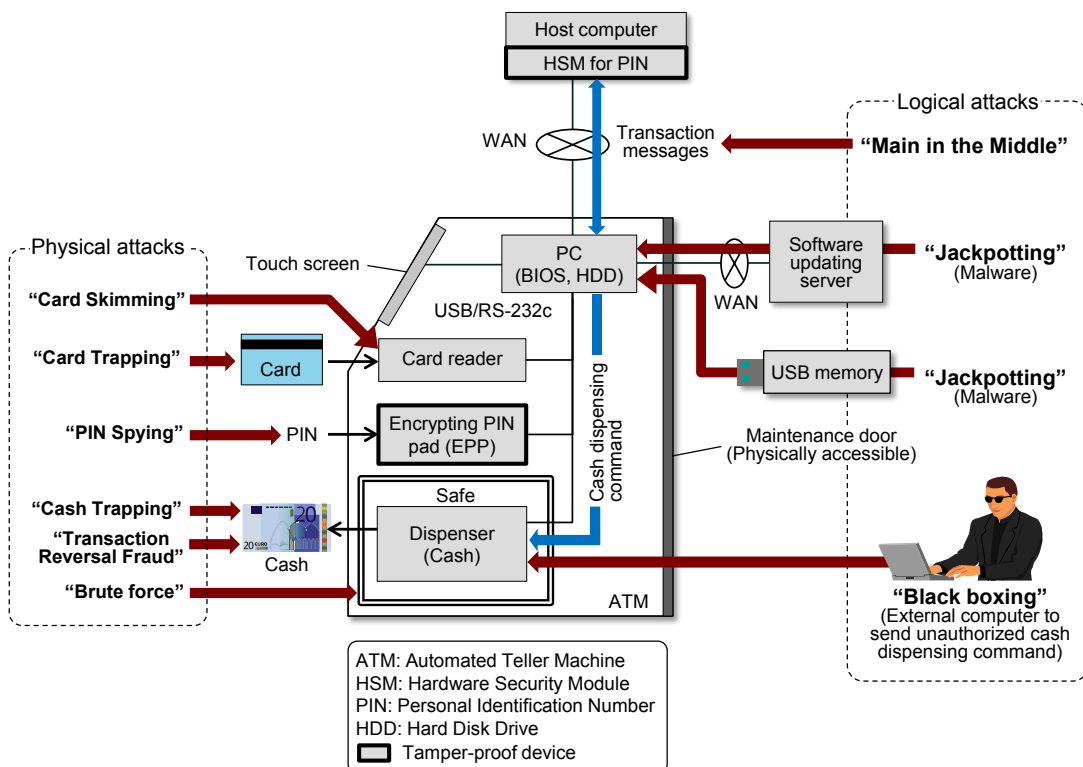


Figure 3.1 Physical attacks and Logical attacks on ATMs

Table 3.1 Description of typical physical attacks and logical attacks on ATMs

#	Category	Attack name	Description
1	Physical attack	Card Skimming	A skimming device is placed on or in the card reader of an ATM to capture the information on the magnetic stripe card.
2		Card Trapping	A card is physically captured by the modified ATM. When the customer leaves the ATM without the card, the card is retrieved by the criminals and used to make fraudulent cash withdrawals or stolen.
3		PIN Spying	There are some methods. Hidden video camera, PIN pad Overlay, and so on. Shoulder surfing is that standing behind the victim, a criminal reads the PIN as it is entered.
4		Cash Trapping	A device is fixed to the cash dispensing slot by criminals, causing cash to get stuck inside the ATM when a customer attempts to withdraw cash. After the customer leaves the ATM, the criminal returns to the ATM to retrieve the cash.
5		Transaction Reversal Fraud	The fraud involves the creation of an error that makes it appear as though the cash had not been dispensed. The account is re-credited the amount 'withdrawn', though the criminal pockets the money.
6		Brute Force	Explosive devices or crashing vehicles into ATMs are used to steal cash with physically crashing ATMs.
7	Logical attack	Man-in-the Middle	Manipulating server responses or recording critical data inside the network.
9		Jackpotting	Malware force the ATM to fraudulently cash-out from the ATM without generating a transaction. Malware installed ATM's PC by means of a CD, DVD, or USB memory; or a network-based action.
10		Black Boxing	An external computer is connected to the cash dispenser and commands the dispenser to fraudulently cash-out.

to install the malware in it and the malware sends unauthorized cash dispensing command to the dispenser without the support of a transaction. There are two entry points; one is the software updating server, and the other is a USB memory with accessing the inside of the ATM by opening the maintenance door. "Black boxing" is an attack on the USB cable between the PC and the dispenser. An external computer directly sends unauthorized cash dispensing commands to the dispenser by connecting the computer with the dispenser. We focus on the logical attacks in this dissertation. The detailed scenarios of those logical attacks are described in the following section.

3.2 Typical Logical attacks on ATMs

3.2.1 Jackpotting

One of the typical logical attacks is called “Jackpotting” [1] [2] [4] [5] [6] [7], which is such an attack that malware in the PC of an ATM sends unauthorized cash dispensing commands to the dispenser to withdraw cash from the ATM without generating a transaction. Jackpotting is prevailing much more than “Black Boxing” described in the following section because Jackpotting uses only software that attacks the vulnerable ATM platform commonly installed in the PCs. There are two types of Jackpotting; physical ATM malware attacks and network-based ATM malware attacks. Typical attack steps of the physical ATM malware attacks are depicted in Figure 3.2 and the detailed attack steps are described below.

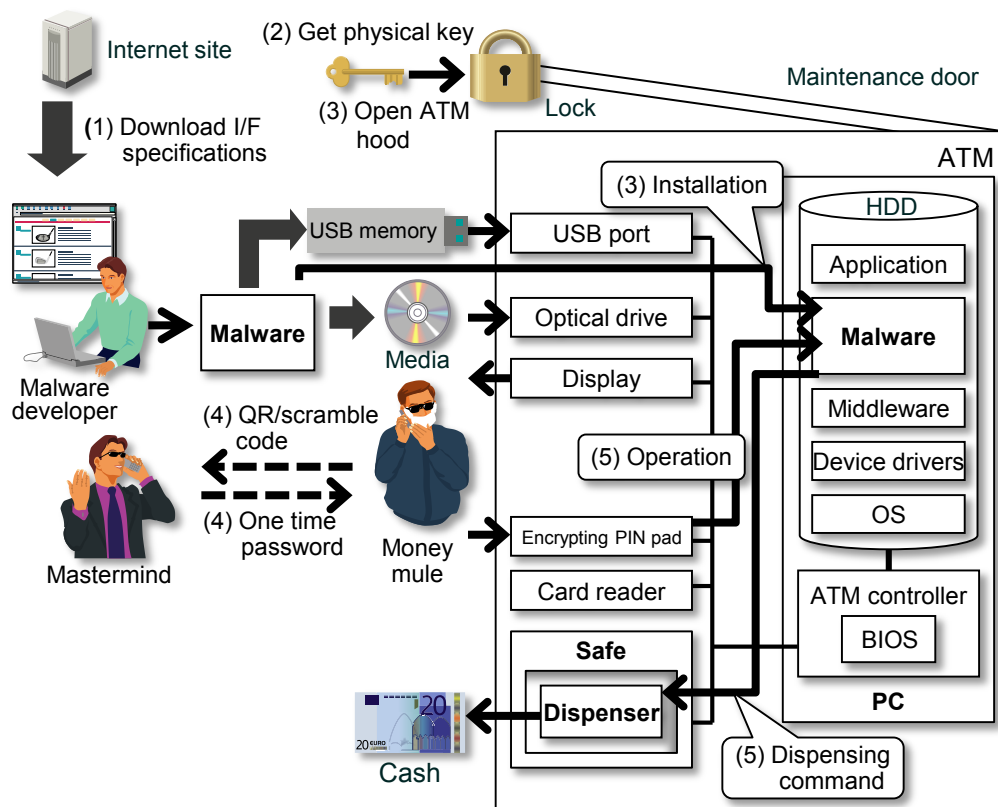


Figure 3.2 A typical attack steps of “Jackpotting”

- Step 1: Malicious people download industry-standard interface specification documents and develop malware according to the specification documents.
- Step 2: Malicious people get or copy a physical key to open the maintenance door of poorly managed ATMs. In some cases, the physical key is the same for all of the ATMs for easy management of the ATMs. Malicious people also get an OS administrator's password to install malware into the PC. The administrator's password is also the same for all of the poorly managed ATMs in many cases.
- Step 3: A money mule in the malicious people opens the maintenance door of a targeted ATM with the physical key acquired beforehand. There are two ways to install malware into the PC after that. One is that the money mule reboots the PC with a medium such as a USB memory stick or a CD-ROM containing an OS and malware to install the malware in the medium into the PC. Another is that the money mule logs in the OS administrator mode of the PC with the administrator's password acquired beforehand, and is that the money mule inserts a USB memory stick or a CD-ROM into the PC to install the malware from the medium in the administrator mode. Then the money mule reboots the PC for installed malware to be run.
- Step 4: An additional procedure to get a one-time password is required for the money mule to dispense cash from the ATM with malware due to malicious people's self-defense. As malware is just software that anyone can duplicate it, other malicious people may utilize it to dispense cash from targeted ATMs without authorization of the malicious people. After booting malware, the money mule sends some QR code or a scramble code, which is required to get a one-time password, shown on the ATM screen by malware to a remote server or a mastermind with a cell phone or an SMS mail. Some malware has a further mechanism for malicious people's self-defense, namely, cash traceability to prevent even a friendly money mule from cheating the amount of cash dispensed from an ATM. Some malware can count the amount of the cash stored in the ATM and can implement the cash amount to the QR code or the scramble code so that the mastermind can check later whether the money mule does not cheat the amount of the collected cash.
- Step 5: The money mule receives a one-time password from the server or the mastermind and inputs it into malware with the encrypting PIN pad, and then malware can be activated to send unauthorized dispensing commands to the dispenser. Then the money mule repeats to send the unauthorized dispensing command to the dispenser and to receive the cash dispensed from the ATM

until cash in the safe becomes empty.

Regarding the network-based ATM malware attacks, malicious people hack into the financial institute's intranet with such a way as phishing e-mails sent to the financial institute's employees, or other ways. Once the malicious people intruded in the intranet, they perform lateral movement to find the ATM network and compromise the ATM software delivery server in the ATM network to distribute malware to the ATMs.

3.2.2 Black boxing

“Black Boxing” [1] [8] [9] is a variant of “Jackpotting”, where the ATM PC is not used. Instead, a malicious person brings an external computer with him/her, and then the computer is directly connected with the dispenser. Malware on the computer sends unauthorized cash dispensing commands to the dispenser. As the malware communicates directly with the dispenser, each “Black Box” attack is only valid for one type of dispenser. Typical attack steps are depicted in Figure 3.3 and the detailed

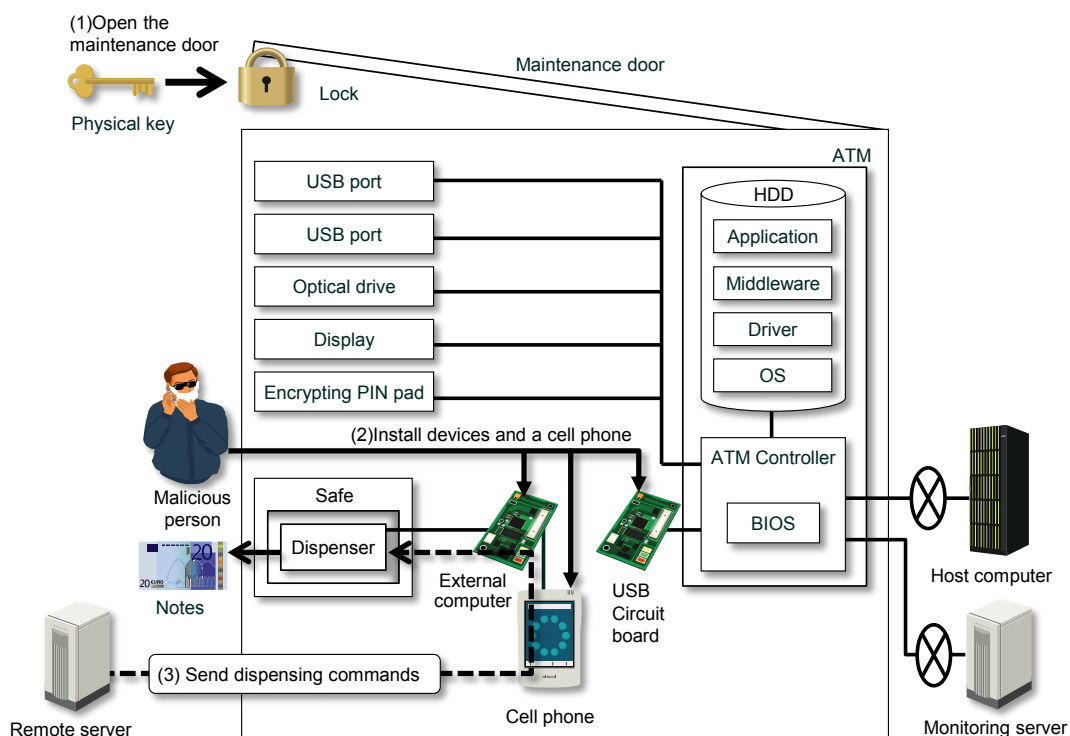


Figure 3.3 A typical attack steps of “Black Boxing”

attack steps are described below.

Step 1: A malicious person opens the maintenance door of an ATM with a poorly managed physical key described above.

Step 2: The malicious person installs an external computer pretending an ATM PC, a USB circuit board pretending a dispenser, and a cell phone to remotely control the external computer into an ATM. The external computer takes control of communication between the ATM PC and the dispenser. The external computer can send dispensing commands to the dispenser following a command transmitted through the cell phone, which has no relation to ATM transactions. The USB circuit board is used to make the monitoring server delay to find the anomaly.

Step 3: The malicious person waits for the commands from the remote server transmitted through the cell phone, and receives cash dispensed from the ATM.

3.2.3 Man in the Middle

“Man-in-the-Middle” [1] [28] [29] focuses on the communication between an ATM PC and the host computer. For example, malware can fake host response messages to withdraw money without debiting the fraudster’s account. Typically the malware is triggered during transactions with pre-configured card numbers. The malware can be implemented at a high software layer of the ATM PC or somewhere within the financial institution’s network. Typical attack steps are depicted in Figure 3.4 and the

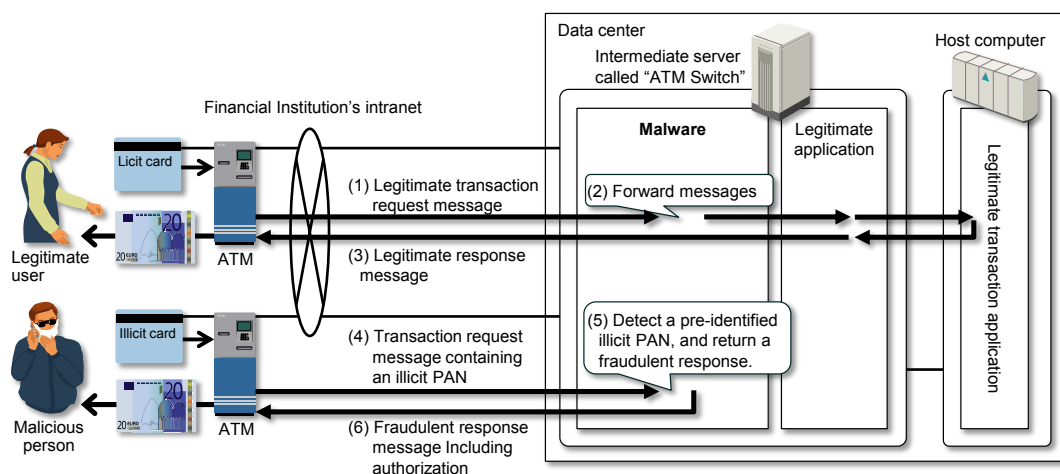


Figure 3.4 A typical attack steps of “Man in the Middle”

detailed attack steps are described below.

- Step 1: An ATM sends an authorized transaction request message NOT including an illicit Primary Account Number (PAN) to the intermediate server called “ATM Switch”.
- Step 2: The Malware running on the compromised switch server inspects the transaction request message whether the request message contains one of the pre-identified illicit PANs. If the request message does NOT contain one of the pre-identified illicit PANs, the malware forwards the message to the legitimate application.
- Step 3: The legitimate application on the ATM switch sends a legitimate response message including the host computer’s decision: transaction authorization/rejection.
- Step 4: A malicious person inserts a card including an illicit PAN into an ATM. And then, the ATM sends a transaction request message including the illicit PAN to the ATM Switch.
- Step 5: Since the request message contains one of the pre-identified illicit PANs, the malware generates a fraudulent response message for any Personal Identification Number (PIN) included in the request message, leaving the host computer without the knowledge of the transaction.
- Step 6: The malware sends the fraudulent response message including host authorization back to the ATM to dispense cash to the malicious person.

3.3 Existing Guidance

Existing guidance [1] [3] [10] [13] including recommended measures are issued from many countries and ATM vendors, whether they are public or private. The primary concepts of the guidance are similar among them. Thus, the diagram that summarizes multiple guidelines by associating logical attacks with recommended countermeasures is depicted in Figure 3.5. “Guidance and Recommendations regarding logical attacks on ATMs” [1] issued from “European law enforcement agency” is introduced as typical examples of existing guidelines and measures. The document is hereinafter called “EUROPOL’s guidance”. The EUROPOL’s guidance addresses multiple logical attacks and provides guidance and recommendations for countermeasures against the attacks. The addressed attacks are Jackpotting, Black boxing, and Software Skimming.

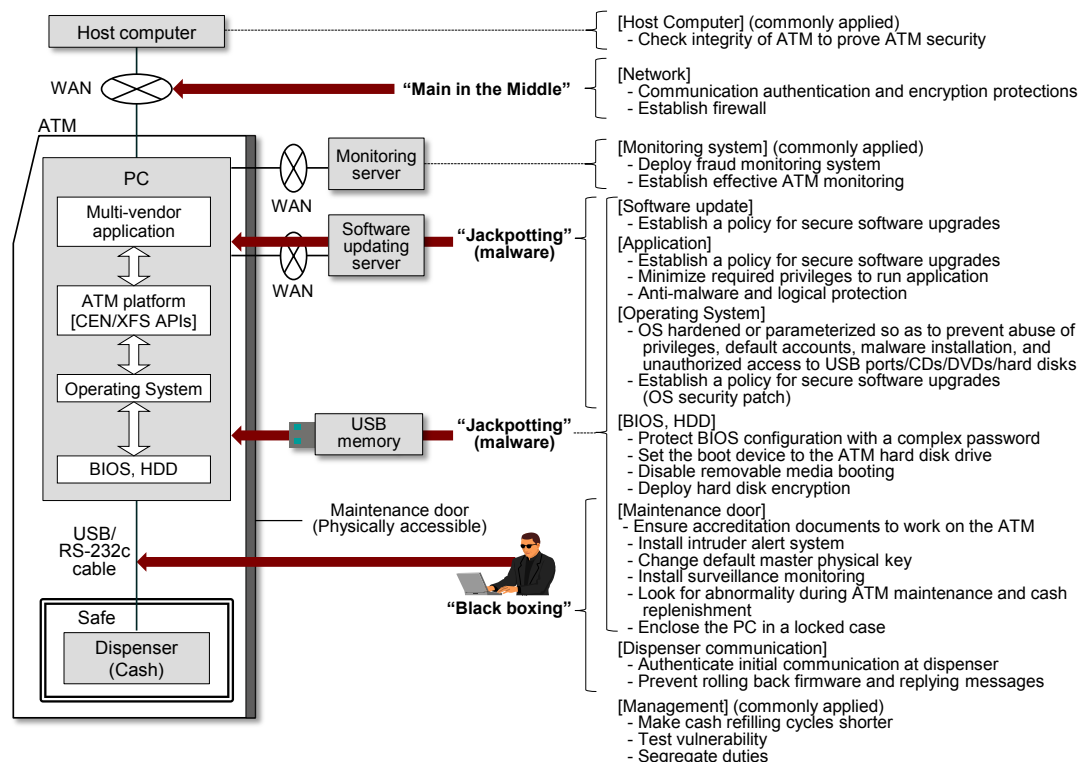


Figure 3.5 An overview of the existing guidance

Concerning the third attacks, software skimming malware intercepts primary account numbers and personal identification number at the ATM, allowing the malicious people to copy the data and later create counterfeit magnetic stripe cards. Since the existing measures [14] [15] [30] have already been provided, we do not discuss Software Skimming and focus Jackpotting and Black boxing for unauthorized cash withdrawals in this dissertation. The recommendations of the EUROPOL's guidance almost overlap those of the existing guidance in Figure 3.5. There are guidance and recommendations for each layer of an ATM system and management: a monitoring system, host computer, application, operating system, BIOS, Hard Disk Drive (HDD), dispenser communication, ATM body, and management.

The EUROPOL's guidance recommends a layered approach to protect cash in ATMs from the logical attacks, and the requirements are categorized to the following lines of defense:

The First Line: Physical access to the ATM

The Second Line: Offline protection

The Third Line: Online protection

The Forth Line: Additional measures

Table 3.2 The numbers of guidance and recommendations that eventually protect the target

No.	Guidance and Recommendations	PC	Peripheral device	FI's network
1	The First Line: Physical access to the ATM	4/4	3/4	0/4
2	The Second Line: Offline protection	6/8	2/8	0/8
3	The Third Line: Online protection	6/8	0/8	2/8
	Total	16/20	5/20	2/20

Although there are various guidance and many recommendations in the EUROPOL's guidance, a large part of them is dedicated protecting information property in the PC of ATMs. To clarify the situation, each guidance and recommendation is categorized into three protective targets: Information property in the ATM PC (hereafter called "PC" in this section, peripheral devices in an ATM (hereafter called "peripheral device" in this section), and the financial institution's (FI's) network. Since the large part of the fourth line is related to policy and monitoring, the following comparison focuses on the first line, the second line, and the third line. As shown in Table 3.2, the eventual goal of the 80% of requirements in the EUROPOL's guidance are related to protecting information property in the PC, the 25% are protecting peripheral devices, and the 10% are protecting the financial institution's network, respectively. Thus, it is found that the EUROPOL's guidance tries to prevent logical attacks by primarily protecting information property in the PC. The following tables show categories of each guidance and recommendation in the three defense lines.

The First Line: Physical access to the ATM

Table 3.3 Eventual protected targets of the guidance and recommendations in the first line

No.	Guidance and Recommendations	PC	Peripheral device	FI's network
1	Ensure that authorized service providers carry accreditation documents and that there is a procedure for ATM site personnel to authenticate their authorization to work on the ATM.	✓	✓	
2	Usually the top compartment (top box) of an ATM contains the PC. This area should be secured by an intruder alert to prevent unauthorized opening, or the access lock to the top box should be changed to avoid the usage of default master keys provided by the manufacturer.	✓		
3	Surveillance monitoring (cameras) should be in place, which will also detect and record suspicious activity around the ATM. If surveillance monitoring is used, then camera/video images should be stored externally to the ATM, and operations of the cameras should not be interrupted by an ATM reboot.	✓	✓	
4	There should be adequate lighting in and around the ATM.	✓	✓	

The Second Line: Offline protection

Table 3.4 Eventual protected targets of the guidance and recommendations in the second line

No.	Guidance and Recommendations		PC	Peripheral device	FI's network
1-1	BIOS configuration	Consider robust password management policies. Best practice indicates that these passwords should be as complex as the BIOS can support.	✓		
1-2		Set the BIOS to boot only from the ATM hard drive.	✓		
1-3		Bootling from removable media should be disabled by default.	✓		
1-4		Apply a robust operating system administrator password.	✓		
1-5		Ensure AUTORUN has been fully and effectively disabled.	✓		
2	Hard disk encryption	Hard disk encryption should be deployed to prevent unauthorized changes to the content of the hard drive.	✓		
3	Cash Dispenser Communications	To prevent unauthorized devices from sending commands to the cash dispenser, the initial communication should require authentication at the cash dispenser. e.g. by physical access to the safe.		✓	
4		It should not be possible to circumvent the communication's protection e.g. by rolling back firmware, or by replaying messages.		✓	

The Third Line: Online protection

Table 3.5 Eventual protected targets of the guidance and recommendations in the third line

No.	Guidance and Recommendations		PC	Peripheral device	FT's network
1	Network	Communication authentication and encryption protections should be applied to all ATM network traffic. The recommendation is to use TLS 1.2 or a VPN, and by implementing MACing to provide cryptographic authentication of sensitive messages.			✓
2	Firewall	A Firewall should be established to restrict all inbound communication to the ATM.			✓
3-1	Operating system	The OS is to enforce strict application separation. For example the unauthorized use of various services (OS, Platform, including XFS and Applications) is to be prevented at all times e.g., runtime, service and administration.	✓		
3-2		Unused services and applications are to be removed.	✓		
3-3		Establish a policy for secure software upgrades.	✓		
3-4		Ensure the application runs in a locked down account with the minimum required privileges not being root or administrator.	✓		
4	Anti-malware and logical protection	An ATM specific anti-malware and logical solution based on the “white listing” or “sandboxing” principles should be employed	✓		
5	USB protection	The use of unknown USB devices should be blocked.	✓		

3.4 Keeping Security in Global Supply Chain

This section describes maintenance and improvement of ATM security in the global supply chain [31]. In the domestic market, security risks in the supply chain could be addressed with the support of established vendors. However, in a global market where security, quality, delivery and cost priorities are completely different from the domestic market, different ideas are needed. Specifically, after clarifying the roles of domestic and overseas roles in the supply chain, the necessary security measures have been taken, including security standards compliance and existing security measures, with the cooperation of the financial institutions.

There are three challenges when a domestic ATM vendor joins overseas markets.

Challenge 1: The quality of ATM security should be maintained throughout the ATM supply chain.

When entering overseas markets, design and production locations are often different. Even if ATMs have secure designs, the resulting ATM products will not always be manufactured according to those designs.

Challenges 2: Regulations and/or standards required by financial institutions should be conformed to efficiently.

When an ATM vendor expands the market to another country, the vendor may be required to conform to unknown regulations and/or standards in the target country. The greater the number of unknown or updated regulations or standards is, the higher the cost of internal processes such as query and reply become. Efficient ways to reduce internal costs are required.

Challenge 3: Companies must become more competitive to succeed globally and to get more market share.

Security features and capabilities are more important than other functions and features since global vendors' value security patent positions more than other feature patent positions.

To raise the security level of a domestic ATM vendor to the global standards, the following security practices and controls which consist of three steps are required.

First Step: Setting a Security Target for ATMs

An ATM vendor-specific security target was created based on publicly available Protection Profiles [32] [33] [34] related to ATM security, and the security target helped to identify ATM security risks.

Second Step: Conforming to Regulations and Standards

Regulations and standards in overseas markets need to be categorized into similar and unique ones in order to conform to those regulations and standards efficiently and promptly. Thus, all requirements were mapped to a two-dimensional matrix of location and time, and similar requirements in each cell were merged into one new requirement.

Third Step: Enhancing Supply Chain Security

As shown in Figure 3.6, global supply chains such as the Chinese or Indian markets can be represented using the supply chain operations reference (SCOR) model [35]. There are very few entities that a vendor can control in overseas markets, unlike the

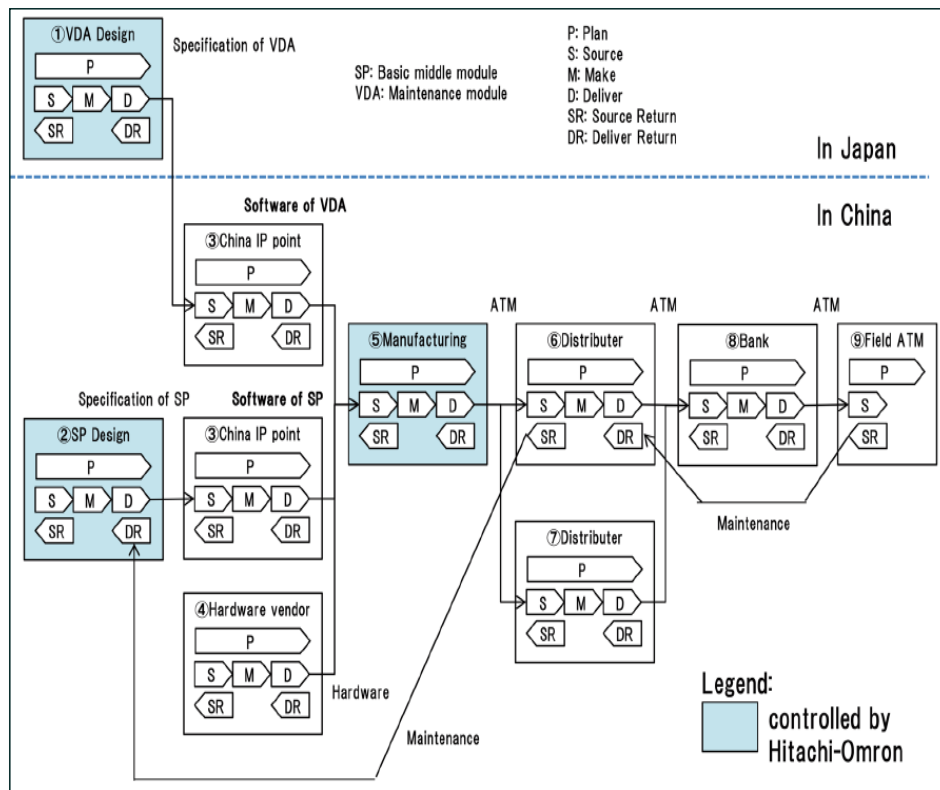


Figure 3.6 ATM supply chain model

Table 3.6 Market priorities

Priority	Domestic (Japan)	Overseas
1	Quality (Security)	Delivery
2	Cost	Cost
3	Delivery	Quality (Security)

domestic market. The priorities of quality, cost and delivery are different between domestic and overseas markets as shown in Table 3.6. To cope with the situation, it is important to establish shared responsibility. To analyze shared responsibility, we applied a risk management framework according to ISO 31000 [36]. Specifically, a top event that is shared between one stakeholder and another is broken into basic events by the Fault Tree Analysis, and each basic event is assigned as the responsibility of one stakeholder. From a vendor perspective, hardware security is very important as a point of trust in the supply chain.

Chapter 4 Command Verification by

Controlled Devices

In this chapter, a security measure called “Command Verification” and a primary model of the measure are proposed in order to solve the issues of the existing security measures described in section 1.1.

4.1 Concept of Command Verification

A model of an existing control system with physical action is shown in Figure 4.1. A controller (PC) totally controls an actuator (peripheral device) on the basis of a controller-actuator model. An existing ATM also works on the basis of a controller-actuator model. The controller (PC) sends a valid command to an actuator

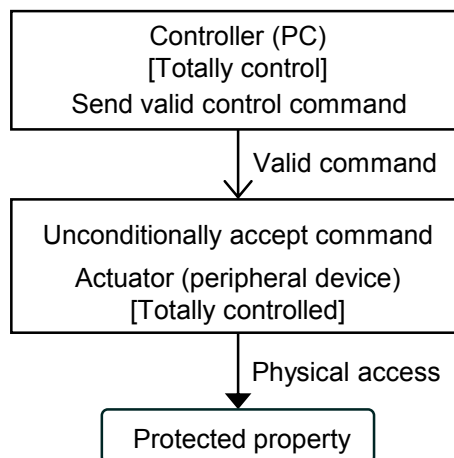


Figure 4.1 A model of control system with physical action

Table 4.1 Model features

Device	Structure/ functions	Securely operating condition
Controller	Complicated	Send only valid commands
Actuator	Simple	Receive only valid commands

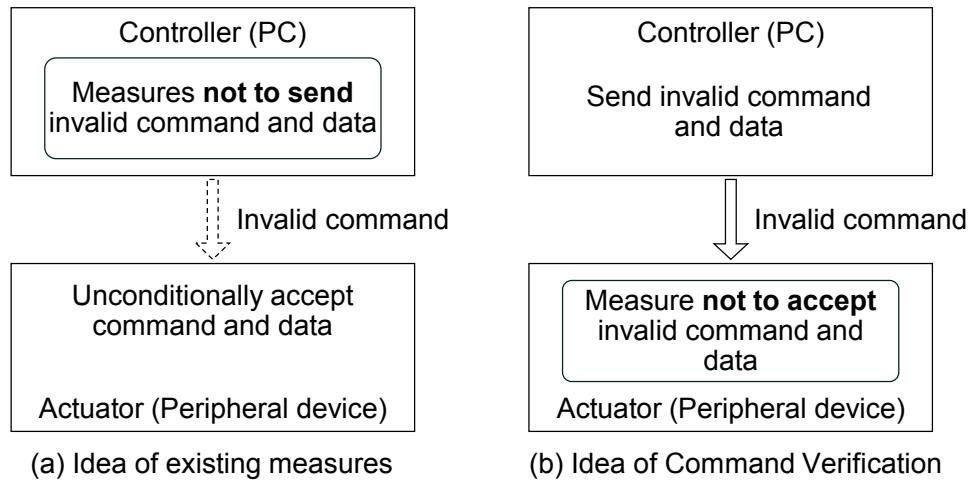


Figure 4.2 Objective of existing measures and Command Verification

(peripheral device) to control the actuator, and the actuator accepts the command unconditionally to access protected property. In general, the controller has complicated structures and functions while the actuator has simple structures and functions (Table 4.1). The system securely works if the controller is secure, although it is not always secure in the actual situations.

Comparison between objectives of existing measures and the “Command Verification” proposed in the paper [37] is depicted in Figure 4.2. One objective of the existing measures is to protect information property in the PC so that the PC does not send invalid commands and data to the peripheral device (Figure 4.2 (a)). Another objective is to protect the communication cable between the PC and the peripheral device so that the peripheral device does not receive invalid commands from an external PC although it is not depicted in the figure. However, there are issues of existing measures as illustrated in Figure 4.3. One is that there are many potential intrusion routes in the PC due to the complicated structures. Even though a security patch is applied to the PC to block the intrusion route, there could be still some potential intrusion routes. The other is that those measures could be bypassed or disabled by criminals since frequent physical/logical access to the inside of ATMs are required in existing ATM operations. For example, once a few days to a week periodical cash replenishment and collection for cash services, once every six months cleaning/maintenance, and quarterly periodical software/content updating for better services. ATM management costs could increase if the integrity of executable files is assured by tight ATM operational managements to cope with that issue. Furthermore,

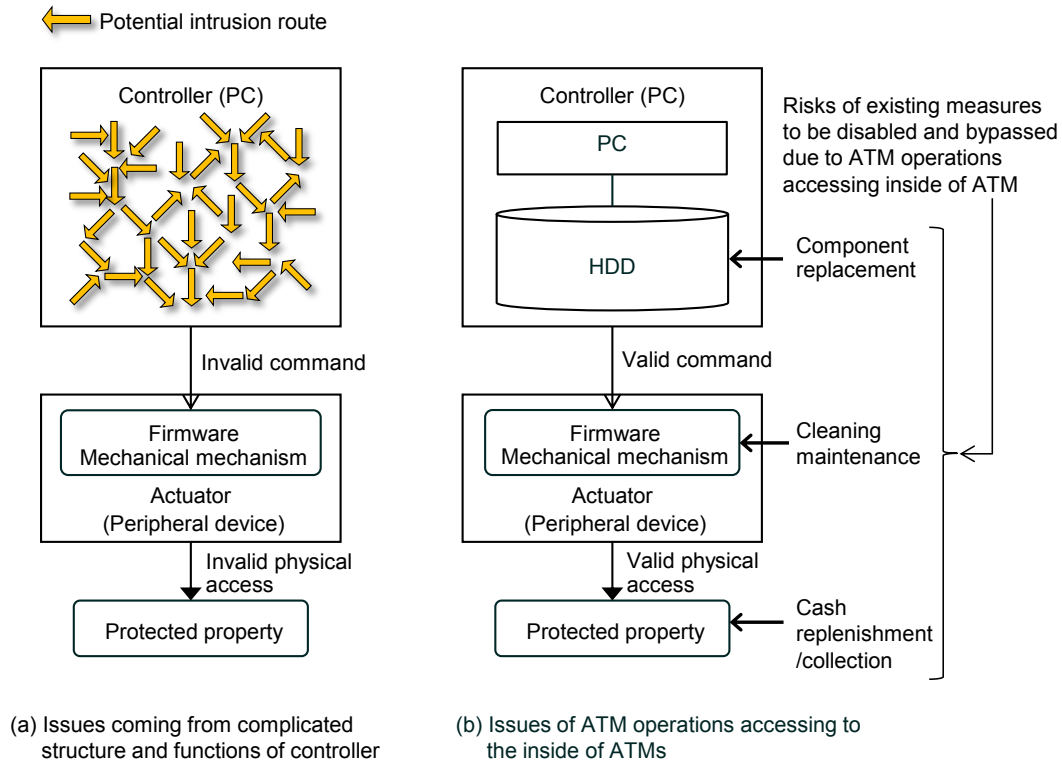


Figure 4.3 Issues of existing measures

it is difficult to secure integrity by limited human resources 24 hours 7days in case that a financial institution operates more than ten thousand ATMs.

On the other hand, the objective of Command Verification is for peripheral devices themselves to verify received commands and data so that the peripheral devices do not accept invalid commands and data accessing protected property (Figure 4.2 (b)). In the proposed measure, tight protection of the PC is not required and the peripheral device should be tightly protected. As the peripheral device has simple structures and functions, the tight protection of the peripheral device is much easier than the PC.

The primary model of the “Command Verification” is illustrated in Figure 4.4, which was proposed to achieve the objective in the paper [37]. An existing peripheral device accessing property may not have any information to verify the validity of a command received from the control unit (PC). Therefore, two kinds of peripheral devices protected with a tamper-proof mechanism, are defined in the primary model: an information acquiring device and a verified command executing device. The function of the information acquiring device is to extract verification information from input data of the device and to send the verification information in a cryptographically

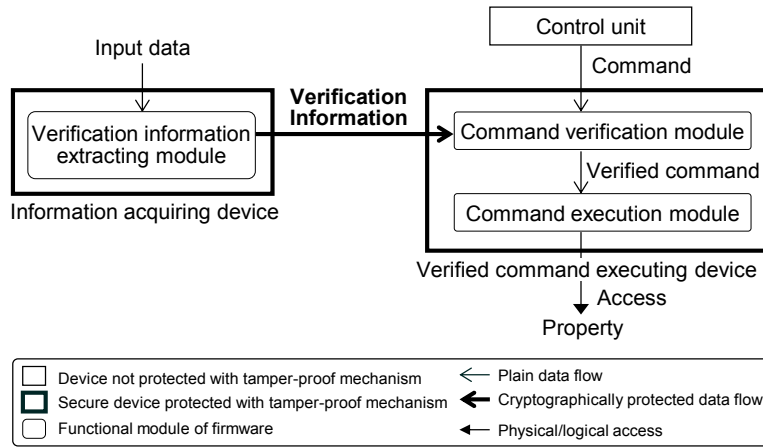


Figure 4.4 Primary model of “Command Verification”

protected form to the verified command executing device. The function of the verified command executing device is to verify a command accessing the property with the verification information and to execute the command if the command is successfully verified.

4.2 Application examples of Command Verification

Application examples of Command Verification are explained in this section. An ATM transaction consists of four transaction sub-processes as outlined in section 2.3, application examples to one transaction sub-process, two transaction sub-processes, and all transaction sub-processes are illustrated.

(a) Application to one-transaction sub-process

An application to the transaction sub-process, handling cash in a cash withdrawal transaction with a smart card is described as an application to one transaction sub-process. Figure 4.5 outlines a data flow example of the existing cash withdrawal transaction with a smart card. It is supposed that the multi-vendor application includes “transaction application” processing transaction messages and “cash dispensing application” controlling the dispenser. A transaction consists of four sub-processes: (S1) generating a transaction request message, (S2) sending the transaction request message, (S3) receiving a response message, and (S4) handling cash. The transaction sub-process S1 and the system related to a PIN are omitted in Figure 4.5. Refer to section 2.3 regarding the detailed data flow of the transaction.

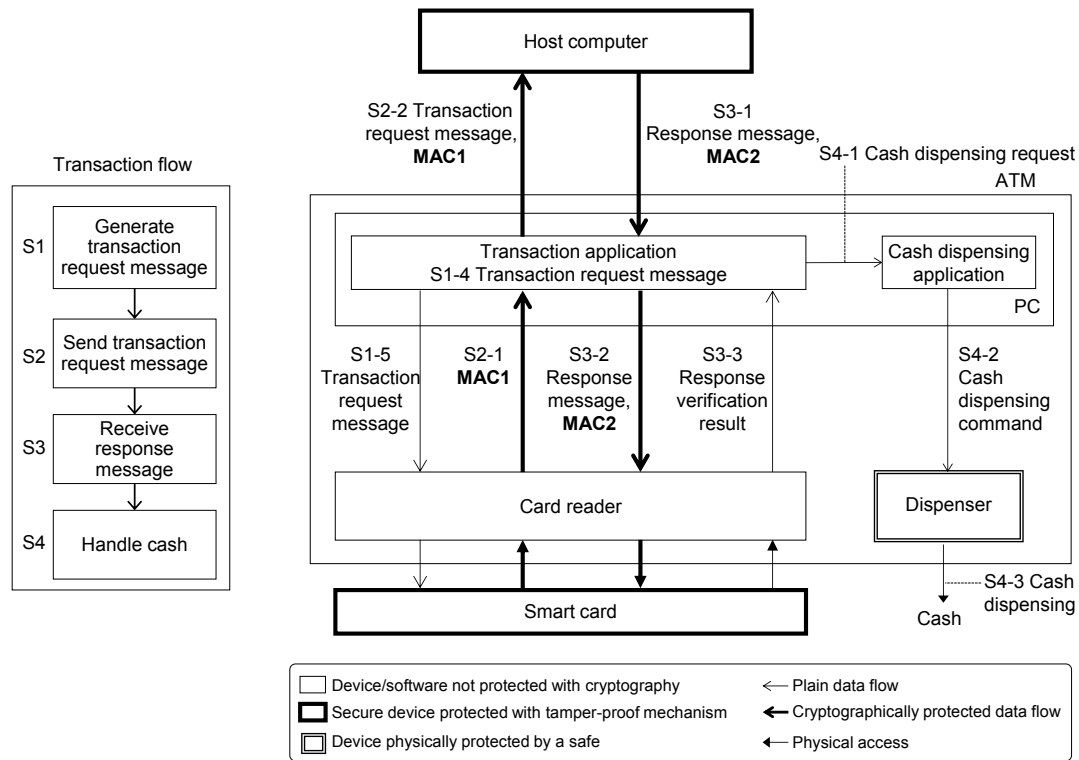


Figure 4.5 Existing cash withdrawal transaction with a smart card

An application of Command Verification to the S4 transaction sub-process [37] is outlined in Figure 4.6. There is not any physical communication cable between the existing card reader and the existing dispenser. An encrypted communication to transfer the authorized withdrawal amount is implemented with the PC and the existing communication cables between the PC and each peripheral device. Data Transfer Library (DTL) is newly introduced to simply provide a communication path between those devices to transfer encrypted data. The DTL is supposed to be installed in a layer below the applications. Even if the DTL is infected with malware, the integrity of encrypted data transferred in the DTL is still assured. A tamper-proof secure element, whose detail is explained in 5.3.2, is implemented in the proposed card reader. The secure element is equipped with two functions; one is to extract an authorized withdrawal amount from S1-5 and S3-3, and the other is to securely transfer the authorized withdrawal amount to the proposed dispenser. A tamper-proof secure element is also implemented in the proposed dispenser, which is equipped with two functions; one is to securely receive the authorized withdrawal amount from the proposed card reader, the other is to verify a received cash dispensing command with

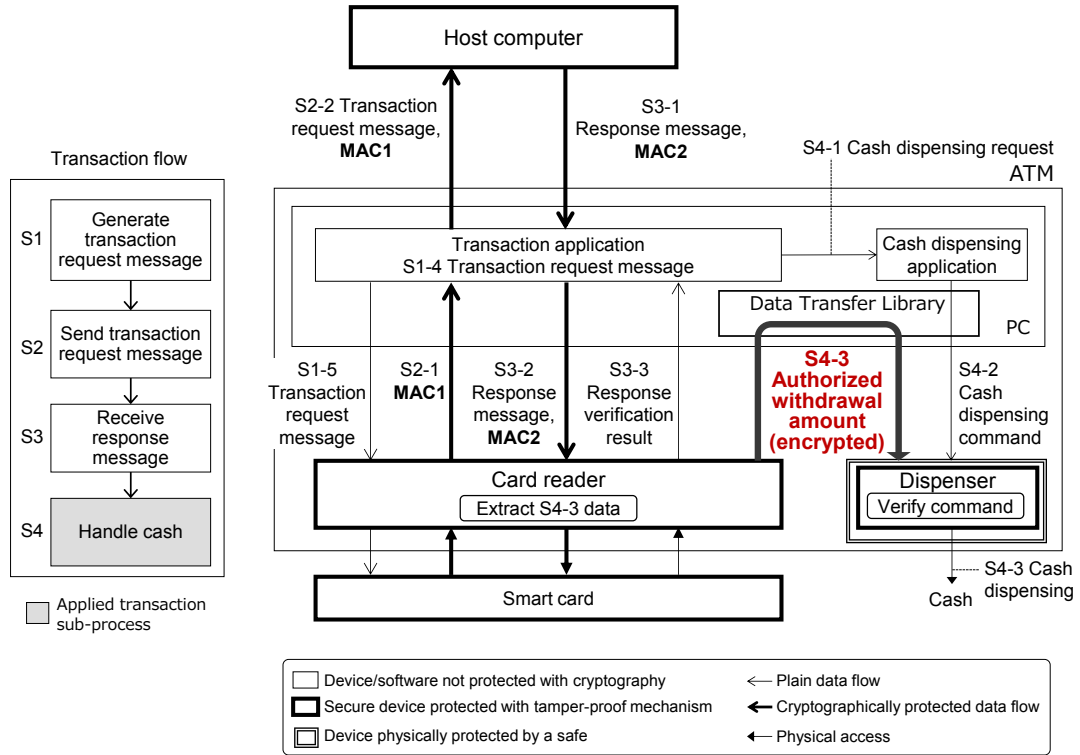


Figure 4.6 Application example to one transaction sub-process

the authorized withdrawal amount. The cryptographic key management and a session creation for the encrypted communication are supposed to conform to the international standards [14] [15] [19] [38] [39] [40] [30] [41] [42]. A session is supposed to have been preliminarily created. The detailed process flows are described as follows. Only modified processes are explained here. The transaction application sends the S1-5 transaction request message to the smart card through the card reader. The secure element in the card reader captures the message and extracts a withdrawal amount and stores it in the element.

S3: Receive a request message

The smart card returns the S3-3 response verification result to the transaction application through the card reader. The secure element in the card reader captures the verification result and generates the authorized withdrawal amount from the withdrawal amount stored in S1-5 and the verification result. Then the secure element encrypts the amount and stores the encrypted amount in the element.

S4: Handle cash

The dispensing application sends an S4-2 cash dispensing command to the dispenser through the DTL. When the DTL receives the dispensing command, the DTL requests the proposed card reader to send the S4-3 encrypted amount. Then the DTL forwards the dispensing command and the S4-3 encrypted amount to the proposed dispenser. The secure element in the dispenser decrypts the S4-3 encrypted amount and confirms whether the dispensing amount in S4-2 and the authorized withdrawal amount are identical or not. When multiple bill denominations are specified in the dispensing command, the aggregate amount in the command is compared with the authorized withdrawal amount. If those amounts are identical, the dispenser dispenses cash following the dispensing command.

(b) Application to two-transaction sub-processes

An application to a deposit transaction with a smart card is explained as an application to two transaction sub-processes. Figure 4.7 outlines a data flow example of the existing deposit transaction with a smart card, and Figure 4.9 illustrates an example of the applied system of Command Verification and the related the data flow. It is supposed that the multi-vendor application includes “transaction application” processing transaction messages and cash handling application” controlling a cash handling module instead of a dispenser. The cash handling module dispenses and deposits cash. The detailed processes of the four transaction sub-processes of the existing transaction are described as follows:

S1: Generating a transaction request message

The transaction application receives an S1-1 Primary Account Number (PAN) on a smart card from the card reader. Cash put into the cash pocket of the cash handling module by an ATM user is transported to the bill validator to confirm whether it is genuine or counterfeit, and to count the cash amount (Figure 4.8). The cash is further transported to the intermediate stacker. The cash handling module outputs the S1-2 cash amount to the cash handling application. The cash handling application sends the S1-3 cash amount to the transaction application. The transaction application creates a S1-4 transaction request message from the PAN and the cash amount for deposit. The transaction application sends the S1-5 request message to the smart card through the card reader.

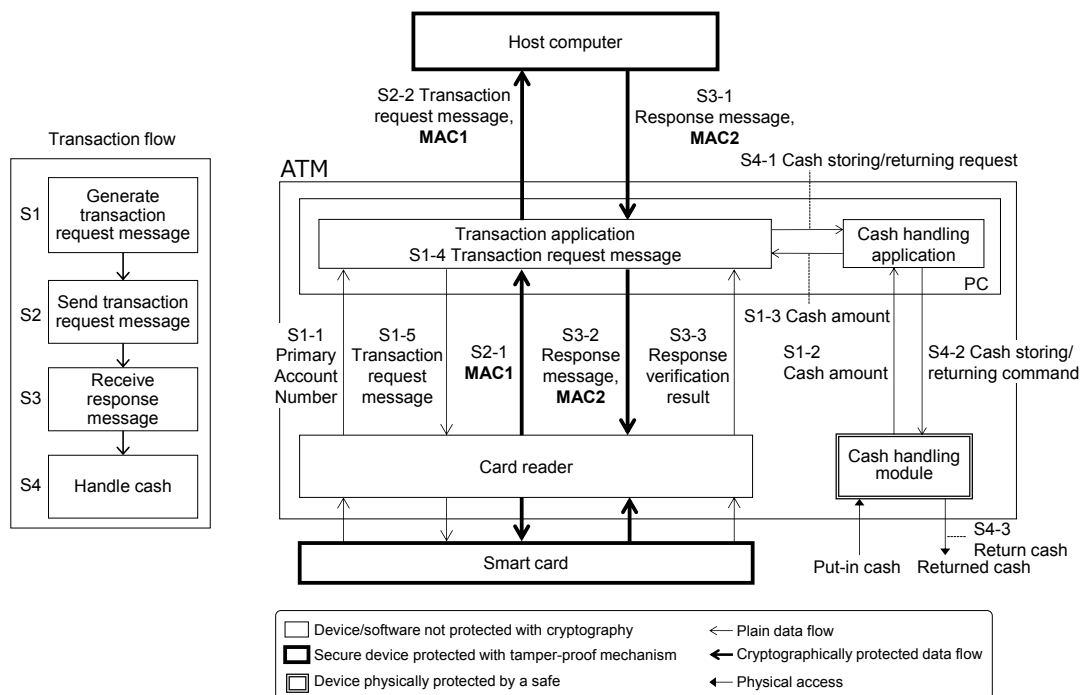


Figure 4.7 Existing deposit transaction with a smart card

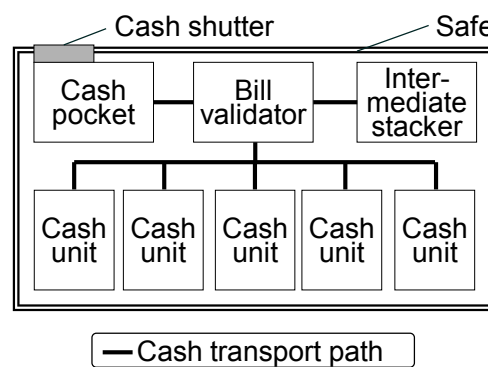


Figure 4.8 Mechanical structure of cash handling module

S2: Send the transaction request message

The smart card generates a MAC to the message called “MAC1”, and sends the S2-21 MAC1 back to the transaction application. The transaction application sends the S2-2 transaction request message and the MAC1 to the host computer. The host

computer verifies the transaction request message with the MAC1, and then decides whether the transaction is authorized or not.

S3: Receive a request message

The host computer creates a response message and generates a MAC to the message called “MAC2”, and then sends the S3-1 response message and the MAC2 back to the transaction application. The application forwards S3-2 data to the smart card through the card reader. The smart card verifies the received message with the MAC2 and returns the S3-3 response verification result to the transaction application. The value of the verification result varies in accordance with the host computer’s decision.

S4: Handle cash

The transaction application provides the cash handling application with a S4-1 cash storing request if the transaction is authorized in the response verification result. Otherwise the application provides with a S4-1 cash returning request. The cash handling application sends either a S4-2 cash storing command or a cash returning command to the cash handling module. The cash handling module transports the cash in the intermediate stacker into the cash units to store the cash in the safe if the cash handling module receives the cash storing command as shown in Figure 4.8. Or the cash handling module transports cash in the stacker back into the cash pocket to return the cash to the ATM user (S4-3).

An application of Command Verification to S1 transaction sub-process and S4 transaction sub-process [43] is outlined in Figure 4.9. A tamper-proof secure element is implemented in the proposed card reader, which provides three functions: verifying a transaction request message, extracting verification information such as an authorization/rejection flag and a reference time, and cryptographic communication between the card reader and the cash handling module. A tamper-proof secure element is also implemented in the proposed cash handling module, which supports three functions: extracting a cash amount, verifying cash storing/returning command, and cryptographic communication between the card reader and the cash handling module. Data Transfer Library (DTL) is also introduced to simply provide a communication path between those devices to transfer encrypted data. The cryptographic key management and a session creation for encrypted communication are supposed to conform to the international standards. A session is supposed to have been

request sent from the transaction application. When the DTL receives either command, the DTL requests the card reader to send the S4-3 encrypted authorization/rejection flag, and the message receiving time as the reference time and then forwards them to the cash handling module. The message receiving time is used to verify whether the command transferring is delayed or not. The cash handling module decrypts the S4-3 encrypted data and verifies the S4-2 cash storing/returning command with the decrypted data. That is, the cash handling module verifies whether a cash storing command has been received when S4-3 data has an authorization flag, or whether a cash returning command has been received when S4-3 data has a rejection flag. If the validity is successfully verified, the cash handling module executes the received command.

(c) Application to all-transaction sub-processes

An application to a cash withdrawal transaction with a magnetic stripe is described as an application to all transaction sub-processes. Figure 4.10 depicts a data flow example of the existing transaction, and Figure 4.11 illustrates an example of the applied system of Command Verification and the related the data flow. It is supposed that the multi-vendor application includes “transaction application” processing transaction messages and “cash dispensing application” controlling the dispenser. The detailed processes of the four transaction sub-processes of the existing transaction are shown below:

S1: Generating a transaction request message

The transaction application receives an S1-1 Primary Account Number (PAN) stored on a magnetic stripe card from the card reader, an S1-2 withdrawal amount from the touch screen, and an S1-3 encrypted PIN from the encrypting PIN pad. And then, the application generates a S1-4 transaction request message from the PAN and the withdrawal amount.

S2: Send the transaction request message

The transaction application sends the S2-1 transaction request message and the S2-2 encrypted PIN to the host computer. When the host computer receives them, the hardware security module verifies the encrypted PIN. And then, the host computer decides whether authorizes the transaction or not by confirming the ATM user’s account balance.

S3: Receive a request message

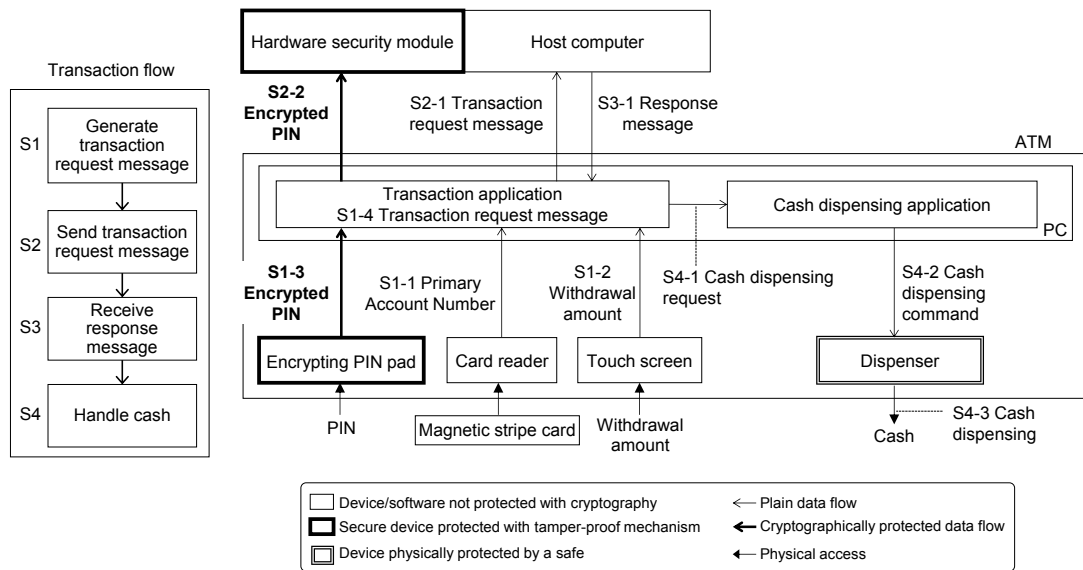


Figure 4.10 Existing cash withdrawal transaction with a magnetic stripe card

The host computer sends a S3-1 response message including the host authorization/rejection flag which indicates the host computer's decision, back to the transaction application.

S4: Handle cash

The transaction application provides the cash dispensing application with an S4-1 cash dispensing request in accordance with the host authorization flag. The cash dispensing application sends an S4-2 cash dispensing command to the dispenser. The dispenser dispenses cash according to the command.

An applied system of Command Verification to all transaction sub-process [44] [45] is outlined below. The system related with a PIN is omitted. A tamper-proof secure element is implemented in the proposed card reader, which is equipped with three functions: verifying a transaction request message, extracting verification information such as an authorization/rejection flag and a reference time, and cryptographic communication with the encrypting PIN pad and the cash handling module. A tamper-proof secure element is also implemented in the proposed cash handling module, which supports three functions: extracting a cash amount, verifying cash storing/returning command, and cryptographic communication with the cash handling module. Data Transfer Library (DTL) is also introduced to simply provide a communication path between those devices to transfer encrypted data. The

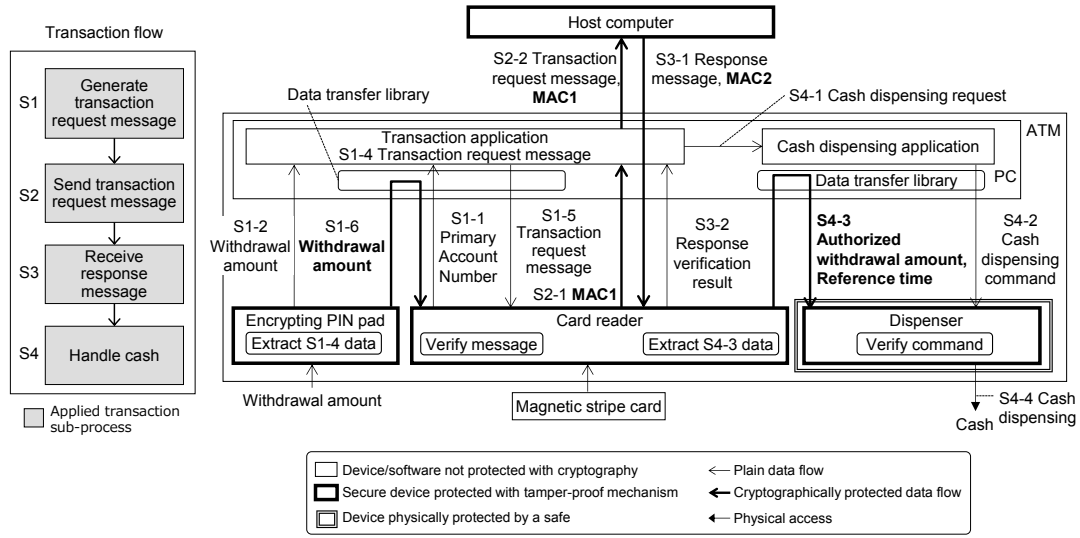


Figure 4.11 Application to all transaction sub-processes

cryptographic key management and a session creation for encrypted communication are supposed to conform to the international standards. A session is supposed to have been preliminarily created. The detailed process flows are described as follows. Only modified processes are explained here.

S1: Generating a transaction request message

The proposed card reader sends an S1-1 Primary Account Number to the transaction application, and stores it in the secure element. The proposed encrypting PIN pad instead of the touch screen sends an S1-2 withdrawal amount to the transaction application, and stores the amount in it. The transaction application sends an S1-3 transaction request message to the card reader through DTL in order to make it generate a MAC1 for the message. When the DTL receives the message, the DTL requests the encrypting PIN pad to send the S1-4 withdrawal amount in an encrypted form and forwards it to the card reader. The card reader verifies the message with the Primary Account Number stored in the secure element and the S1-4 withdrawal amount.

S2: Send the transaction request message

The card reader generates an S2-1 MAC1 for the verified message and sends it to the transaction application. The card reader also stores the withdrawal amount in the secure element. The transaction application sends the S2-2 request message

and the MAC1 to the host computer, and then the host computer verifies the received message.

S3: Receive a request message

The host computer generates an S3-1 reply message including a host authorization/rejection flag and a MAC2 for the message and sends them back to card reader through the transaction application. When the card reader receives them, it stores the message receiving time as the reference time. The card reader verifies the message with the MAC2 and returns the S3-2 response verification result to the transaction application. The card reader also generates an authorized withdrawal amount with the response verification result and the withdrawal amount stored in the secure element.

S4: Handle cash

The transaction application provides the cash dispensing application with an S4-1 cash dispensing request, and the cash dispensing application sends an S4-2 cash dispensing command to the dispenser through the DTL. The DTL requests the card reader to send the S4-3 authorized withdrawal amount and the reference time in an encrypted form and then forwards them to the dispenser. The dispenser receives the command and the S4-3 data and calculates the command transfer time with the reference time. And then the dispenser verifies the command with the authorized withdrawal amount to confirm whether the dispensing amount in the command is identical to the authorized withdrawal amount. The dispenser also verifies the command transfer time to confirm whether the transfer time exceeds a predetermined threshold. The transfer time that exceeds the threshold suggests that the command may be maliciously delayed to make false trouble in order to let the user leave the ATM for stealing the cash from the ATM. If they are successfully verified, the dispenser dispenses cash.

Chapter 5 Application of Command Verification to One Transaction Sub-process

5.1 Introduction

Recently, criminals frequently utilize logical attacks for the sake of unauthorized cash withdrawal from ATMs. Typical logical attacks are so-called “Jackpotting” [1] [2] [4] [5] [6] [7] and “Black Boxing” [1] [8] [9] which is a variant of Jackpotting” In general, an ATM consists of a PC running the Windows Operating System (OS) and peripheral devices such as a card reader and a dispenser. The ATM platform provides financial institutions’ multi-vendor application on the PC with standardized Application Programming Interfaces (APIs) [20] to control peripheral devices. As the APIs’ specifications are open to the public and the API’s are not cryptographically protected, malware frequently utilizes the ATM platform for unauthorized cash dispensing.

The existing security guidance [1] [3] [10] [11] [12] [13] primarily try to protect information property in the PC and the communication cable between the PC and the dispenser to prevent those logical attacks. However, there is an issue that those measures could be bypassed or disabled by criminals since frequent physical/logical access inside ATMs are required in existing ATM operations. For example, periodical cash replenishment and collection for cash services once a few days to a week, and quarterly periodical software/content updating for better services. ATM management costs could increase if the integrity of executable files is assured by tight ATM operational management to cope with that issue. Furthermore, it is difficult to secure integrity by limited human resources 24 hours 7days when a financial institution operates more than ten thousand ATMs.

To solve the issues, we propose an ATM security measure called “Command Verification” in section 4.1, in which controlled peripheral devices themselves verify commands sent from the PC before executing the commands to access the property. In this chapter, we qualitatively compare Command Verification and existing guidance described in section 3.3 in an application of Command Verification to one transaction sub-process of a cash withdrawal transaction with a smart card [37]. Three conditions

to effectively prevent Jackpotting without imposing a heavy burden on financial institutions to tightly protect the PCs are extracted from analyses of issues regarding existing ATM systems and operations. Command Verification and the existing guidance that is described in section 3.3 are compared from the viewpoint of conformity with the three conditions. The EUROPOL's guidance [1] is selected as a representative of existing measures. It is shown that Command Verification meets the conditions, although the existing guidance does not meet them.

Section 5.2 addresses the issues of existing ATM systems and operations, and conditions that a security measure can effectively protect cash in an ATM from Jackpotting. Section 5.3 presents the comparison result of Command Verification and the existing guidance to evaluate the effect of Command Verification. Section 5.4 is a discussion.

5.2 Issues of Existing ATM Systems and Operations

5.2.1 Existing Cash Withdrawal Transaction with a smart card

Figure 5.1 outlines an example of an existing cash withdrawal transaction with a smart card, which is the same as Figure 4.5. The messages transferred between a smart card and the host computer are cryptographically protected with Message Authentication Codes (MACs) and a smart card and the host computer conforming to the EMV specifications [21] [22]. The brief summary of the EMV specifications is as follows.

A smart card and the cryptographic processing module of the host computer must be a tamper-proof secure device. A message and the corresponding MAC to verify the authenticity of the message are transferred between the smart card and the host computer. The smart card generates a MAC to a transaction request message that is created by the transaction application and also verifies a MAC to a response message received from the host computer. A master key to generate a session key for a MAC has been installed in a smart card and the host computer through a card personalization process before issuing the card. The master key is linked with the card holder's

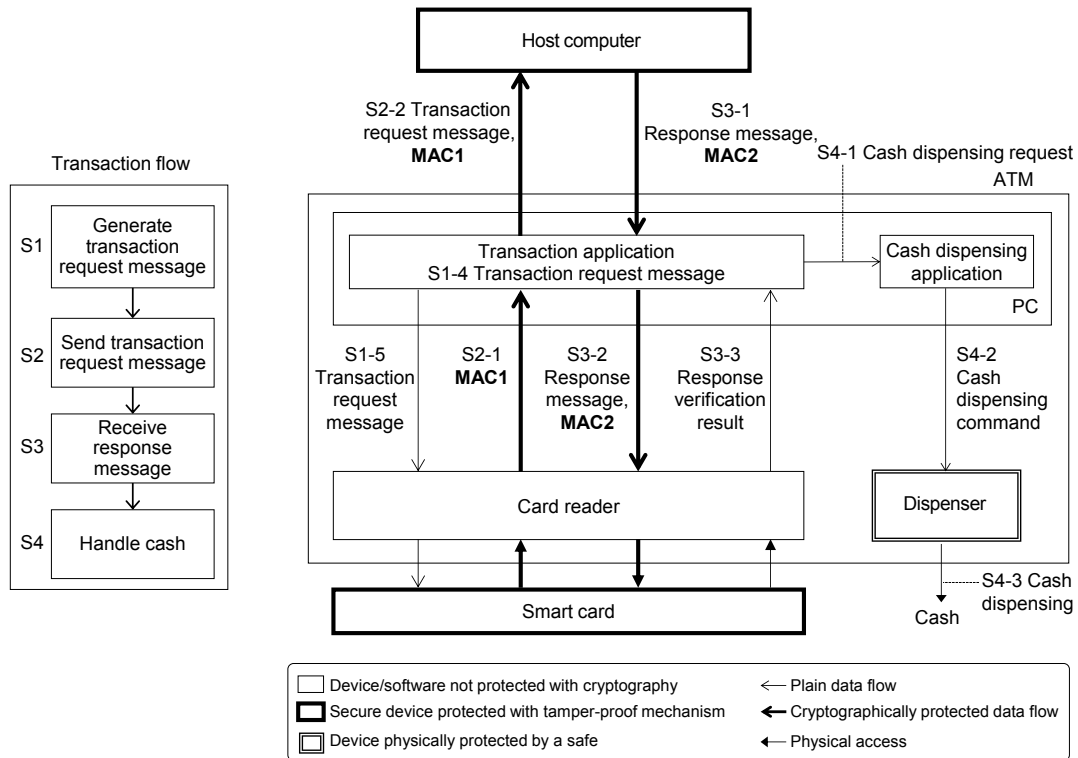


Figure 5.1 Existing cash withdrawal transaction with a smart card

Primary Account Number (PAN). A unique session key for each transaction is generated from the master key and a transaction counter output from the smart card according to the EMV specification [21]. The host computer also shares the same session key conforming to the specification.

The system is vulnerable other than the communication between a smart card and the host computer, namely, S2 and S3 in the figure. The dispenser is secure against unauthorized physical manipulation because it is physically protected by a safe. The ATM platform and the OS are omitted in the figure.

5.2.2 Issues of ATM Systems and Operations

As explained in section 5.2.1, devices, software, data other than S2-2, S2-3, and S3-1 are vulnerable. Hence criminals can perpetrate Jackpotting by attacking S4-1 and the cash dispensing application, and Black Boxing by attacking S4-2. The existing measures [1] [3] [10] [11] [12] [13] try to protect information property in the PC against Jackpotting and try to cryptographically protect the communication cable for S4-2

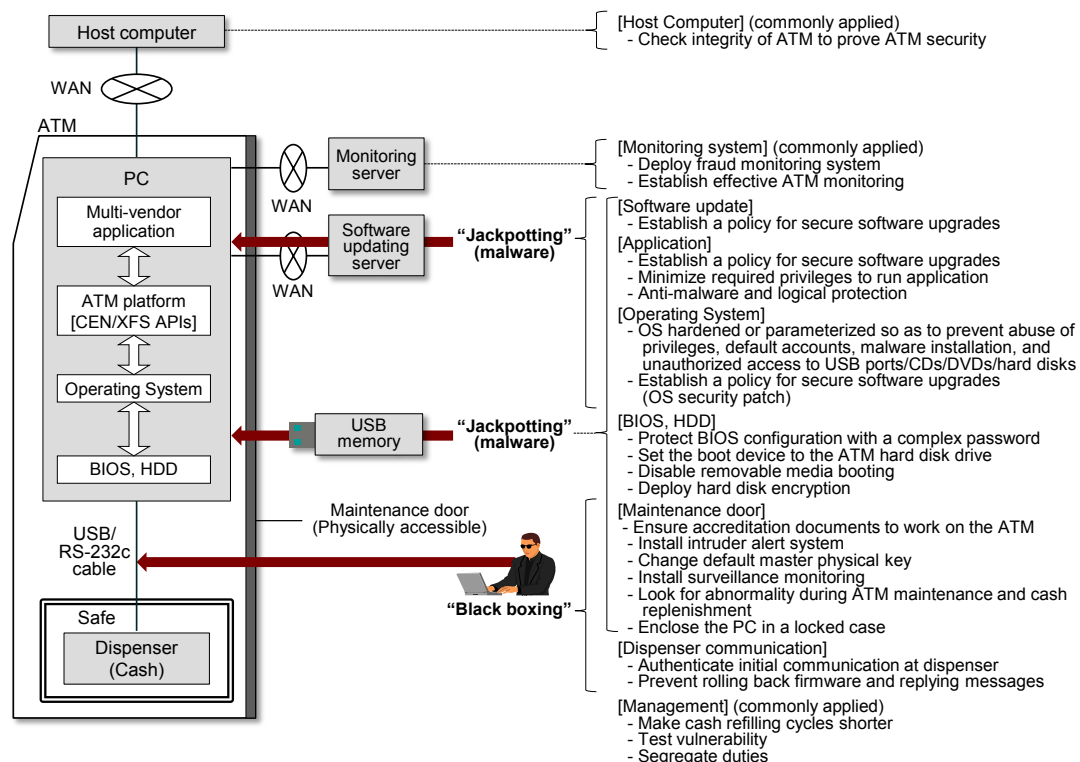


Figure 5.2 An outline of the existing guidance

against Black Boxing as shown in Figure 5.2. The communication cable's protection also relies on protecting information property in the PC and the peripheral devices, which include cryptographic keys and cryptographic processing modules. As a result, the whole ATM, which consists of the PC, the peripheral devices, and the communication cable between the PC and the peripheral device, should be tightly protected in the existing guidance. However, such measures are not so effective or efficient by virtue of the following situations of existing ATM systems and operations. The most critical issue is the PC's protection.

System aspects:

(1) Vulnerable CEN/XFS APIs

The primary specifications of CEN/XFS APIs have established in the 90s and their security functionality is rather poor than that of the current standards. Hence malware frequently utilizes CEN/XFS APIs for Jackpotting. Many financial institution's multi-vendor applications have been developed on CEN/XFS APIs. Even though secure CEN/XFS APIs are newly developed, it may take a long time to

make such secure APIs common since a lifetime of ATMs is usually 7 to 10 years.

(2) Complicated logical structures of ATM software

Financial institutions must provide ATM users with various kinds of ATM services, e.g. not only transactions within the financial institution but also transactions with other financial institutions. Logical structures and data processing of ATM software is very complicated and there are so many software components more than twenty thousand in each ATM. Even a tiny change of ATM software may bring a serious ATM system trouble in some cases because it is quite difficult to completely confirm all software components and configurations in an ATM before releasing it. It is further difficult to completely confirm all software components in cases of OS updating for security patch and OS hardening since an OS is the base of all layers above.

Operational aspects:

(3) Frequent physical access to the inside of ATMs

Frequent physical access to the inside of ATMs is required according to the reasons listed below.

- Replenishment of bills in the dispenser
- Replenishment of receipt paper sheets
- Periodical cleaning of bill dust in the dispenser
- Removal of bill jam for troubleshooting
- Replacement of parts for troubleshooting
- Off-line system updating and log data collection due to poor network performance in some cases.

Accessing the PC is also allowed during such physical access to the inside of ATMs.

(4) Frequent and unprotected system updating

Financial institutions frequently must update ATM systems in order to improve services, to update advertisement contents, to patch the OS and so forth. System updating is not tightly controlled if it is not covered by the EMV specifications and the PCI requirements. Tightly controlled system updating may impact timely launching services as it would take a very long time to completely confirm the integrity and compatibility of all software in an ATM.

5.2.3 Conditions to Effectively Prevent Jackpotting

Tightly protecting the whole ATM is not so practical due to the situation of existing ATM systems and operations as described above. The most critical issue is protecting the PC containing a lot of information property in the ATM. Therefore, we focus on preventing Jackpotting compromising the PC and a security measure should satisfy the following three conditions to cope with the existing situations.

(A) A security measure should not significantly impact management workloads of existing ATM operations.

The existing guidance recommends tight access controls to the PC in each ATM with a unique login password, a unique physical lock, and additional tight management measures. On the other hand, frequent physical and logical access to the inside of several thousands of ATMs is required during ATM operations in some cases. Such tight and frequent access controls to ATMs may result in a heavy burden to manage so many ATMs. One idea is to enclose the PC with a tamper-proof box to protect it from unauthorized physical accessing; however, it would take a long turnaround time to fix the PC for troubleshooting. As one of the most breakable devices in an ATM is a hard disk drive in the PC, such a long turnaround time could not be accepted for financial institutions. Consequently, the condition (A) is required.

(B) A security measure should not significantly impact ATM system availability.

The existing guidance has required financial institutions to update and harden the OS of ATMs in the aim of the patch for the vulnerability. However, financial institutions may hesitate to conduct them since occasional ATM system troubles accompanying OS updating is not allowed as a social infrastructure. As an OS is the base of all software layered above, it is quite difficult for financial institutions to comprehensively test the compatibilities of so many software components within a limited time to keep the OS up to date. Consequently, the condition (B) is required.

(C) Jackpotting cannot be successful even though the integrity of all software related to dispensing commands is not assured.

Taking into consideration the conditions explained above, it is quite difficult to completely assure the integrity of all software and data on the PC related with cash

dispensing commands in existing ATM operations. Furthermore, as the primary objectives of vulnerable CEN/XFS APIs are interoperability and compatibility for multi-vendor applications, it is not practical to drastically change the APIs specifications for a security objective. Different approaches are needed to cope with the situation. Consequently, the condition (C) is required.

The eventual objective of the existing guidance to prevent Jackpotting is to tightly protect information property in the PC. It is obvious that those requirements do not meet the conditions (A) to (C).

5.3 Application of Command Verification to transaction sub-process handling cash

5.3.1 Implementation Idea of Command Verification

The primary idea of Command Verification Implementation is shown in Figure 5.3. The existing dispenser does not have any information to verify a cash dispensing command. The encrypted data flow, which corresponds to the verification information in the primary model, S4-3 authorized withdrawal amount is newly introduced between the card reader and the dispenser so that the dispenser can get the information to verify a cash dispensing command. Two secure peripheral devices are proposed for implementation of Command Verification. A proposed card reader, which corresponds to the information acquiring device in the primary model, extracts an authorized withdrawal amount from the S1-5 transaction request message and S3-3 response verification result that flow in the card reader. A proposed dispenser, which corresponds to the verified command executing device in the primary model, verifies a cash dispensing command with the S4-3 authorized withdrawal amount.

The authenticity of the command is confirmed with two kinds of conditions; approval of the cash withdrawal transaction by the host computer including the withdrawal amount and the proved identity for the transaction. The two conditions are assured only by the S4-3 authorized withdrawal amount according to the mechanism explained below. Regarding the first condition, the proposed card reader can extract

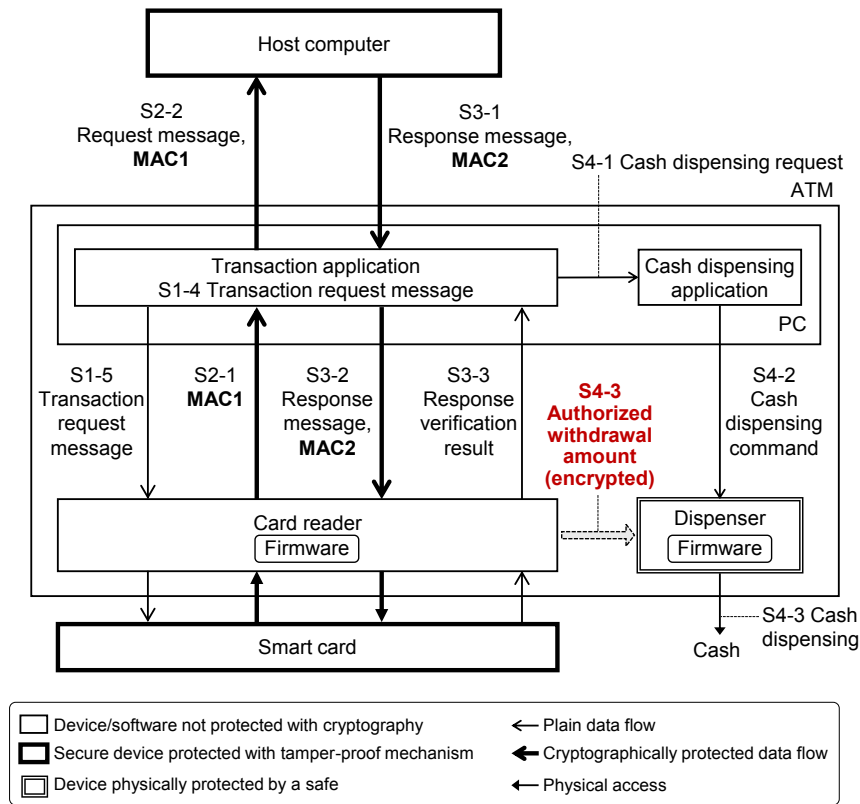


Figure 5.3 Implementation idea of Command Verification

the withdrawal amount from the S2-1 transaction request message, and extract whether the amount is authorized or not from the S3-3 response verification result in order to generate the S4-3. Although the S3-3 is plain data, the proposed card reader can receive the valid S3-3 data since the card reader can receive it as soon as the smart card outputs the S3-3 before malware, if any, in the PC received it. Concerning the second condition, the card reader's receiving S3-3 suggests that the identity is successfully proved in the S3-1 response message. It is assured because the host computer sends the S3-1 indicating the authorized response only when the PIN is successfully verified.

The withdrawal amount and the Primary Account Number in the S2-1 request message can be altered by malware in the proposed idea; nevertheless, the protection priority is low due to the reasons described below. As an altered withdrawal amount in the S2-1 goes directly to the altered amount of cash dispensed to the ATM user, either the user or the financial institution does not suffer any monetary loss. If the Primary Account Number in the S2-1 is altered by malware, it becomes inconsistent with the

session key and the master key in the smart card since the Primary Account Number is linked with those keys as explained in section 5.2.1. In this way, Command Verification can effectively and efficiently prevent unauthorized cash dispensing by protecting only the card reader, the dispenser and the S4-3. It is a contrast to the existing guidance that tries to tightly protect the whole ATM, which is not so practical as described in section 5.2.2.

As an alternative implementation of the proposed measure, the proposed card reader can send an encrypted cash dispensing command to the dispenser instead of the S4-3. It does not work according to the existing ATM services. Some users select denominations of dispensed bills on the screen before cash dispensing. Therefore, the proposed card reader must support such an application and must control the Graphical User Interface on the screen as a substitute for the dispensing application. Furthermore, a card reader constantly needs to know the state of the PC to control the Graphical User Interface. It is not practical from a viewpoint of the card reader's hardware resource and cost.

5.3.2 Implementation of Command Verification

An implementation example of Command Verification is outlined in Figure 5.4. There is not any physical communication cable between the existing card reader and the existing dispenser. Therefore, encrypted communication to transfer the authorized withdrawal amount is implemented through the PC and the existing communication cables between the PC and each peripheral device. Data Transfer Library (DTL) is newly introduced to simply provide a communication path between those devices. DTL is supposed to be installed in a layer below the CEN/XFS APIs, namely below the applications.

A secure element providing functions related to Command Verification is implemented in the proposed card reader and dispenser. A secure element is a tamper-proof hardware environment that can securely host applications and encrypted sensitive data. An example of a secure element is shown in Figure 5.4, which is a one-chip secure microcontroller on a smart card. The secure element is supposed to be programmable with a Software Development Kit such as the JAVA card™³ platform. Each secure element is equipped with a device-specific function and common functions. The device-specific function of the secure element in the proposed card reader is to

³ Java and Java Card are registered trademarks of Oracle and/or its affiliates.

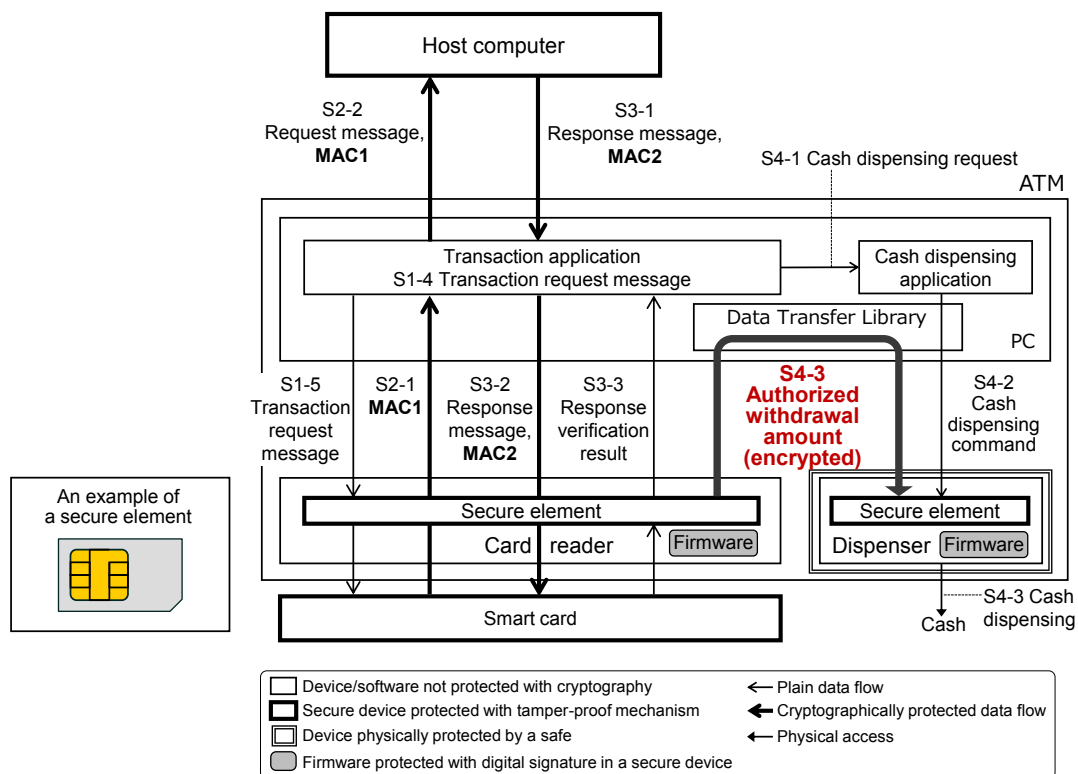


Figure 5.4 Implementation of Command Verification to one transaction sub-process

extract verification information: an authorized withdrawal amount, from transaction messages transferred between a smart card and the host computer. The device-specific function of the secure element in the proposed dispenser is to verify a cash dispensing command with the verification information.

There are two common functions; one is cryptographic functions for encrypted communication between the proposed card reader and the proposed dispenser, the other is to provide authenticity of the firmware in each peripheral device with the digital signature installed in the secure element in order to protect the firmware from unauthorized manipulation. The firmware running on the RAM in each peripheral device is still secure by self-tests with the digital signatures. For example, the firmware hash is calculated periodically such as once every day in the controller. The hash is transferred to the secure element and is verified with the digital signatures. It is noted that the digital signatures provide not the integrity but the authenticity of the firmware since the implementer's identity should be proved with the digital signature according to the PCI PTS POI requirements.

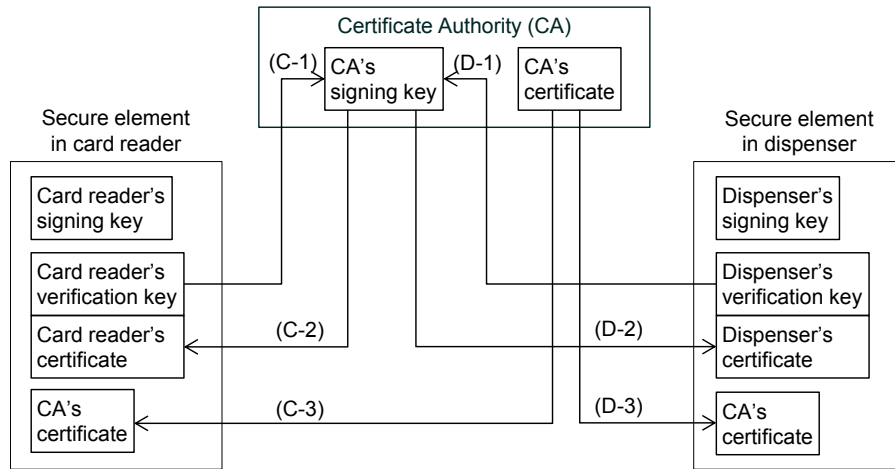


Figure 5.5 Installation of Certificate Authority's certificate

The cryptographic key management and a session creation for the encrypted communication in the secure elements are supposed to conform to the international standards [14] [15] [19] [30] [38] [39] [40] [41] [42]. The cryptographic communication in an ATM should also meet the following conditions from the two viewpoints of operation and technology. Since one of the objectives to implement cryptographic communication is to prevent internal crime by staff, not common keys but unique keys should be used assuming that maintenance parts including cryptographic keys may be abused by staff. If a common key is used, there is a risk that all parts including the key are compromised when the key is leaked. As we suppose internal crime by staff in ATM operation, the device may have been manipulated while the ATM was down. Thus a device needs to properly authenticate the communication partner with not a secret key but a public key when the ATM is booted because the partner may not be trusted at the ATM booting. Although a cryptographic session creation with a public key takes a much longer time than a common key, it is acceptable since a session is created only once when the ATM is booted. On the other hand, real-time responses are required for cryptographic communication in ATM transactions, and it is necessary to shorten the processing time and to perform authentication with a common key such as a message authentication code.

As an example to meet the conditions, an example to create a cryptographic session with Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Diffie–Hellman key exchange (ECDH) is illustrated in Figure 5.5 to Figure 5.8.

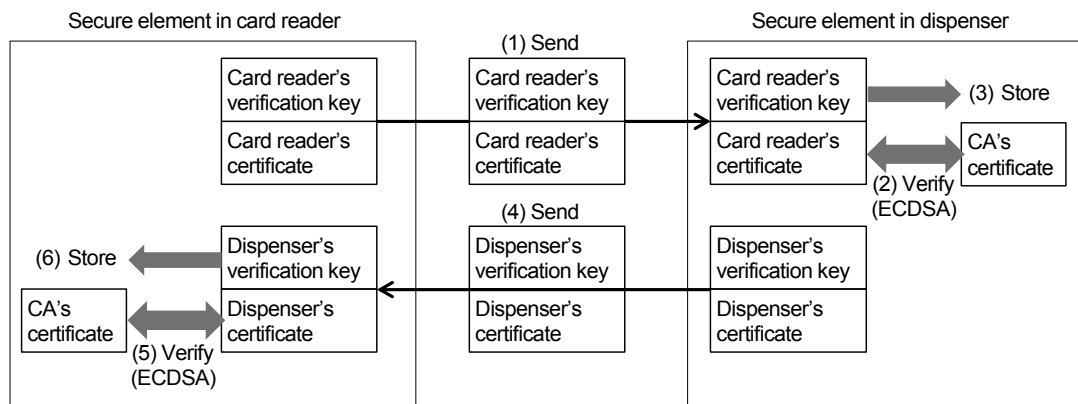


Figure 5.6 Signature verification with ECDSA

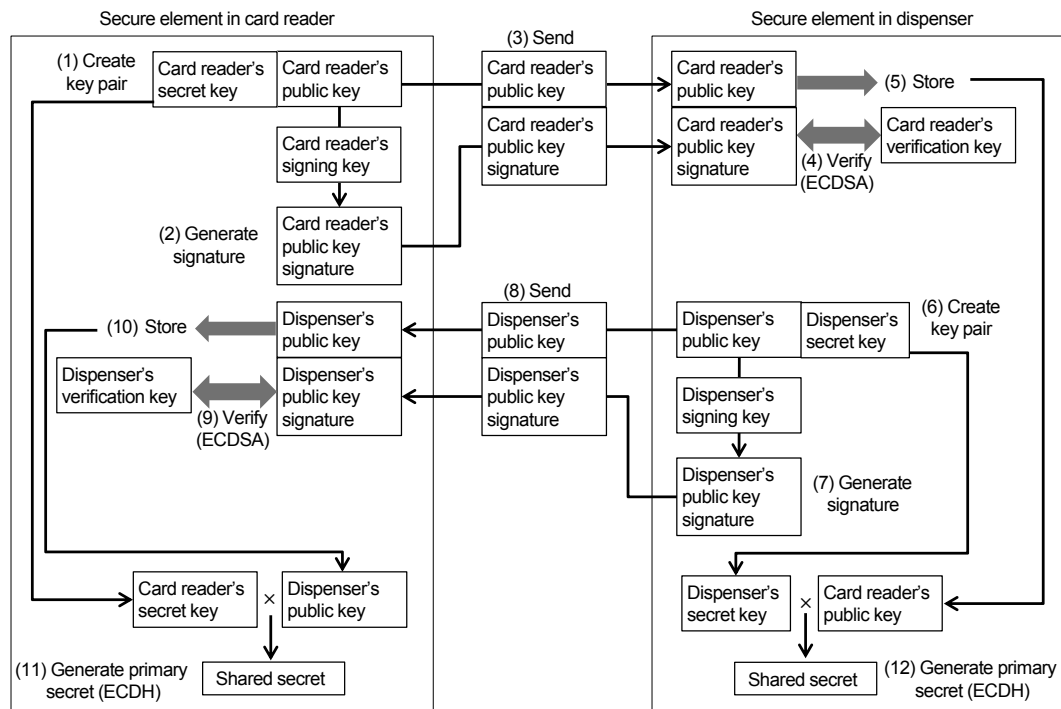


Figure 5.7 Key exchange with ECDH

Processing time of asymmetric encryption such as ECDSA is much longer than symmetric encryption such as the Advanced Encryption Standard (AES) algorithm. Asymmetric encryption taking a long time is used in mutual authentication to create a cryptographic session, which is conducted in the initialization process of ATM booting.

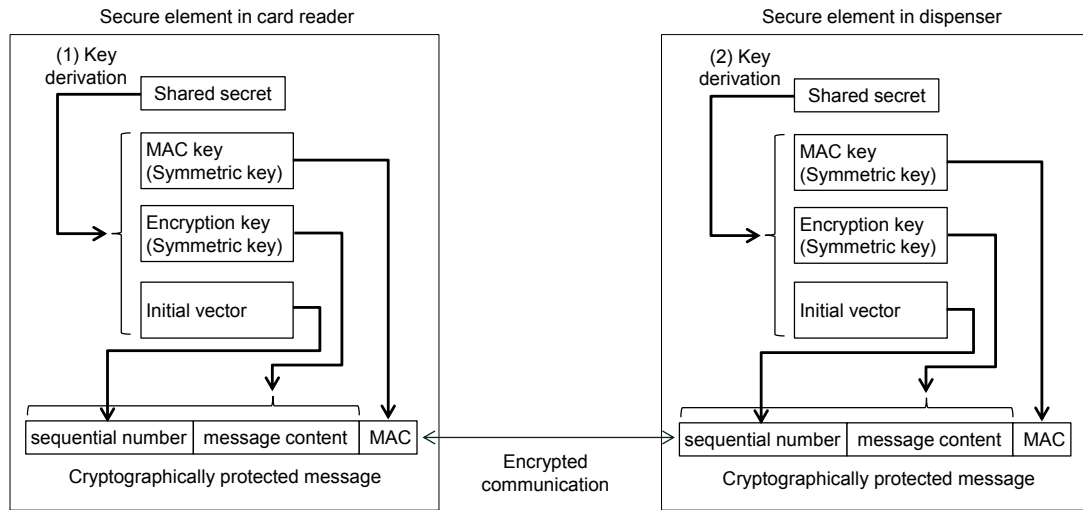


Figure 5.8 Key derivation for encrypted communication

On the other hand, symmetric encryption taking a short time is used in encrypted messages of the session for a fast turnaround.

In Figure 5.5, a Certification Authority (CA) generates a certificate for a verification key in each secure element, and the certificate and the CA's certificate are installed in each secure element. This installation must be conducted beforehand in a secure room protected with tight access control. To create a session of encrypted communication, the certificate for the verification key in the card reader's secure element is verified to be accepted by the dispenser's secure element according to step from (1) to (6) in Figure 5.6. In Figure 5.7, each secure element creates a pair of a temporary private key and a public key and generates a signature for the public key with the signing key in the secure element following step (1) (2). The dispenser's secure element verifies the public key sent from the card reader with the attached signature and the card reader's verification key to accept the public key in accordance with step (3) to (5). The accepted public key and the temporary private key in the dispenser's secure element are used to generate a shared secret to be shared with the card reader's secure element according to step (12). The same shared secret is generated in the card reader's secure element following step (8) to (11). After that, symmetric keys and an initial vector are derived from the shared secret for encrypted communication as depicted in Figure 5.8. Refer to section 4.2 regarding the detailed processes of Figure 5.4.

The structure examples of the proposed peripheral devices are depicted in Figure 5.9. In general, an existing card reader is equipped with a slot to install a secure

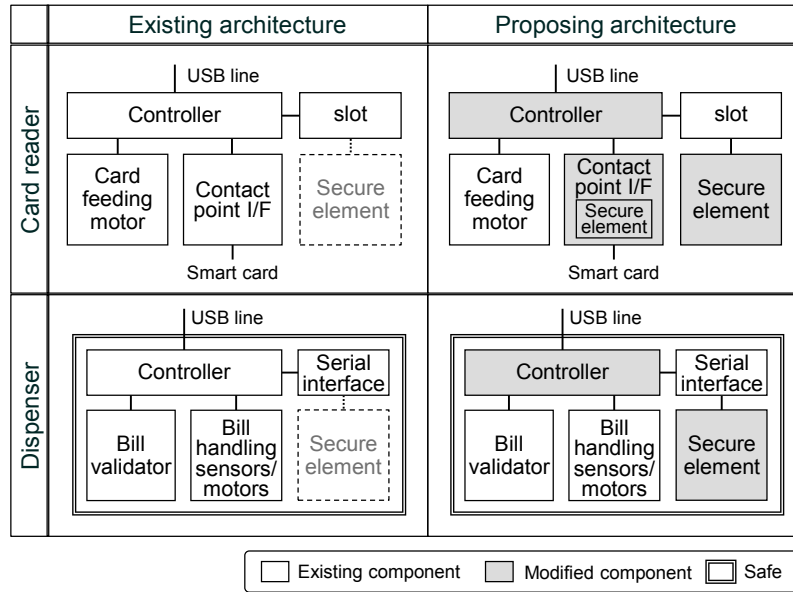


Figure 5.9 Comparison of existing devices and proposing devices

element for mutual authentication between a smart card and a terminal. The secure element in Figure 5.4 can be installed in the slot. The contact point interface to communicate with a smart card is equipped with another secure element. The secure element is cryptographically connected with the secure element installed to the slot in order to protect contents from/to a smart card from unauthorized access to the inside of the card reader. Such a structure is practical since the PCI requirements [14] [30] define similar requirements. Concerning a dispenser, an existing dispenser is equipped with a serial interface to expand the functions in many cases. A circuit board implementing a secure element can be installed on the serial interface. The firmware in the controller is also supposed to be protected from unauthorized manipulation with digital signatures installed in the secure element as well. The firmware running on the RAM in the controller is supposed to be still secure by self-tests as well as the proposed card reader. Furthermore, the whole dispenser is protected from unauthorized physical access by a tightly controlled safe. Thus the firmware is logically and physically protected.

Regarding a development cost of the implementation, there are three development items: (a) DTL in the PC, (b) modification of existing firmware of the card reader and the contact point interface implementing a secure element, and a secure element installed in the slot in the card reader, (c) modification of existing firmware of the dispenser and a circuit board implementing a secure element in the dispenser. Some

country's regulations require a similar implementation to (a) and (c) in ATMs for other security objectives. Concerning (b), existing device vendors provide card readers equipped with similar components and structures to protect cardholder data in conformity to PCI PTS POI [14]. Thus items (a) (b) (c) can be developed based on existing implementation and components at a reasonable cost.

5.3.3 Evaluation of the Measure

It is described here that Command Verification can meet the conditions addressed in section 5.2.3 using the implemented system illustrated in section 5.3.2.

(A) A security measure should not significantly impact management workloads of existing ATM operations.

Command Verification can prevent invalid cash dispensing using the proposed peripheral devices equipped with a tamper-proof secure element. Therefore, quite heavy management workloads to tightly protect the PCs are not required.

(B) A security measure should not significantly impact ATM system availability.

Command Verification does not rely on the tightly protected PC but on the proposed peripheral devices. Frequent OS updating/hardening for a security patch, which would significantly impact ATM system availability, is not a necessary condition in the implemented system. Financial institutions can take enough time to comprehensively test so many software components in the PC before releasing them to prevent occasional system troubles.

(C) Jackpotting cannot be successful even though the integrity of all software related to dispensing commands is not assured.

Command Verification can prevent Jackpotting without relying on the integrity of all software of the PC. Even if the integrity of DTL is not assured, Jackpotting cannot still be successful as DTL is just a communication pass to transfer encrypted data. Command Verification can work as a defense in depth in cases that the PC is compromised.

In this way, Command Verification can harmonize with existing ATM systems and operations by meeting the three conditions. The recommendations of the existing guidance do not meet the three conditions as described in section 5.2.3. The comparison between the existing guidance and Command Verification is summarized in Table 5.1. As far as the authors investigated the existing security guidelines of other

Table 5.1 Comparison of the existing requirements and the proposed measure

Conditions	Existing Guidance	Command Verification
Condition (A)	Not satisfied	Satisfied
Condition (B)	Not satisfied	Satisfied
Condition (C)	Not satisfied	Satisfied

countries and ATM vendors, security company's solutions and patents, there are neither methods nor solutions meeting the conditions (A) to (C). It is noted that the implemented system can also prevent Black Boxing since the proposed dispenser does not accept invalid dispensing commands received from an external computer as it does not receive any authorized withdrawal amounts.

We developed a prototype system of Command Verification with an existing ATM system to confirm the operational feasibility. Circuit boards equipped with a Java Card-based secure element were implemented into an existing card reader and an existing dispenser. We confirmed that the dispenser dispensed cash when the dispensing amount in a command received from the PC and the authorized withdrawal amount received from the card reader are identical. We also confirmed that the dispenser did not accept dispensing commands without any authorized withdrawal amount or with an altered withdrawal amount.

5.4 Discussion

In this chapter, we explained an application of Command Verification to one transaction sub-process in a cash withdrawal transaction with a smart card, namely the sub-process handling cash. We also described the detailed implementation of Command Verification regarding the card reader, the dispenser, the PC, and cryptographic communication between these peripheral devices. We qualitatively compared Command Verification and existing measures in an application of Command Verification to the one transaction sub-process. The existing guidance was selected as representative existing measures. Three conditions to effectively prevent Jackpotting without imposing a heavy burden to tightly control the PCs on financial institutions were extracted from analyses of issues regarding existing ATM systems and operations. The conditions are (A) to not significantly impact management workloads of existing ATM operations, (B) to not significantly impact ATM system availability, and (C) to

effectively prevent Jackpotting even though integrity of all software related to dispensing commands is not assured. Command Verification and the existing guidance were compared from the viewpoint of conformity with the three conditions. It was shown that Command Verification meets the conditions although the existing guidance does not meet them. In this way, all the recommendations of the existing guidance are important and costly because they are effective in total. As Command Verification protects property with the peripheral devices, there is no need to fully enforce the recommendations of the existing guidance, and the incurred management costs are reduced. Furthermore, Command Verification can work as a defense in depth when the PC is compromised.

We did not propose a measure to protect a withdrawal amount in a transaction request message sent from the PC to a smart card since any monetary loss does not occur even if the message is altered. That is, an altered withdrawal amount in the message goes directly to the altered amount of cash dispensed to the ATM user. However, some users may be embarrassed by the unexpected cash amount and the situation would bring other frauds. Hence protecting the transaction request message is also a remaining issue that will be tackled.

Chapter 6 Application of Command

Verification to Two Transaction

Sub-processes

6.1 Introduction

In this chapter, an application of Command Verification to two transaction sub-processes, issues of the application, and a solution to the issues is presented using a deposit transaction with a smart card. Recently, criminals frequently carry out logical attacks on Automated Teller Machines (ATMs) and financial Institutions' networks to steal cash. Two kinds of logical attacks to steal cash are considered; one is "unauthorized cash withdrawal" that is cash withdrawal without debiting a financial institution's account [1] [4] [5] [6] [7] [8] [9], and the other is "unauthorized deposit" that is fraudulent increase of account balance without the equivalent amount of cash. Regarding the unauthorized cash withdrawal, we analyzed issues of existing guidance, and proposed Command Verification utilizing peripheral devices to solve the issues in chapter 4 and 5. On the other hand, measures for unauthorized deposits have not been argued sufficiently. To be more specific, transaction sub-processes in a deposit transaction is manipulated using malware and malicious hardware to fraudulently increase account balance with no actual cash or less cash. And then, criminals can withdraw cash with ATMs from the accounts. Unauthorized deposit is highly likely to occur in the near future because of the following strong incentives to criminals [43].

"Easiness"; (a) few existing guidance explicitly focuses on unauthorized deposit, (b) vulnerable ATM platform can be abused, and (c) new logical attacks can be easily created by modeling for widespread existing frauds using a physical trap such as "Transaction Reversal Fraud" [46] and "Cash Trap" [47] [48], which are explained in section 3.1.

"Expandability"; criminals can also conduct fraudulent international remittance if the deposit accounts are linked with internet banking accounts. Since remittance services on ATMs are usually limited to domestic accounts, international remittance using internet banking is more preferable for criminals due to the difficulty of the trace. The

transferred money can be spent even to buy cryptocurrencies. They can unlimitedly create internet banking accounts with a large balance utilizing ATMs without strenuous efforts to search such accounts on the internet.

“Covertness”; the average amount per deposit is much higher than that per withdrawal, typically double to eight times for personal and business deposit, and even more than twenty times for business deposit in some cases [49]. This difference is because of the limit set by financial institutions; a cash deposit has no limit or the limit is much higher than a cash withdrawal limit. Thus, criminals can easily conduct unauthorized transactions with a large amount of deposit as if they are usual transactions.

In this chapter, an application of Command Verification to two transaction sub-processes in an ATM transaction is described. That is an application to a transaction sub-process before/after communication between an ATM and the host computer in a deposit transaction with a smart card. There are multiple protected properties from multiple attack surfaces in the transaction sub-processes, and constraints to harmonize with existing systems and operations. It is difficult to achieve properly implementable systems of Command Verification to meet the requirements. Therefore, an implementation model analysis is proposed to compare the features of the models in a preliminary step to achieve the proper systems [43].

Section 6.2 addresses issues of existing ATM systems, existing operations, and existing guidance. Section 6.3 presents the implementation model analysis. Section 6.4 describes proper implementations of Command Verification with the proposed analysis. Section 6.5 is a discussion.

6.2 Issues of Existing ATM Systems and Operations

6.2.1 Existing Deposit Transaction

Figure 6.1 outlines an example of data flow and processes of an existing deposit transaction with a smart card [21] [22], which are the same as Figure 4.7. See section 4.2 (b) for the detailed processing flow of the example. In smart card transactions, cryptographic authenticity of a transaction message is assured by a Message

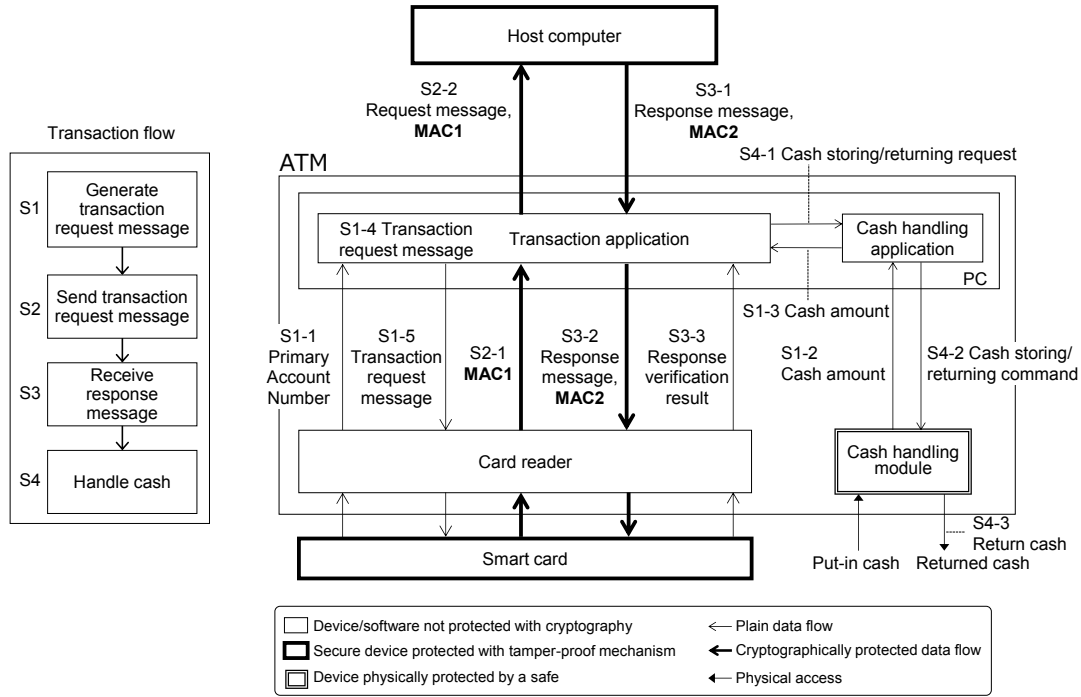


Figure 6.1 Data flow of existing deposit transaction

Authentication Codes (MACs) generated by either a smart card or the host computer, which are tamper-proof devices. A cryptographic key for MACs is shared between a smart card and the host computer in conformity to the EMV specifications [21]. The key is also linked with the Primary Account Number (PAN) on the smart card, which is preliminarily assigned to the card by the financial institution to identify the user. An altered PAN in a transaction request message can be detected in a smart card transaction according to the EMV specifications. The transaction sub-processes: S2 Sending the transaction request message, and S3 receiving a response message are protected with the EMV specification. Therefore, the remaining transaction sub-processes: S1 generating a transaction request message, and S4 Handling cash, could be targets of logical attacks.

6.2.2 Issues of Existing Security Measures

The logical attacks for unauthorized deposit target the data flow and the processes of the two transaction sub-processes: S1 generate a transaction request message, and S4 handle cash in a deposit transaction. The cash handling module is secure against

Table 6.1 Logical attacks for unauthorized deposit

No.	Sub-process	Attack objective	Attack method	Targeted property	Outline of possible logical attack
A1	Generate transaction request message	Manipulation of transaction request message	Malicious device	S1-2 Cash amount S1-5 Transaction request message	A malicious device on RS-232C/USB cable manipulates a transaction request message and a cash amount for unauthorized deposit.
A2			Malware	S1-3 Cash amount S1-4 Transaction request message	Malware manipulates a transaction request message and a cash amount for unauthorized deposit.
B1	Handle cash	Manipulation of data and processes for unauthorized cash return	Malicious device	S4-2 Cash storing/ returning command	A malicious device on RS-232C/USB cable manipulates data and a command for unauthorized cash return despite the deposit transaction was authorized.
B2			Malware	S4-1 Cash storing/ returning request	Malware forces the ATM to return cash in intermediate stacker despite the deposit transaction was authorized.
C1		Making a false trouble for fraudulent cash return	Malicious device	S4-2 Cash storing/ returning command	A cash returning command sent to the cash handling module is temporally held by a malicious device on the USB cable, and then sent to the cash handling module again after a user leaves from the ATM.
C2			Malware	S4-1 Cash storing/ returning request	A cash returning command sent to the cash handling module is temporally held by malware, and then sent to the cash handling module again after a user leaves from the ATM.

unauthorized physical manipulation because it is physically protected by a safe. A1 and A2 in Table 6.1 are attacks to manipulate a transaction request message before a smart card generates a MAC1 for the message on the USB/RS-232c cable and the PC, respectively. A manipulated transaction request message and a MAC1 are sent to the host computer in order to fraudulently increase the account balance with no actual cash or less cash. And then, malicious persons withdraw cash from the fraudulently increased account using ATMs. B1 and B2 are attacks to manipulate data flow and processes when the cash stored in the intermediate stacker is transported into the cash units of the cash handling module (Figure 4.8) after the host computer authorized the deposit transaction. These attacks force an ATM to return the cash in the intermediate stacker to a malicious user. Manipulating a transaction request message is not required in the attacks. The malicious user can increase an account balance unlimitedly by repeating a cycle of sending an invalid transaction request message to the host computer and conducting unauthorized cash return after the host computer's authorization. Such unauthorized cash return can be fulfilled by applying an existing

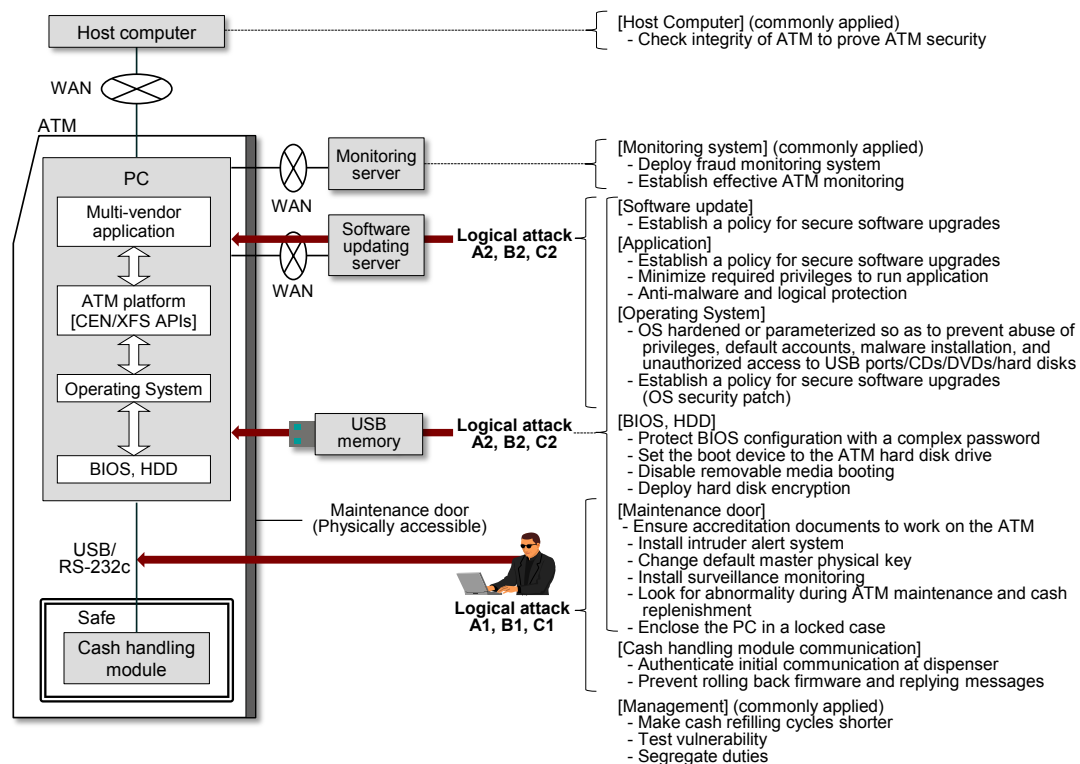


Figure 6.2 An outline of the existing guidance

fraud called “Transaction Reversal Fraud” [46], in which a criminal induces a fault at an ATM during a cash dispense operation such that the transaction application reverses the cash withdrawal transaction, i.e. retracting to debit the account, although the criminal removes dispensed cash from the ATM with some trick.

C1 and C2 are replay attacks to manipulate data flow and processes to return cash in the intermediate stacker to not a legitimate user but a malicious person when the host computer rejects the deposit transaction. The malicious person utilizes malware or a malicious device for the attacks, which temporarily holds a cash returning command transferred from the PC to the cash handling module. Such an attack causes false trouble to the ATM to make the user recognize the ATM has trouble. And then, the malicious person steals cash returned from the ATM operated by the malware or the malicious device after the user left the ATM. Such cash stealing can be conducted by applying an existing fraud called “Cash Trap” [47] [48] targeting cash withdrawal, in which a device inserted inside the cash dispenser traps cash before the cash is presented to the ATM user while fooling the user into thinking that the cash shutter has not opened. The average amount per deposit is much higher than per withdrawal

and even more than twenty times for business deposit as described in section 6.1. The replay attacks in deposit transactions can be much more effective and efficient for criminals than Cash Trap. A replay attack to S3-3 in Figure 6.1 is not supposed to bring fraudulent cash return since it results in “Out of service” of the ATM as system trouble. Thus the attack is omitted here.

As shown in Figure 6.2, the existing guidance [1] [3] [10] [11] [12] [13] [31] try to protect the PC against A2, B2, and C2. Furthermore, the existing guidance tries to cryptographically protect the USB/RS232C cables from A1, B1, and C1. However, it is difficult to prevent C1 with cryptographic communication since it is a kind of replay attack. It is noted that cryptographic protection of the USB/RS-232c cables also depends on the PC’s security measures because the cryptographic keys stored in the PC could be stolen or manipulated. In this way, the existing guidance tries to protect the whole ATM system. If the whole ATM system is protected with tight operational management, it may result in operational cost issues since financial institutions must operate many ATMs 24 hours 7days, for example, more than ten thousand ATMs in some cases.

6.3 Application of Command Verification

6.3.1 Implementation Model Analysis

The implementation model analysis of Command Verification is described here to select candidate models having preferable features, that are consistent with existing systems and operations, and that can effectively prevent the unauthorized deposit. Figure 6.3 outlines implementation models of Command Verification. Each peripheral device consists of tamper-proof hardware such as a secure element, and an existing control mechanism including firmware. The authenticity of the firmware is supposed to be assured by digital signatures stored in the tamper-proof hardware although the signatures are not shown in the figure. The verification information extracting module extracts verification information from input data and physical objects (cash) put-in the peripheral device. In model 1, the verification information is securely transferred from the information acquiring device to the verified command executing device. Command Verification module verifies a command received from the control unit and forwards the verified command to the firmware to access the property. Figure 6.3 (b) shows the model 2 that the verified command executing device in the model 1 is split into two

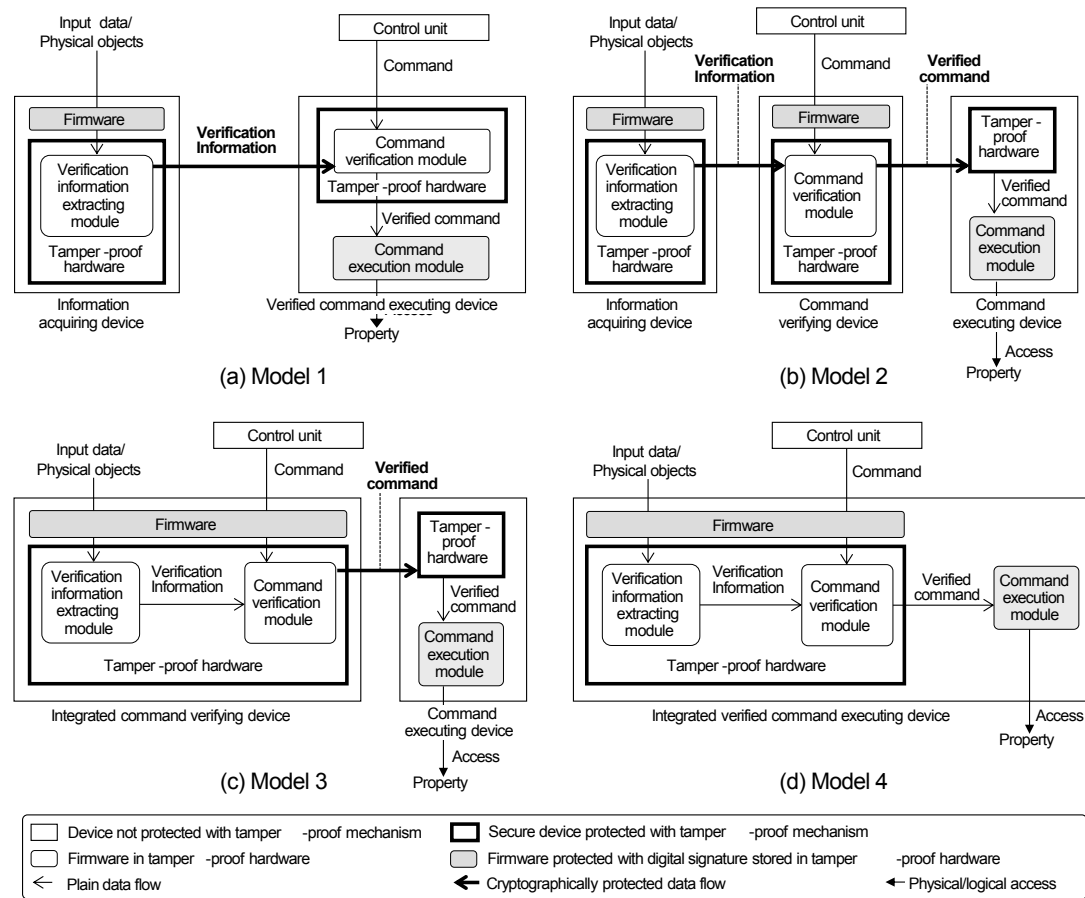


Figure 6.3 Implementation models of Command Verification

Table 6.2 Comparison of implementation models of Command Verification

No.	Features	Model 1	Model 2	Model 3	Model 4
1	Applicability to smart card transaction	Applicable	Applicable	Applicable	Not applicable
2	Cryptographic communication between peripheral devices	One	Two	One	Zero
3	Validity of command from a viewpoint of command transfer time	Verifiable	Not verifiable	Not verifiable	Verifiable
4	Validity of command except a viewpoint of command transfer time	Verifiable	Verifiable	Verifiable	Verifiable

□ Recommended implementation

peripheral devices: a command verifying device and a command executing device. Figure 6.3 (c) outlines the model 3 that the information acquiring device and the command verifying device in Figure 6.3 (b) are integrated into one device. Figure 6.3 (d) depicts the model 4 that all peripheral devices in Figure 6.3 (b) are integrated into one device.

Table 6.2 summarizes the features of each implementation model in Figure 6.3.

The condition No.1 and No.2 are derived from consistency with existing systems, and existing operations, respectively. The condition No.3 and No.4 are derived from preferable features to cover a wide range of attacks to prevent the unauthorized deposit. In conclusion, the model 1 and 3 are recommended to implement Command Verification for deposit transactions with a smart card. The preferable point of each feature in the table is explained as follows. In terms of the No.1 feature, model 4 is not preferable since it is difficult to install a smart card in a safe, which must be returned to an ATM user. When generating a MAC for a transaction request message in the model 4, the related devices: the cash handling module and a smart card must be installed in a safe because the cash is counted to generate a MAC. In terms of the No.2 feature, a smaller number is better from a viewpoint of minimizing cryptographic communication to harmonize with existing ATM operations. Cryptographic key settings for cryptographic communication could be an attack target [50] and tightly controlled key settings are required. Such key settings should be minimized in terms of work efficiency since maintenance staff may exchange a troubled part in an ATM with a service part for troubleshooting, which requires cryptographic key settings in some cases. Accordingly, model 2 is not preferable. Although the model 4 is the most preferable, the model is not applicable to smart card transactions. Thus the model 1 and 3 should be acceptable.

In terms of the No.3 feature, “verifiable” is preferable. The model 2 and 3 are not verifiable since the command executing device is not equipped with a function to verify the transfer time of the received command in these models. The function is required to detect replay attacks, i.e. whether a received command was temporarily held or not. Cryptographic protection does not work for replay attacks. In terms of the No.4 feature, “verifiable” is preferable, and all the models can verify a command except a viewpoint of command transfer time. In this way, models 1 and 3 are most and second recommended, respectively. Model 3 should remain in the candidate models since there are demands to install a critical device in the safe. Some SIers install a secure element in the card reader, inside the safe, which is wired with the card reader. The objective is to physically protect the secure element from being stolen by staff and criminals even if the device is tamper-proof. As the most important modules: the verification information extracting module and the command verification module, are installed in one device in model 3, the device can be installed in the safe.

6.3.2 Conditions to Prevent Logical Attacks

To securely protect deposit transactions from A1 to B4, the following four conditions R1 to R4 should be confirmed in secure domains of an ATM system.

- (R1) The cryptographic keys of cryptographic USB/RS-232c communication in an ATM should be protected in secure domains of an ATM.
- (R2) (Cash amount received from cash handling module) is equal to (Cash amount included in a transaction request message).
- (R3) The cash handling module accepts a cash returning command if the transaction is rejected by the host computer, otherwise, the cash handling module rejects the cash returning command.
- (R4) {(Cash handling module's receiving time of a cash returning command) - (card reader's receiving time of the response message verification result)} is less than a threshold.

R1 is an existing key management condition to protect USB/RS-232c communication from logical attacks on USB/RS-232C. R2 is the first defense point to protect a transaction request message from being manipulated before the message is cryptographically protected with MAC1. R2 is used to confirm consistency between the cash amount outputted from the cash handling module and the cash amount included in the transaction request message. R3 is the second defense point to prevent an unauthorized cash returning command by confirming consistency between a cash returning command and the response message verification result. R4 is the third defense point to protect a cash returning command from replay attacks. R4 is used to confirm whether the cash returning command received by the cash handling module is significantly delayed or not. Table 6.3 shows the correspondence relation between the

Table 6.3 Correspondence between four conditions and prevented logical attacks

Conditions	Attack No.					
	A1	A2	B1	B2	C1	C2
R1	✓	✓	✓	✓	✓	✓
R2	✓	✓				
R3			✓	✓		
R4					✓	✓

conditions and prevented logical attacks. In Command Verification, the secure domains are created in peripheral devices to meet R1, which was proposed in section 4.2, we focus R2, R3, and R4 to prevent logical attacks specific to unauthorized deposit.

6.4 Implementation

6.4.1 Implementation Outline

The outline of the model 1 and 3 to verify the conditions R2, R3, and R4 is depicted in Figure 6.4. Model 3 can be applied to R2 because R2 is not a condition to verify a command transferring time. Secure elements as tamper-proof hardware are supposed to be installed in the proposed card reader and the proposed cash handling module. Figure 6.4 (a) shows the model 1 to verify a MAC generating a command for a transaction request message with R2 in the card reader using the cash amount from the cash handling module. The cash handling module, the card reader, the put-in cash, the MAC generating command, and the MAC key in the smart card correspond to the information acquiring device, the verified command executing device, the physical objects, the command, and the property in Figure 6.3 (a), respectively. If the command is verified, the verified command is forwarded to the smart card to generate a MAC for the message. Figure 6.4 (b) shows the model 3 to verify the command with R2 in the cash handling module using the cash amount stored in the cash handling module.

“Extract cash amount” and “Verify command with R2” in the cash handling module, the card reader, and the MAC key in the smart card corresponds to the verification information extracting module, the command verification module, the command executing device, and the protected property in Figure 6.3 (c), respectively. And then, the verified command is securely transferred to the card reader. It is noted that even if the command transferring is maliciously delayed, it is not an issue since it does not contribute to an unauthorized increase of an account balance. Some SIers may select the model 3 since they prefer to install the most important modules: the verification information extracting module and the command verification module, into the cash handling module inside the safe even if the modules are protected with a tamper-proof mechanism as described in Section 6.3.1.

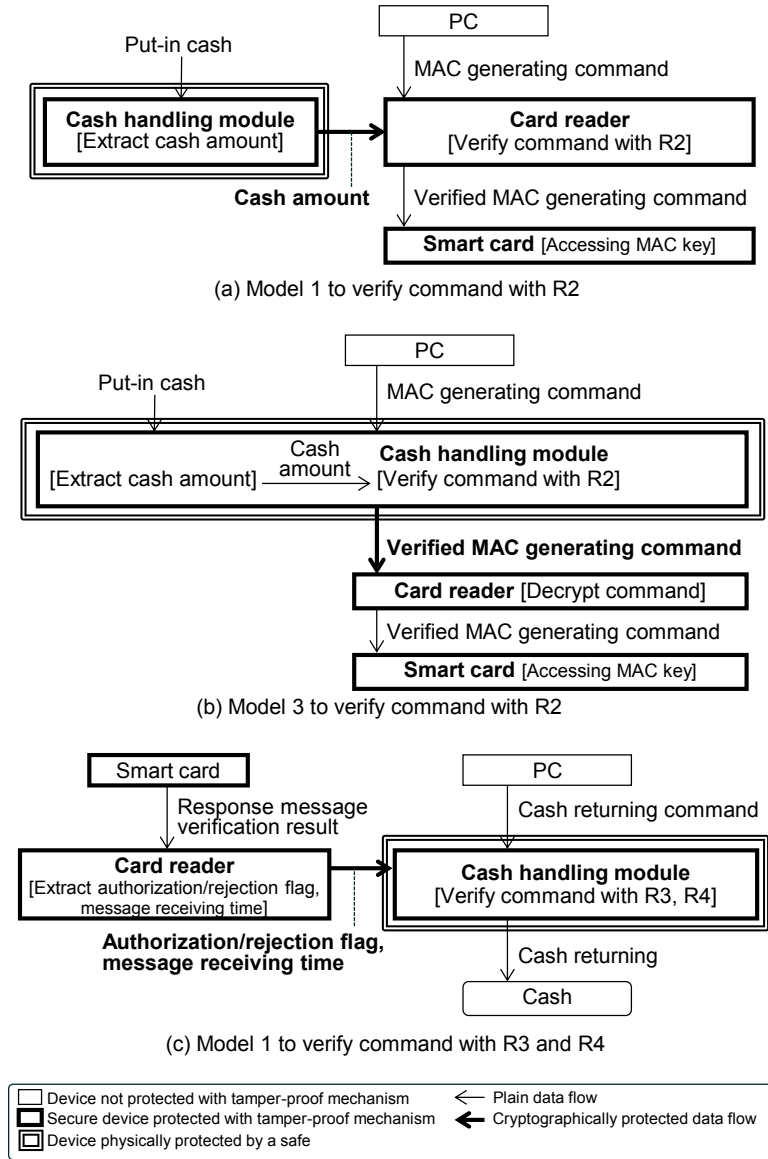


Figure 6.4 Implementation outline of Command Verification

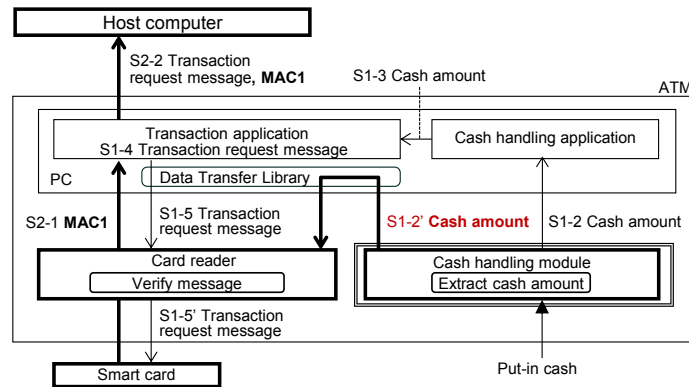
Figure 6.4 (c) shows the model 1 to verify a cash returning/storing command with R3 and R4 in the cash handling module. The card reader, the cash handling module, the response message verification result, the cash returning command, and the cash correspond to the information acquiring device, the verified command executing device, the physical objects, the command, and the property in Figure 6.3 (a), respectively. Since R4 is a condition for a command transferring time, the only model 1 can verify the command. The implementation uses a (transaction) authorization/rejection flag

and the received time of the response message verification result (hereinafter called “message receiving time”). These are securely transferred from the card reader to the cash handling module to verify the command. The response message verification result and the cash returning command correspond to the input data and command in Figure 6.3 (a), respectively. The card reader can receive the valid response message verification result since the card reader directly contacts the smart card. The card reader generates the authorization/rejection flag from the verification result.

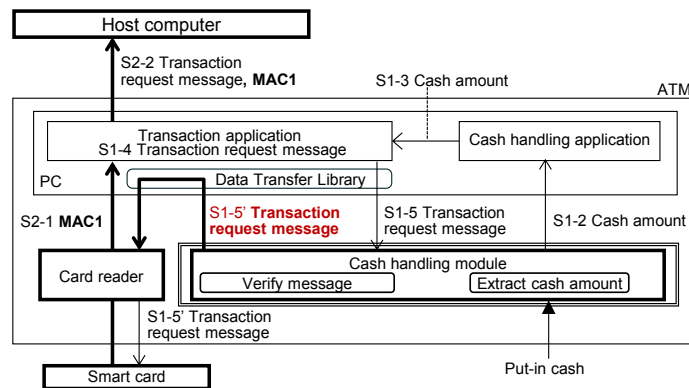
6.4.2 Detailed Data Flow of Implementation

The data flow of the implementation examples of Command Verification is described in this section. There is not any physical communication cable such as USB/RS-232c cable directly connecting between existing peripheral devices. A cryptographic communication between the peripheral devices is implemented by utilizing the existing USB/RS-232c cables between peripheral devices and the PC. “Data Transfer Library” is newly introduced in the PC to simply provide a communication path transferring encrypted data between the peripheral devices. Even if the Data Transfer Library is infected with malware, the integrity of encrypted data transferred in the Data Transfer Library is still assured. Data Transfer Library is supposed to be installed in a layer below the applications. Figure 6.5 illustrates the data flow of the implementation outlines shown in Figure 6.4. The cryptographic key management and a session creation for each cryptographic communication are supposed to conform to either the PCI requirements [14] [19] [30] or the EMV specifications [21] to meet confidentiality, integrity, and authenticity. A session of each cryptographic communication is supposed to be preliminarily created. The detailed process flows of each implementation of Figure 6.5 are described as follows. Only modified processes and the related processes are explained here.

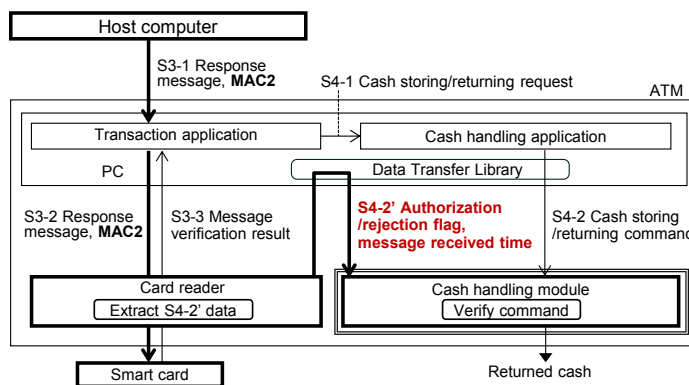
Figure 6.5 (a) shows the modified data flow corresponding to Figure 6.4 (a). The cash handling module returns an S1-2 cash amount to the cash handling application. And then the cash handling module stores the cash amount in it. The transaction application sends the S1-5 transaction request message to the card reader through the Data Transfer Library. Once the Data Transfer Library receives the message, the Data Transfer Library requests the cash handling module to send the S1-2 encrypted cash amount and forwards it to the card reader. The card reader decrypts the S1-2 encrypted cash amount and verifies the message with R2 using the cash amount. If the message validity is successfully verified, the card reader forwards the S1-5’ transaction



(a) Model 1 to verify request message with R2



(b) Model 3 to verify request message with R2



(c) Mode I1 to verify a command with R3 and R4

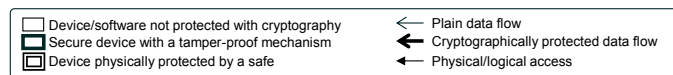


Figure 6.5 Implementation example of Command Verification

request message to the smart card.

Figure 6.5 (b) shows the modified data flow corresponding to Figure 6.4 (b). S1-2 is omitted since it is the same as in Figure 6.5 (a). The transaction application sends the S1-5 transaction request message to the cash handling module through the Data Transfer Library. The cash handling module verifies the S1-5 message with R2 using the cash amount stored in the cash handling module. The Data Transfer Library requests the cash handling module the S1-5 verified message in an encrypted form, and forwards it to the card reader. The card reader decrypts the encrypted message and forwards it to the smart card.

Figure 6.5 (c) shows the modified data flow corresponding to Figure 6.5 (c). The card reader receives the S3-2 response message and MAC2. The card reader receives an S3-3 response message verification result from the smart card, and stores the message receiving time in it. And then, the card reader generates an authorization/rejection flag from the result and stores the flag in it. The cash handling application sends either an S4-2 cash storing command or a cash returning command to the cash handling module through the Data Transfer Library following the S4-1 request from the transaction application. Once the Data Transfer Library receives either command, the Data Transfer Library requests the card reader to send the S4-2 authorization/rejection flag and the message receiving time in an encrypted form and then forwards them to the cash handling module. The cash handling module decrypts the encrypted data and verifies the cash storing/returning command with R3 and R4 using the decrypted data. If the command validity is successfully verified, the cash handling module executes the command.

6.4.3 Architecture of the Proposed Peripheral Devices

The architecture examples of the proposing peripheral devices are depicted in Figure 6.6. In general, an existing card reader is equipped with a slot to install a secure element for mutual authentication between a smart card and a terminal. A secure element to achieve Command Verification can be installed in the slot. The contact point interface to communicate with a smart card is also equipped with a secure element. Those two secure elements are cryptographically connected in order to protect contents transferred from a smart card to the PC and from the PC to the smart card even in the card reader. Additionally, the firmware in the controller is also supposed to be protected from unauthorized manipulation with digital signatures installed in the

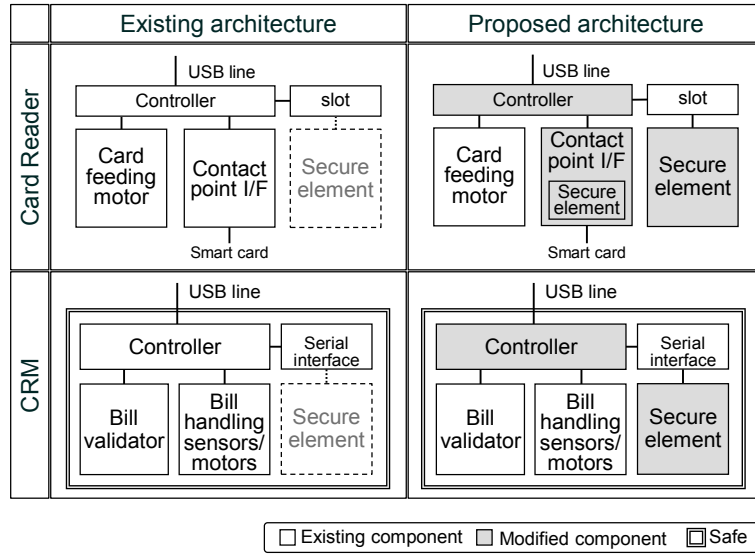


Figure 6.6 Comparison of existing devices and proposing devices

secure element. The firmware running on the RAM in the controller is supposed to be still secure by self-tests with the digital signatures. For example, the firmware hash is calculated once every day in the controller. The hash is transferred to the secure element and verified with the digital signatures. Such structure and processes are practical since the PCI requirements [14] [30] define similar requirements.

An existing cash handling module is equipped with a serial interface to expand the functions in many cases. A circuit board implementing a secure element can be installed on the serial interface. The firmware in the controller is also supposed to be protected from unauthorized manipulation even during running on the RAM with digital signatures installed in the secure element as well as the card reader. Furthermore, the whole cash handling module is protected from unauthorized physical access by a tightly controlled safe. Thus the firmware is logically and physically protected.

6.4.4 Evaluation of Command Verification through Implementation

In this section, we show from a qualitative and quantitative perspective that Command Verification can effectively prevent the logical attacks in actual ATM

operations with the implemented systems presented in Section 6.4.2. Regarding a qualitative perspective, effectively preventing the logical attacks requires meeting the conditions of (A), (B), and (C) as described in section 5.2.3. These conditions were derived from the issues of existing ATM systems and operations, and show that a measure should not overload ATM operations. A practical security effect of a measure without tamper-proof hardware highly depends on operational protection, namely management. And the security effect can be largely decreased if the measure requires heavy management workloads as explained in Section 5.2.2. Command Verification applied to cash withdrawal transactions with a smart card was shown to meet three conditions in section 5.3.3. Command Verification applied to deposit transactions with a smart card is also shown to meet the conditions in the following description.

(A) A security measure should not significantly impact management workloads of existing ATM operations.

Command Verification does not significantly impact the management workloads for the PCs containing many files to protect since the measure relies on the peripheral devices equipped with tamper-proof hardware. The number of firmware in the peripheral devices is typically one or two and much smaller than that of executable files in the PC. Such a small number of firmware can be protected with existing secure elements. In this way, tight protection of the PCs causing quite heavy management workloads is not a critical issue to prevent unauthorized deposit.

(B) A security measure should not significantly impact ATM system availability.

Command Verification does not rely on the PC but on the peripheral devices equipped with tamper-proof hardware. Thus frequent OS updating/hardening for a security patch, which would significantly impact ATM system availability, is not a necessary condition in Command Verification. Financial institutions can take enough time to comprehensively test many software components in the PC before releasing them to prevent occasional system troubles while mitigating zero-day attack risks of unauthorized deposit.

(C) The logical attacks cannot be successful even though the integrity of all software related to dispensing commands is compromised.

Command Verification can prevent the logical attacks for unauthorized deposit without relying on the integrity of all software of the PC. Even if the integrity of the Data Transfer Library is compromised in the PC, the logical attacks cannot

Table 6.4 Annual numbers of potential unauthorized access to the files to protect

Comparison items	Existing guidance	Command Verification
Protection measures	More than 30 requirements including whitelisting-based anti-malware, hard disk encryption, OS hardening	The implemented systems of Command Verification
Number of executable files to protect	20,000	2
Number of executable files to protect by management	20,000	0
Annual numbers of potential unauthorized access to the files	3,120 million (20,000*3000*52)	0 (0*3000*52)

still be successful because the Data Transfer Library just provides a communication pass to transfer encrypted data. Command Verification can work as a defense in depth in cases that the PC is compromised.

The requirements of the existing guidance [1] as representative of existing measures do not meet the three conditions as explained in section 5.3.3.

Concerning a quantitative perspective, let us estimate each measure's annual numbers of potential unauthorized access to files to protect in order to compare the practical effect of the existing guidance and the implemented systems of Command Verification. These numbers are correlating to practical management workloads to prevent unauthorized access to the files in ATM operations. A comparison of the initial costs is omitted here since the costs of the measures when a system is built according to the existing guidance are much more than those of Command Verification. Costly measures: encrypted communication between the PC and the cash handling module, whitelisting-based anti-malware, sandboxing, hard disk encryption, various ATM monitoring systems and so forth are required in the existing guidance. Regarding Command Verification, the proposed card readers and cash handling module based on existing devices and the secure elements can be developed and provided at a reasonable cost as described in section 5.3.2. Either Figure 6.5 (a) and (c), or Figure 6.5 (b) and (c) can be used for the estimation. The assumption is that the number of ATMs is three thousand and that cash replenishment/collection is conducted once a week, i.e. 52 times per year for each ATM. The number of the executable files to protect in the PC is twenty thousand for the existing guidance, while the number of the files to protect in the peripheral devices is two for Command Verification, namely, a file of firmware is implemented for the card reader and the cash handling module. The Data Transfer Library is not counted as described in the condition (C). The result is shown in Table 6.4. The number is 3,120 million for the existing guidance while zero for Command

Verification. “Number of executable files to protect” is 20,000 for the existing guidance, while 2 (one ten-thousandth) for Command Verification. “Number of executable files to protect by management” in Table 6.4 is zero for Command Verification since the firmware is protected by not management but tamper-proof hardware. In this way, our proposal is much better than the existing guidance.

6.5 Discussion

In this chapter, we explained an application of Command Verification to two transaction sub-processes in an ATM transaction, namely application to a transaction sub-process before/after communication between an ATM and the host computer in a deposit transaction with a smart card. There are multiple protected properties from multiple attack surfaces in the transaction sub-processes, and constraints to be satisfied which are coming from existing systems and operations. To cope with the issues, we proposed an implementation model analysis of Command Verification to achieve suitable implementation for each defense point in a deposit transaction. In detail, the features of each implementation model are compared and candidate models are selected to conform to existing systems and operations. And then, the suitable implementation models can be selected among the candidates to meet the requirements at each defense point. Two types of proper implementation were derived in the sub-process before communication between the smart card and the host computer, and one type of the proper implementation is derived in the sub-process after the communication. As the implemented systems of Command Verification protect property with the peripheral devices, it can work as a defense in depth when the PC is compromised.

We also showed that Command Verification prevents the logical attacks effectively in actual ATM operations from a qualitative and quantitative perspective with the implemented systems. Regarding a qualitative perspective, we showed that Command Verification can also meet the three conditions so as not to impose on financial institutions a heavy burden to tightly control the PCs. Since Command Verification protects property with the peripheral devices, there is no need to fully enforce the recommendations of the existing guidance, and the heavy burden is relieved. Regarding a quantitative perspective, we showed that the annual operational cost of Command Verification is reduced to less than ten-thousandth of the EUROPOL’s guidance. We expect that the primary concept of Command Verification can also be

applied to ATM transactions with magnetic stripe cards, contactless cards, smartphones, and QR codes. Securely protecting the new types of transactions are remaining issues to be addressed in future works.

Chapter 7 Application of Command Verification to All Transaction Sub-processes

7.1 Introduction

An application of Command Verification to all transaction sub-processes, issues of the application, and a solution to the issues are described using a cash withdrawal transaction with a magnetic stripe card in this chapter. Recently, criminals frequently carry out logical attacks on ATMs and financial institutions' networks to steal cash in more than 30 countries, and these attacks resulted in serious social issues. Existing guidance trying to protect the PC in an ATM could be bypassed or disabled by criminals since frequent physical/logical access inside ATMs are required in existing ATM operations. ATM management costs could increase if the integrity of executable files in the PCs is assured by tight ATM operational management to cope with that issue.

To solve the issue, Command Verification is proposed in section 4.1, in which controlled peripheral devices themselves verify commands sent from the PC before executing the commands to access the property, and the primary model is depicted in Figure 4.4. Although Command Verification is applied to smart card transactions in chapter 5 and 6, it should be also applied to widespread cash withdrawal transactions with a magnetic stripe card, since there are many logical attacks targeting those transactions. When Command Verification is applied to the magnetic stripe card transactions, there are a variety of implementable systems because all transaction sub-processes in a cash withdrawal transaction must be protected due to the poor existing security mechanisms. In smart card transactions, Command Verification is applied to prevent only unauthorized cash dispensing commands sent from the PC, since the Primary Account Number and transaction messages transferred between an ATM and the host computer, are protected in accordance with EMV specifications [21].

In the magnetic stripe card transactions, properly implementable systems of Command Verification should be selected from the variety of the implementable systems in three viewpoints: preventing a wide range of logical attacks in a transaction,

harmonizing with existing ATM operations, and minimizing the number of peripheral devices to be modified. In general, ATMs are composed of a set of peripheral devices, which are supplied as one of the multiple models by multiple device vendors, in conformity to the required specifications of the financial institution and the country regulations. Thus, many devices equipped with greatly modified functions for Command Verification result in increased costs and delayed delivery times. Thus, the number of peripheral devices to be modified should be minimized, and device/system design to implement Command Verification should be standardized to meet a lot of financial institutions' requirements. This chapter proposes a systematic implementation design method of Command Verification [44] [45] to satisfy the three viewpoints described above. By applying the design method to magnetic stripe card transactions, three proper systems out of the 135 implementable systems can be selected.

7.2 Issues of Command Verification

7.2.1 An ATM System and Magnetic Stripe Card Transaction

Figure 7.1 outlines an example of an ATM system and data flow of an existing cash withdrawal transaction with a magnetic stripe card, which is the same as Figure 4.10. Refer to section 4.2 (c) for the detailed processing flow of the example. An ATM consists of a PC and peripheral devices. The PC logically consists of three layers: multi-vendor application, a standardized ATM platform [20] to control the peripheral devices, and an Operating System (OS). It is noted that the ATM platform and the OS are not shown in the figure. The ATM platform is vulnerable to unauthorized APIs access due to unencrypted APIs and its openness to the public. Encrypting PIN pad is a peripheral device used by an ATM user to enter Personal Identification Number (PIN). The encrypting PIN pad outputs an encrypted PIN [14] [15] [19] and the PIN is transferred to a Hardware Security Module (HSM) connected with the host computer. Then the hardware security module extracts a PIN from the encrypted PIN to verify the PIN for the user's authenticity. The hardware security module and encrypting PIN pads must be a tamper-proof secure cryptographic device meeting the PCI PIN requirements [14] [19]. It is supposed that the multi-vendor application includes "transaction application"

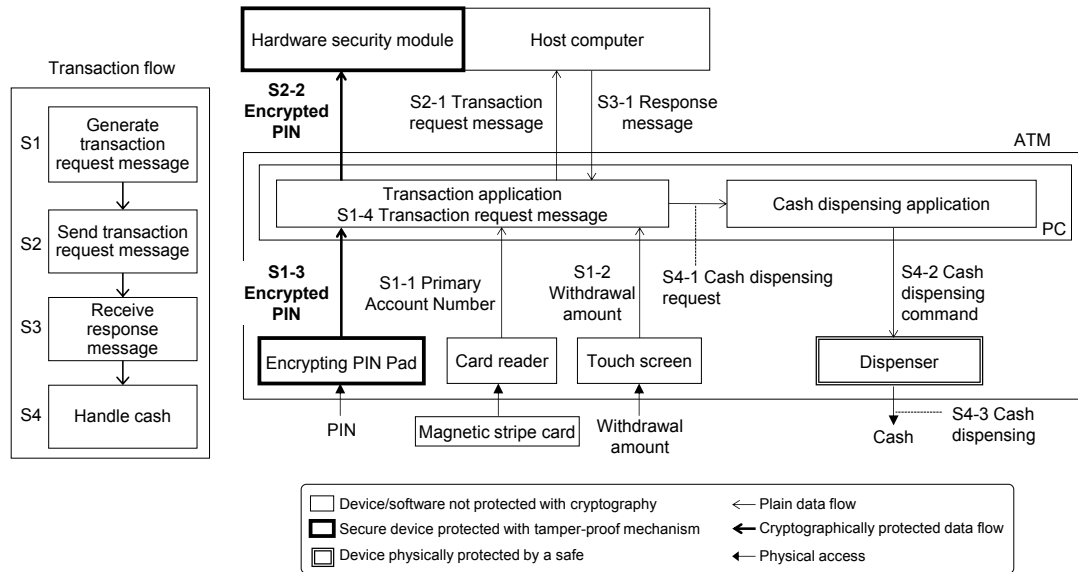


Figure 7.1 Data flow example of existing magnetic stripe card transaction.

processing transaction messages and “cash dispensing application” controlling the dispenser. A transaction consists of four sub-processes: S1 to S4.

7.2.2 Issues of Existing Security Measures

The standard security measures in Figure 7.1 are described below. The encrypting PIN pad and the hardware security module are protected with a tamper-proof mechanism and a PIN is protected cryptographically. The dispenser is supposed to be secure against unauthorized physical manipulation because it is physically protected with a safe. Except for those devices and the data flow, the PC, the peripheral devices, the USB/RS-232C cables in an ATM, and the WAN between ATMs and the host computer could be targets of the logical attacks to steal cash from ATMs in each sub-process of a transaction, which are described in Table 1. It is noted that logical attacks on peripheral devices are omitted since the protection of peripheral devices is included in Command Verification.

The existing guidance [1] [3] [10] [11] [12] [13] try to protect executable files in the PC against A2, C1, and D1. Furthermore, those measures try to cryptographically protect the WAN from B1, and the USB/RS-232C cables from A1 and C2. However, cryptographic communication does not work to prevent D2 since it is a kind of a replay attack that a command is temporarily held by a malicious device to make false trouble

Table 7.1 Logical attacks to steal cash from ATMs

No.	Sub-process	Attack objective	Attack method	Targeted property	Outline of logical attack
A1	S1 Generating transaction request message	Manipulation of transaction request message for fraudulent withdrawal	Malicious device	S1-1 PAN, S1-3 withdrawal amount	- A malicious device on a USB/RS-232C cable manipulates a PAN for a reverse brute force attack to fraudulently withdraw cash from other user's accounts. - A malicious device manipulates a withdrawal amount for cash robbery from a confused ATM user.
A2			Malware	S1-4 Transaction request message in Transaction application	- Malware manipulates a PAN for a reverse brute force attack to fraudulently withdraw cash from other user's accounts. - Malware manipulates a withdrawal amount for cash robbery from a confused ATM user.
B1	S2 Send transaction request message, S3 receive response message	Unauthorized cash withdrawal	Man-in-the-Middle	S2-1 Transaction request message, S3-1 reply message	- Same as A2 - Fake host responses are generated to withdraw money without debiting the fraudster's accounts.
C1	S4 Handle cash	Unauthorized cash withdrawal	Malware	S4-1 cash dispensing request, S4-2 cash dispensing command in PC	Malware forces the ATM to cash-out.
C2			Malicious device	S4-2 cash dispensing command on USB/RS-232C	An external computer connected to the dispenser forces it to cash-out.
D1		Making a false trouble for fraudulent cash dispensing	Malware	Transferring time of S4-2 in PC	Either cash dispensing request or cash dispensing command is temporarily held by malware to make a false trouble, and then is sent again by operating malware to steal cash after a user leaves the ATM.
D2			Malicious device	Transferring time of S4-2 on USB/RS-232C	A cash dispensing command is temporarily held by a malicious device on the USB/RS-232C cable to make a false trouble, and then sent to the dispenser again by operating the malicious device to steal cash after a user leaves the ATM.

in order for the ATM user to leave the ATM for stealing cash. A malicious person steals cash dispensed from the ATM with operating the malicious device. It is noted that cryptographic protection of the communication also depends on the PC's security because the cryptographic keys are stored in the PC. There are issues of increasing management costs if the integrity of executable files is assured by tight ATM operational management as explained in Section 7.1.

7.2.3 Conditions to Implement Command Verification

When Command Verification is applied to magnetic stripe card transactions, there are a variety of implementable systems explained in Section 7.1. Properly implementable systems should be selected among the variety of the systems from three viewpoints.

(1) Preventing a wide range of logical attacks in a transaction

Various logical attacks targeting property in each transaction sub-process shown in Table 1 should seamlessly be prevented in a whole transaction.

(2) Harmonizing with existing ATM operations

Implemented systems should be harmonized with existing ATM operations to minimize an impact on the operations. In particular, the cryptographic key setting implementation for cryptographic communication in an ATM system should be minimized because such key settings could be an attack target [50] and tightly controlled key settings are required. Such key settings should be minimized from the viewpoint of work efficiency since maintenance staffs may exchange a troubled part in an ATM with a service part for trouble shooting, which requires cryptographic key settings in some cases.

(3) Minimizing the number of peripheral devices to be modified

The number of peripheral devices that need to be modified to implement functions of application and functions of other peripheral devices should be minimized so that Command Verification can easily be applied to various systems. For example, if the input data in Figure 4.4 is a transaction request/reply message, the information acquiring device must parse the messages and it is an application function. As explained above, many peripheral devices having functions of applications and functions of other peripheral devices could result in complicated device modification and could affect costs and delivery times. Thus, the number of peripheral devices to be modified should be minimized.

7.3 Design Method to Implement Command Verification

7.3.1 Implementation Models

To design properly implementable systems of Command Verification, the features of implementation models derived from the primary model shown in Figure 4.4 should be clarified. The implementation models are depicted in Figure 7.2. Each device consists of tamper-proof hardware and an existing control mechanism including firmware. The authenticity of the firmware is supposed to be assured by digital signatures stored in the tamper-proof hardware although the signatures are not shown in the figure.

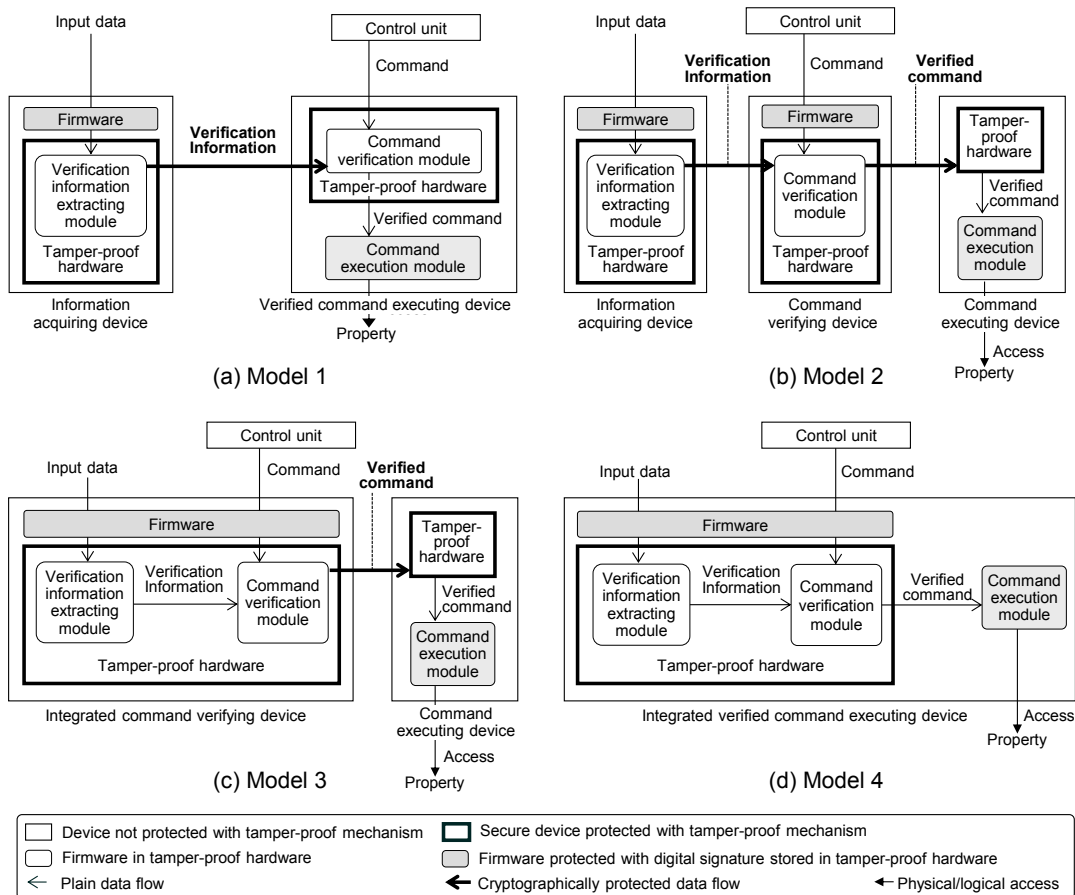


Figure 7.2 Implementation models of Command Verification

Table 7.2 Comparison of implementation models for magnetic stripe card transactions

No.	Features	Model 1	Model 2	Model 3	Model 4
1	Cryptographic communication between peripheral devices	One	Two	One	Zero
2	Validity of command from the viewpoint of command transfer time	Verifiable	Not verifiable	Not verifiable	Verifiable
3	Validity of command except the viewpoint of command transfer time	Verifiable	Verifiable	Verifiable	Verifiable
4	Peripheral device modification to support many vendors' peripheral devices	Good	Poor	Poor	Good

Cryptographic functions implemented in the devices are also not shown in the figure. “Verification information extracting module” receives input data through the firmware in the device. It is supposed that data flowed in the devices are protected with the firmware or a physical measure such as a safe.

Figure 7.2 (a) shows model 1 that each device corresponds to the device of the primary model. Figure 7.2 (b) shows model 2 that the verified command executing device in Figure 7.2 (a) is split into two devices: a command verifying device and a command executing device. Figure 7.2 (c) shows model 3 that the information acquiring device and the command verifying device in Figure 7.2 (b) are integrated into one device, namely an integrated command verifying device. Figure 7.2 (d) depicts model 4 that all devices in Figure 7.2 (b) are integrated into one device.

Table 7.2 summarizes the features of each implementation model in Figure 7.2 for magnetic stripe card transactions. In conclusion, model 1 and model 4 are recommended since they have preferable features. The preferable point of each feature in Table 7.2 is explained as follows. In terms of the No.1 feature, a smaller number is better from the viewpoint of minimizing cryptographic communication as explained in the condition (2) of section 7.2.3. Accordingly, model 2 is not preferable. Although model 4 is the most preferable, the model can be adopted only when a command can be verified with input data of one device. Thus model 1 and model 3 should be also acceptable. In terms of the No.2 feature, “verifiable” is preferable. Model 2 and model 3 are not verifiable since it is difficult to verify command transfer time to detect a command being temporarily held. The command executing device is not equipped with a function to verify the transfer time of the received command in these models. In terms of No.3 feature, “verifiable” is preferable, and all the models can verify a command except a viewpoint of command transfer time. In terms of the No.4 feature, “good” is preferable. Model 2 and model 3 are also not preferable because the command

verification module and the command execution module are not in one device. The command verification module in the (integrated) command verifying device must support the command specifications of the command executing device in order to parse the command for verifying it. In this way, model 2 and model 3 do not meet the condition (3) of section 7.2.3.

7.3.2 Outline of Implementation Design Method

An implementation design method is introduced to systematically design properly implementable systems of the primary model to meet the three conditions described in section 7.2.3. The method consists of three implementation steps and guidance.

Step 1: Enumerate property and logical attacks targeting the property in all transaction sub-processes.

Guidance 1. Each logical attack targets the property of a different transaction sub-process in a whole transaction to steal cash. To prevent such logical attacks, ensuring consistency in each transaction sub-process is required throughout a whole transaction. Since the targeted property is different for each transaction sub-process to ensure the consistency, each property and logical attacks targeting the property should be listed up for all transaction sub-processes.

Step 2: Identify information to verify a command accessing the property, identify the source of the information, and decide a device to securely acquire the information.

Guidance 2. Information to verify a command accessing the property should be acquired in a secure form and in a device as close as possible to the information source in order to ensure the validity of the information.

Step 3: Decide devices to verify a command accessing the property and devices to execute the verified command in light of recommended implementation models of the Control Command Verification.

Guidance 3-1. Select proper devices to verify a command to prevent target logical attacks. Data and parameters included in a command are also targets of validity verification. The proper device should be selected carefully if the validity is verified from the viewpoint of command transfer time since the only two implementation models can verify the validity.

Guidance 3-2. Select implementation models to harmonize with existing system operations. One of the points of harmonization is to minimize cryptographic communications in an implemented system so as to mitigate tight and complicated cryptographic key settings in system operations.

Guidance 3-3. Minimize the number of peripheral devices with functions of applications and functions of other devices so that many vendors can be easier to supply peripheral devices.

Guidance 3-4. Select proper devices that can seamlessly verify a command accessing a property in each transaction sub-process. ‘Seamlessly’ means that a device verifying a command in a transaction sub-process becomes a device providing information to verify a command in the following transaction sub-process. As a result, those selected devices provide a chain of consistency among transaction sub-processes to protect property in a whole transaction process.

7.4 Implementation

7.4.1 Implementation for Magnetic Stripe Card Transaction

In this section, a design process is explained to implement the Control Command Verification to a magnetic stripe card transaction in accordance with the implementation design method.

Step 1: Enumerate property and logical attacks targeting the property in all transaction sub-processes.

The property and logical attacks targeting the property are listed in Table 3 for each transaction sub-process. Protecting a PAN is required for a magnetic stripe card transaction although is not required for a smart card transaction. An altered PAN in a request message can be detected in a smart card transaction according to the EMV specifications [21].

Table 7.3 Targeted property and logical attacks

No	Sub-process	Logical attack	Targeted property
S1	Generating transaction request message	A1 Malicious device	S1-1 PAN, S1-2 withdrawal amount, S1-5 transaction request message
		A2 Malware	S1-4 Transaction request message
S2	Send transaction request message	B1 Man-in-the-Middle	S2-1 transaction request message
S3	Receive response message	B1 Man-in-the-Middle	S3-1 Reply message
S4	Handle cash	C1 Malware	S4-1 Cash dispensing request, S4-2 cash dispensing command in PC
		C2 Malicious device	S4-2 Cash dispensing command on USB/RS-232C
		D1 Malware	Transferring time of S4-1 cash dispensing request, Transferring time of S4-2 cash dispensing command in PC
		D2 Malicious device	Transferring time of S4-2 cash dispensing command on USB/RS-232C

Step 2: Identify information to verify a command accessing the property, identify the source of the information, and decide a device to securely acquire the information.

Information to verify a command and information acquiring devices are summarized in Table 7.4. Since ATMs work in accordance with inputs from peripheral devices and communication with the host computer, the information to verify a command should be acquired in the peripheral devices and a counterpart device of the host computer communication. A withdrawal amount should be input not in the touch screen but in the encrypting PIN pad supporting cryptographic functions according to the guidance 2. A verified request message and a verified reply message need to be acquired in a secure device to make a certain link between cash dispensing and debiting the user's account. However, there are no existing devices of an ATM to securely communicate with the host computer in magnetic stripe card transactions. Therefore, either the card reader, the encrypting PIN pad, or the dispenser should be selected to implement the functions securely acquiring the messages in order to be consistent with step 3. To prevent unauthorized cash withdrawal with a cash dispensing command, an authorized withdrawal amount is required, which is derived from the withdrawal amount in the request message and the host authorization flag in the reply message. The authorized withdrawal amount is compared with the dispensing amount in the command. To prevent a replay attack to a cash dispensing command, a reference time to measure

Table 7.4 Information to verify command and information acquiring device

No	Sub-process	Targeted property	Verification Information	Information acquiring device
S1	Generating transaction request message	S1-1 PAN, S1-2 withdrawal amount S1-5 transaction request message	S1-1 PAN, S1-2 withdrawal amount	Card reader, encrypting PIN pad
		S1-4 Transaction request message	S1-1 PAN, S1-2 withdrawal amount	Card reader, encrypting PIN pad
S2	Send transaction request message	S2-1 transaction request message	MAC1 for S2-1	Either card reader, encrypting PIN pad, or dispenser
S3	Receive response message	S3-1 reply message	MAC2 for S3-1	Host computer
S4	Executing cash dispensing	S4-1 Cash dispensing request, S4-2 Cash dispensing command in PC	Authorized amount (withdrawal amount in S2-1, host authorization flag in S3-1)	Either card reader, encrypting PIN pad, or dispenser
		S4-2 Cash dispensing command in PC	Reference time (RESMSG receiving time)	Either card reader, encrypting PIN pad, or dispenser
		Transferring time of S4-2 in PC	Reply message receiving time (reference time)	Either card reader, encrypting PIN pad, or dispenser
		Transferring time of S4-2 on USB/ RS-232C	Same as above	Same as above

command transferring time is required to detect whether the command is temporarily held or not. The reference time should be the time when either the card reader, the encrypting PIN pad, or the dispenser receives the reply message.

Step 3: Decide devices to verify a command accessing the property and devices to execute the verified command in light of recommended implementation models of the Control Command Verification.

Model 1 and model 4 should be selected as preferable models referring to Table 7.2 in accordance with the guidance 3-1, 3-2 and 3-3. Since a peripheral device communicating with the host computer must parse a request/reply message to verify them, which is an application function, only one device should have such functions to conform to the guidance 3-3. When the card reader is selected, the whole transaction process is depicted in Figure 7.3. The following functions are implemented in each sub-process pursuant to the guidance 3-4.

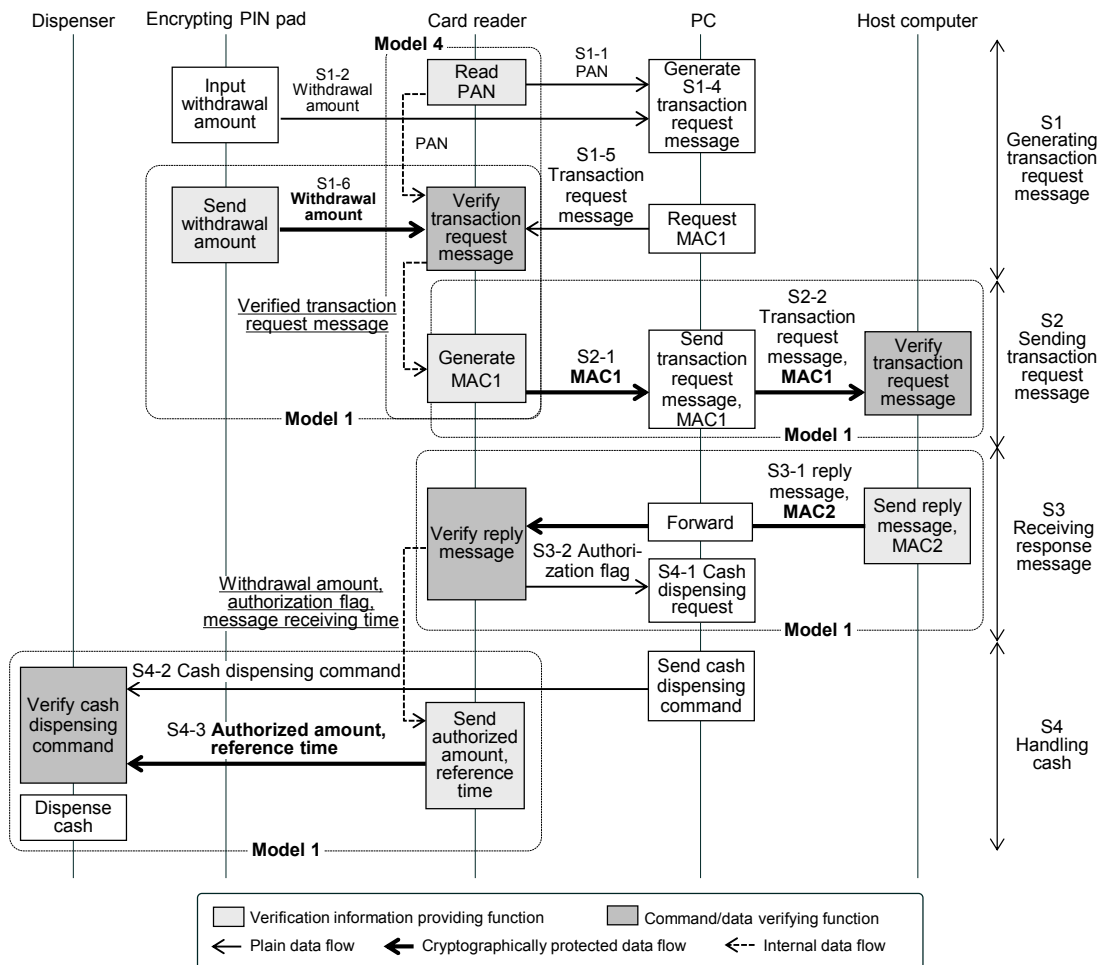


Figure 7.3 Data flow ensuring consistency among transaction sub-processes

(S1) Sub-process generating a transaction request message

The card reader verifies the transaction request message with a PAN internally transferred in the card reader and a withdrawal amount securely transferred from the encrypting PIN pad (Figure 7.4). It is a combination of model 1 and model 4 (Figure 7.3).

(S2) Sub-process sending the transaction request message

The card reader generates a MAC (hereinafter called “MAC1”) for the verified request message through the underlined verified request message so that the host computer can seamlessly verify the request message according to the guidance 3-4. The host computer verifies the request message with the MAC1, which is categorized to model 1.

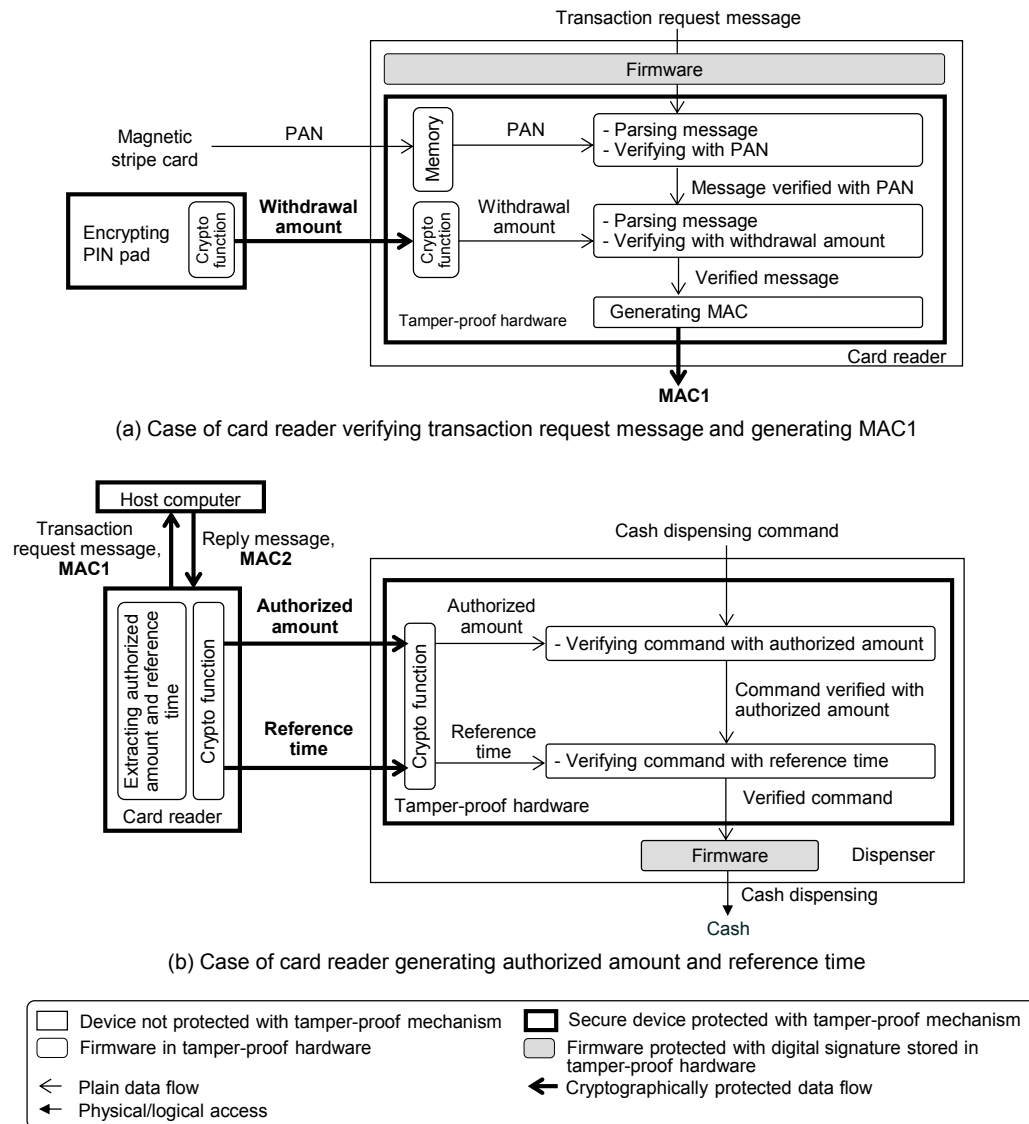


Figure 7.4 Implementation example of card reader communicating with the host computer

(S3) Sub-process receiving a response message

The card reader verifies the reply message with a MAC (hereinafter called “MAC2”) for the reply message received from the host computer, which is also categorized to model 1.

(S4) Sub-process handling cash

The card reader generates an authorized withdrawal amount from the underlined

Table 7.5 Summary of applied implementation models

(S1) Sub-process generating a transaction request message

Property		S1-1 PAN, S1-2 withdrawal amount, Request message in transaction AP	
Logical Attack		A1 Malicious device, A2 Malware	
Verification information		PAN	Withdrawal amount
Information acquiring device		Card reader	Encrypting PIN pad
Verifying device	Card reader	Model 4	Model 1
	Encrypting PIN pad	Model 1	Model 4
	Dispenser	Model 1	Model 1

(S2) Sub-process sending the transaction request message

Property		S2-2 Request message		
Logical Attack		B1 Man-in-the-Middle		
Verification information		MAC1		
Information acquiring device		Card reader	Encrypting PIN pad	Dispenser
Verifying device	Host computer	Model 1	Model 1	Model 1

(S3) Sub-process receiving a response message

Property		S2-3 Reply message		
Logical Attack		B1 Man-in-the-Middle		
Verification information		MAC2		
Information acquiring device		Host computer		
Verifying device	Card reader	Model 1		
	Encrypting PIN pad	Model 1		
	Dispenser	Model 1		

(S4) Sub-process handling cash

Property		S3-1 Cash dispensing request, S3-2 Cash dispensing command		
Logical Attack		C1 Malware, C2 Malicious device		
Verification information		Authorized amount		
Information acquiring device		Card reader	Encrypting PIN pad	Dispenser
Verifying device	Dispenser	Model 1	Model 1	Model 1
Property		Transferring time of S3-2 cash dispensing command		
Logical Attack		D1 Malware, D2 Malicious device		
Verification information		Reference time		
Information acquiring device		Card reader	Encrypting PIN pad	Dispenser
Verifying device	Dispenser	Model 1	Model 1	Model 4

withdrawal amount in the request message and the underlined host authorization flag in the reply message so that the dispenser can seamlessly verify the cash dispensing command sent from the PC with the authorized withdrawal amount. The card reader also generates a reference time from the underlined message receiving time so that the

dispenser can seamlessly verify the command transfer time. These two kinds of verification with the authorized withdrawal amount and the reference time are categorized to model 1.

The applied implementation models for each transaction sub-process are summarized in Table 7.5. Each gray level in Table 7.5 shows applied implementation models when either peripheral device is selected as the counterpart of the host computer communication. There are three proper systems according to the number of devices selected as the counterpart.

7.4.2 Detailed Data Flow of the Proper Systems

The data flow of the implementation examples is shown in Figure 7.5 for each peripheral device selected as the counterpart of the host computer communication. There is no physical communication cable between existing peripheral devices. Encrypted communication between the peripheral devices is implemented by utilizing existing USB/RS-232C cables between peripheral devices and the PC. “Data Transfer Library” (hereinafter called “DTL”) is newly introduced in the PC to simply provide a communication path between the peripheral devices to transfer encrypted data. DTL is supposed to be installed in a layer below the standardized APIs. Figure 7.5 (a) illustrates the data flow of the implementation example that the card reader is the counterpart device communicating with the host computer. The system related to a PIN is omitted in the figure. A programmable tamper-proof secure element providing the cryptographic functions is installed in the proposed card reader, the proposed encrypting PIN pad and the proposed dispenser while a hardware security module is implemented in the proposed host computer. The cryptographic key management and a session creation for each encrypted communication are supposed to conform to either the PCI requirements [14] [15] [19] [30] or the EMV specifications [21] to meet confidentiality, integrity, and authenticity. A session of each encrypted communication is supposed to be preliminarily created. The detailed process flows of Figure 7.5 (a) are described as follows.

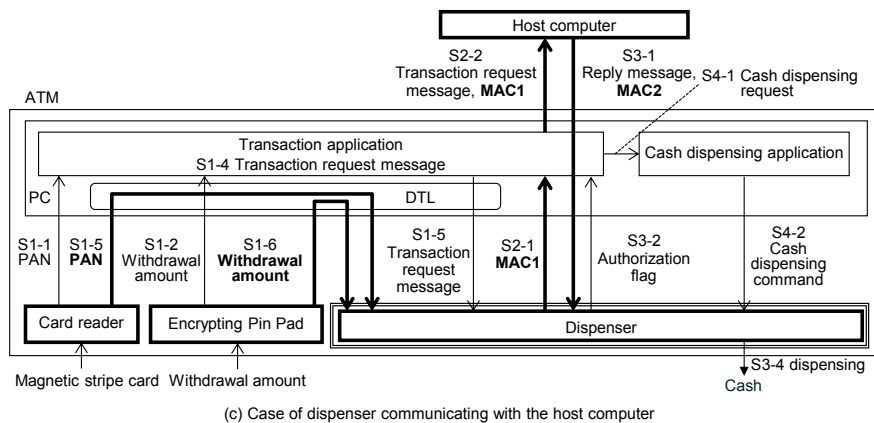
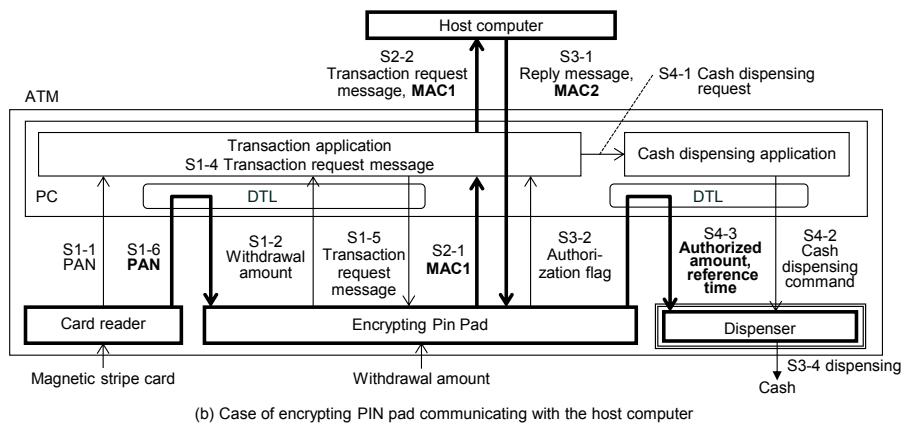
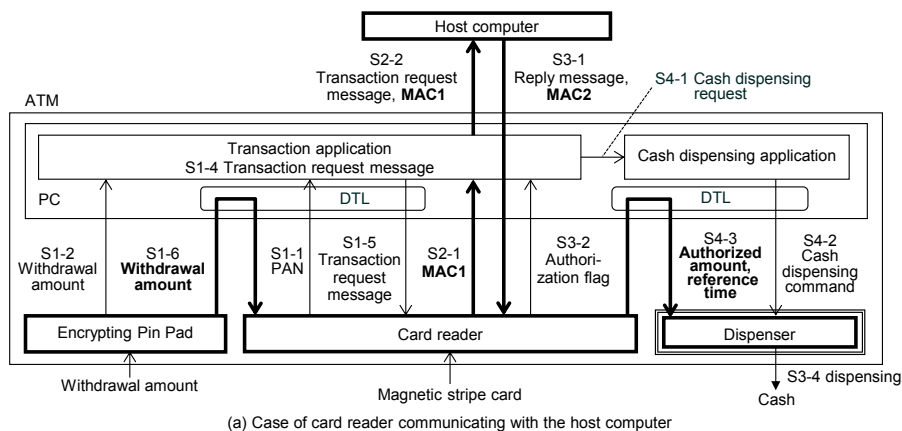


Figure 7.5 Implementation examples of Command Verification

(S1) Generating a transaction request message

The card reader sends an S1-1 PAN to the transaction application, and stores it in the secure element. The encrypting PIN pad sends an S1-2 withdrawal amount to the transaction application and stores the amount in it. The transaction application creates an S1-4 transaction request message and sends the S1-5 message to the card reader through DTL so as to make it generate a MAC1 for the message. When the DTL receives the message, the DTL requests the encrypting PIN pad to send the S1-6 withdrawal amount in an encrypted form and forwards it to the card reader. The card reader verifies the received message with the PAN stored in the secure element and the S1-6 withdrawal amount. The card reader also stores the withdrawal amount in the secure element.

(S2) Send a transaction request message

The card reader generates an S2-1 MAC1 for the verified message and sends it to the transaction application. The transaction application sends the S2-1 transaction request message and the MAC1 to the host computer, and then the host computer verifies the message with the MAC1.

(S3) Receive a response message

The host computer generates an S3-1 reply message including a host authorization flag and a MAC2 and sends them back to card reader through the transaction application. When the card reader receives them, it stores the message receiving time as the reference time. The card reader verifies the message with the MAC2 and returns the S3-2 authorization flag to the transaction application. The card reader also generates an authorized withdrawal amount with the flag and the withdrawal amount stored in the secure element.

(S4) Handle cash

The transaction application provides the cash dispensing application with an S4-1 cash dispensing request, and the cash dispensing application sends an S4-2 cash dispensing command to the dispenser through the DTL. The DTL requests the card reader to send the S4-3 authorized withdrawal amount and the reference time in an encrypted form and then forwards them to the dispenser. The dispenser receives the command and the S4-3 data and calculates the command transfer time with the reference time. And then the dispenser verifies the command with the authorized withdrawal amount to confirm whether the dispensing amount in the command is identical to the authorized withdrawal amount. The dispenser

also verifies the command transfer time to confirm whether the transfer time exceeds a predetermined threshold. If they are successfully verified, the dispenser dispenses cash.

Figure 7.5 (b) shows the data flow of the implementation example that the encrypting PIN pad is the counterpart device communicating with the host computer. The functions of the encrypting PIN pad and the card reader are inversely positioned in Figure 7.5 (a) and (b). Figure 7.5 (c) depicts the implementation example that the dispenser is the counterpart device communicating with the host computer. The detailed data flow of those examples is omitted. Deciding the most recommended implementation in Figure 7.5 depends on the development costs and harmonization with the detailed specifications of the existing system and the operations. However, it is out of scope in this discussion.

7.4.3 Evaluation of the Design Method

The number of all implementable systems of the Control Command Verification is estimated to evaluate the effect of the design method. The model 2 of Figure 7.2 (b) is utilized to estimate that number since the model consists of the three elementary devices. There are two steps to estimate the number. The first step is to estimate the number of peripheral device combinations in each transaction sub-process with three devices: the card reader, the encrypting PIN pad, and the dispenser. In the sub-process generating a transaction request message, the information acquiring devices are the card reader outputting a PAN and the encrypting PIN pad outputting a withdrawal amount. Since those devices are fixed, there is one device combination. On the other hand, the command verifying devices can be selected from the three devices. A verifying device for PAN and a verifying device for a withdrawal amount can be independently selected from the three devices. Since the transaction request message is sequentially verified by a verifying device for a PAN and by a verifying device for a withdrawal amount, there are 9 ($= 3 \times 3$) verifying device combinations. Order of the verifying devices can be transposed except that the both verifying devices are identical. Thus, there are additional 6 ($= 3 \times 3 - 3$) combinations and total of 15 combinations. The command executing device, namely, a device generating MAC1 for the transaction request message can be selected independently among the three devices. Therefore, the number of total device combinations is 45 ($= 1 \times 15 \times 3$).

In the sub-process communicating with the host computer, a device

communicating with the host computer should coincide with the device generating MACs since a cryptographic session for MACs must be established between the device and the host computer, and there is only one device combination. In the sub-process executing cash dispensing, the information acquiring device should also coincide with the communicating device. The command verifying device can be selected from the three devices. The command executing device must be the dispenser. In this way, the number of the device combinations is 3 ($= 1 \times 3 \times 1$). The second step is to multiply the estimated numbers of the peripheral device combinations in each sub-process. That is 135 ($= 45 \times 1 \times 3$). By designing the systems pursuant to the proposed design method, three proper systems out of the 135 implementable systems can be selected as described in section 7.4.2.

7.4.4 Architecture of the Proposing Peripheral Devices

The architecture examples of the proposed peripheral devices are depicted in Figure 7.6. In general, an existing card reader is equipped with a slot to install a secure element for mutual authentication between a smart card and a terminal. The secure element can be installed to the slot. The magnetic head to read the PAN on a magnetic stripe card is equipped with another secure element. That secure element is cryptographically connected with the secure element installed on the slot in order to protect PAN from unauthorized access inside the card reader. Such a structure is practical since the PCI requirements [14] [30] define similar requirements for card readers in point-of-sale terminals. Additionally, the firmware in the controller is also supposed to be protected from unauthorized manipulation with digital signatures installed in the secure element. The firmware running on the RAM in the controller is supposed to be still secure by self-tests with the digital signatures in conformity to the PCI requirements.

Concerning the dispenser, an existing dispenser is equipped with a serial interface to expand the functions in many cases. A circuit board implementing a secure element can be installed on the serial interface. The firmware in the controller is also supposed to be protected from unauthorized manipulation even during running on the RAM with digital signatures installed in the secure element as well as the card reader. Furthermore, the whole dispenser is protected from unauthorized physical access by a tightly controlled safe. Thus the firmware is logically and physically protected. As for

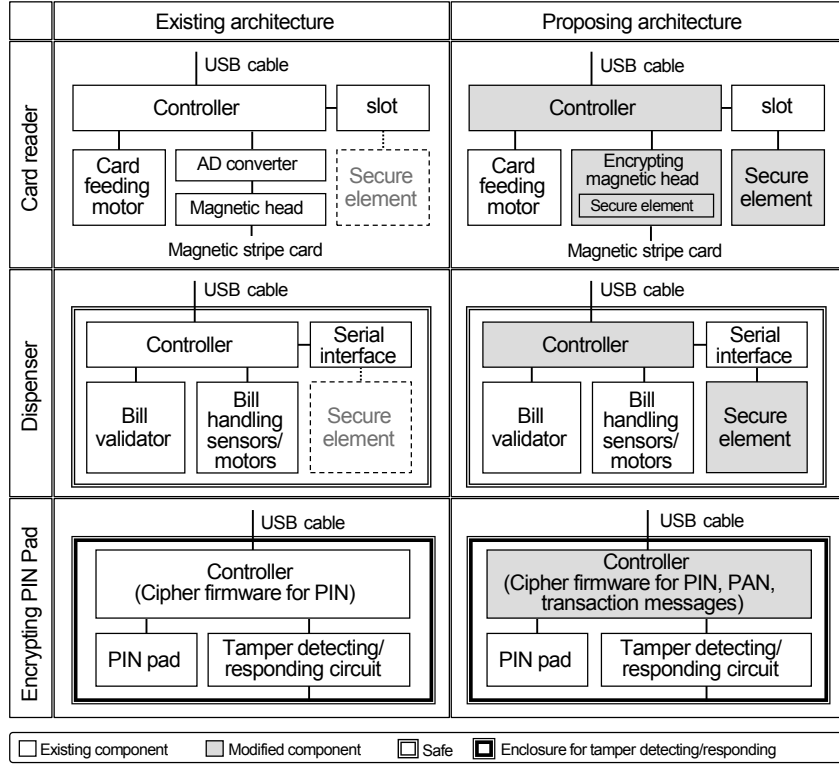


Figure 7.6 Comparison of existing devices and proposing devices

the encrypting PIN pad, an existing encrypting PIN pad is protected with an enclosure for tamper detecting/responding in conform to the PCI requirements [14]. While existing cryptographic functions are implemented in the firmware conforming to the PCI requirements in many cases, additional cryptographic functions can also be implemented in the firmware.

7.5 Discussion

In this chapter, we proposed an implementation design method of Command Verification, which is a verification method of control commands by controlled devices themselves. When Command Verification is applied to magnetic stripe card transactions, there are a variety of implementable systems because Command Verification must protect all transaction sub-processes in a cash withdrawal transaction due to poor existing security mechanisms. Proper systems can be selected with the proposed design method from the variety of the systems from three

viewpoints: preventing a wide range of logical attacks in a transaction, harmonizing with existing ATM operations, and minimizing the number of peripheral devices to be modified. The proposed design method to select proper systems consists of three design steps. Step 1 is to enumerate logical attacks and targeted property in all transaction sub-processes. Step 2 is to decide proper devices providing information to verify a command accessing the property. Step 3 is to decide proper devices verifying the command with the provided information so that consistency in each transaction sub-process is ensured with recommended implementation models throughout a whole transaction process. By applying the implementation design method to magnetic stripe card transactions, three proper systems out of the 135 implementable systems were selected. That is, the number of candidate systems to be examined in detail was reduced to one forty-fifth. We expect that the implementation design method can also be applied not only to ATM deposit and remittance with a magnetic stripe card, but also such devices operating with payment transactions as ticketing machines and vending machines. They are going to be proposed as future works.

Chapter 8 Conclusion

In this dissertation, we proposed a security measure called “Command Verification” to effectively prevent logical attacks stealing cash from ATMs while harmonizing existing ATM systems and operations. The basic idea of Command Verification is that a controlled peripheral device itself verifies a control command sent from the PC before executing the command to access a protected property. A primary model of Command Verification was also proposed to apply it to various ATM transactions. We also proposed a general application scheme of Command Verification: the implementation model analysis and the implementation design method. When Command Verification is applied to multiple transaction sub-processes in an ATM transaction, there are many items to be considered because of the simplicity of the primary model; preventing a wide range of logical attacks targeting many properties, and harmonizing with existing ATM systems, operations, and peripheral device supply chains. Therefore, we proposed two methods to apply Command Verification to various systems and transactions. One is an implementation model analysis in order to compare the features of the implementation models in a preliminary step to derive proper systems. The other is an implementation design method to reduce the number of candidate systems to be examined in detail with the systematic implementation design steps and guidance. The detailed contributions are as follows.

In chapter 4, we proposed Command Verification to solve issues of existing security measures for control systems that operate based on a controller-actuator model. We also proposed the primary model of Command Verification to apply it to various ATM systems and transactions. Since peripheral devices usually do not have any information to verify a command, two peripheral devices are defined in the model; an information acquiring device and a verified command executing device. The information acquiring device extracts command verification information from input data of the acquiring device and securely transfers the information to the verified command executing device. The verified command executing device verifies a command from the PC with the received information. And we also showed applied system examples of Command Verification for one transaction sub-process, two transaction sub-processes, and all transaction sub-processes in an ATM transaction.

In chapter 5, practical effects of Command Verification and the existing measures were compared in an application of Command Verification to one transaction sub-process in an ATM transaction, namely, the cash handling sub-process in a cash

withdrawal transaction with a smart card. Three conditions to effectively prevent unauthorized cash withdrawal in existing ATM operations were derived from analysis of existing ATM systems and operations. It was shown that Command Verification can meet the three conditions while the existing measures do not meet them.

In chapter 6, an application of Command Verification to two transaction sub-processes in an ATM transaction, the issue of the application, and the proposed solution were described. There are multiple properties to be protected from multiple attack surfaces in the transaction sub-processes, and constraints to harmonize with existing systems and operations. It is difficult to design properly implementable systems of Command Verification to meet the requirements. Thus, we proposed an implementation model analysis to select preferable implementation models of Command Verification by comparing the features of the models, whose models are abstract models derived from the primary model of Command Verification. In the application to the two transaction sub-processes in a deposit transaction with a smart card, two recommended implementation models were derived from the model analysis. And two types of properly implementable systems were finally derived using the recommended models. The management cost of the properly implementable system can be reduced to less than one ten-thousandth of the existing measures in the evaluation.

In chapter 7, an application of Command Verification to all transaction sub-processes in an ATM transaction, the issue of the application, and the proposed solution were described. Command Verification should be applied to all transaction sub-processes in a cash withdrawal transaction with a magnetic stripe card since there are few existing security mechanisms. However, there are many implementable systems of Command Verification due to the poor existing security mechanisms. It is difficult to derive proper systems among the many implementable systems, since the proper systems should meet many conditions; preventing a wide range of logical attacks, harmonizing with existing ATM operations, and minimizing modification costs of peripheral devices, which is related with supply chains. We proposed a systematic implementation design method of Command Verification to derive proper systems, which consists of three steps and guidance. Three proper systems out of the 135 implementable systems were selected by applying the design method to magnetic stripe card transactions. That is, the number of candidate systems to be examined in detail was reduced to one forty-fifth.

We expect that Command Verification can be applied not only to ATMs, but also to such devices operating with payment transactions as ticketing machines and vending

machines, IoT systems such as smart home, automobile, robots, and industrial control systems. In these systems, controllers are expected to have complicated structures and functions, to be frequently updated, to be physically accessed for maintenance, and even not to be properly managed by administrators. Thus, it could be quite difficult to effectively and efficiently protect such controllers from logical attacks, and Command Verification is expected to work properly as a defense in depth in such situations. Those are future works to be tackled.

.

Bibliography

- [1] Europol, "Guidance and recommendations regarding logical attacks on ATMs," November 2015. [Online]. Available:
https://www.ncr.com/content/dam/ncrcom/content-type/brochures/EuroPol_Guidance-Recommendations-ATM-logical-attacks.pdf.
- [2] Trend Micro Forward-Looking Threat Research (FTR) Team and Europol's European Cybercrime Centre (EC3), "'Cashing in on ATM Malware,'" 26 September 2017. [Online]. Available:
<https://www.europol.europa.eu/publications-documents/cashing-in-atm-malware>. [Accessed 9 February 2020].
- [3] NCR Corporation, "ATM SECURITY EXPLAINING ATTACK VECTORS, DEFENSE STRATEGIES AND SOLUTIONS," 2018. [Online]. Available:
https://www.ncr.com/content/dam/ncrcom/content-type/white_papers/12518fin-b-atm_security_attack_vectors_and_solutions_update-fin-web.pdf. [Accessed 07 July 2019].
- [4] Symantec, Press Release, Symantec Security Response, "Backdoor.Padpin," October 2014. [Online]. Available:
https://www.symantec.com/security_response/writeup.jsp?docid=2014-051213-0525-99&tabid=2.
- [5] Kaspersky Lab., Press Release, "Tyupkin Virus (Malware) | ATM Security," [Online]. Available:
<https://www.kaspersky.com/resource-center/threats/tyupkin-malware-atm-security-malware>.
- [6] Symantec Official Blog, "Backdoor.Ploutus Reloaded – Ploutus Leaves Mexico," 25 October 2013. [Online]. Available:
<https://www.symantec.com/connect/blogs/backdoorploutus-reloaded-ploutus-leaves-mexico>.
- [7] The Times of India, TIMES NATION | Politics & Policy, "ATM JACKPOT WITH MALWARE," May 2015. [Online]. Available:
<http://www.pressreader.com/india/the-times-of-india-mumbai-edition/20150509/282003260992233>.

- [8] Europol, "27 arrested in successful hit against ATM Black Box attacks," May 2017. [Online]. Available:
<https://www.europol.europa.eu/newsroom/news/27-arrested-in-successful-hit-against-atm-black-box-attacks>.
- [9] The European Association for Secure Transactions (EAST), "EAST reports 2016 crime stats for Europe's ATMs; black box attacks up 287 percent," April 2017. [Online]. Available:
<https://www.atmmarketplace.com/news/east-reports-2016-crime-stats-for-europes-atms-black-box-attacks-up-287-percent/>.
- [10] China Zhijian Publishing House, "GA 1280-2015, Security requirements for automatic teller machines (in Simplified Chinese)," 2015. [Online]. Available:
http://fsms.bsmi.gov.tw/cat//opac_book/book_detail.asp?systemno=0000296088.
- [11] ATM marketplace, "ATMs left behind as Windows XP support ends," April 2014. [Online]. Available:
<http://www.atmmarketplace.com/articles/atms-left-behind-as-windows-xp-support-ends/>.
- [12] B. G. a. J. S. J. Bräuer, "A Risk Assessment of Logical Attacks on a CEN/XFS-based ATM Platform," International Journal on Advances in Security, Vol. 9, No 3&4, pp. 122–132, December 2016.
- [13] Diebold Nixdorf, "How crime can undermine the convenience of cash," 24 June 2015. [Online]. Available:
<https://www.atmia.com/whitepapers/how-crime-can-undermine-the-convenience-of-cash/86/>. [Accessed 09 February 2020].
- [14] PCI Security Standards Council, "Payment Card Industry (PCI) PIN Transaction Security (PTS) Point of Interaction (POI) Modular Security Requirements Version 5.1," March 2018. [Online]. Available:
https://www.pcisecuritystandards.org/documents/PCI_PTS_POI_SRs_v5-1.pdf.
- [15] PCI Security Standards Council, "Payment Card Industry (PCI) PIN Security Requirements Version 2.0," December 2014. [Online]. Available:
https://www.pcisecuritystandards.org/documents/PCI_PIN_Security_Requirements_v2__Dec2014_b.pdf.
- [16] International Organization for Standardization, "ISO 9564-1:2017, Financial services -- Personal Identification Number (PIN) management and security -- Part 1: Basic principles and requirements for PINs in card-based systems," November

2017. [Online]. Available: <https://www.iso.org/standard/68669.html>.
- [17] International Organization for Standardization, "ISO 9564-2:2014, Financial services -- Personal Identification Number (PIN) management and security -- Part 2: Approved algorithms for PIN encipherment," August 2014. [Online]. Available: <https://www.iso.org/standard/61448.html>.
- [18] International Organization for Standardization, "ISO 9564-4:2016, Financial services -- Personal Identification Number (PIN) management and security -- Part 4: Requirements for PIN handling in eCommerce for Payment Transactions," March 2016. [Online]. Available: <https://www.iso.org/standard/61246.html>.
- [19] PCI Security Standards Council, "Payment Card Industry (PCI) PIN Transaction Security (PTS) Hardware Security Module (HSM) Modular Security Requirements Version 3.0," June 2016. [Online]. Available: https://www.pcisecuritystandards.org/documents/PCI_HSM_Security_Requirements_v3_2016_final.pdf.
- [20] Comité Européen de Normalisation, "Extensions for Financial Services (XFS) interface specification Release 3.30 - Part 1: Application Programming Interface (API) - Service Provider Interface (SPI) - Programmer's Reference," August 2015. [Online]. Available: <ftp://ftp.cen.eu/CWA/CEN/WS-XFS/CWA16926/CWA%2016926-1.pdf>.
- [21] EMVCo, LLC, "EMV Integrated Circuit Card Specifications for Payment Systems Book 2 Security and Key Management Version 4.3," November 2011. [Online]. Available: https://www.emvco.com/terms-of-use/?u=wp-content/uploads/documents/EMV_v4.3_Book_2_Security_and_Key_Management_20120607061923900.pdf.
- [22] EMVCo, LLC, "EMV Integrated Circuit Card Specifications for Payment Systems Book 3 Application Specification, Version 4.3," November 2011. [Online]. Available: https://www.emvco.com/wp-content/uploads/documents/EMV_v4.3_Book_3_Application_Specification_20120607062110791.pdf.
- [23] ATM Marketplace, "Managing ATM Security: Layered Approaches for 21st Century Issues," 21 June 2017. [Online]. Available: <https://www.atmmarketplace.com/whitepapers/managing-atm-security-layered-approaches-for-21st-century-issues/>. [Accessed 29 12 2019].
- [24] Diebold, "Diebold Solutions Corporate and ATM security," 2009. [Online].

- Available:
<http://rfinance.ru/upload/files/CS%20Moscow%20Summit%20Presentation%20Final.pdf>. [Accessed 29 12 2019].
- [25] NCR Corporation, "Six types of ATM attacks and fraud," 09 July 2015. [Online]. Available:
<https://www.ncr.com/company/blogs/financial/six-types-of-atm-attacks-and-fraud>. [Accessed 29 12 2019].
- [26] The European Union Agency for Cybersecurity, "ATM Crime: Overview of the European situation and golden rules on how to avoid it," 07 September 2009. [Online]. Available:
<https://www.enisa.europa.eu/publications/archive/atmcrime>. [Accessed 29 12 2019].
- [27] FS-ISAC, "Understanding ATM Attacks," FS-ISAC, 28 August 2018. [Online]. Available:
https://www.fsisac.com/hubfs/5442200/Resources/FS-ISAC_Understanding_ATM_Attacks.pdf. [Accessed 29 12 2019].
- [28] Securonix, "Securonix Threat Research: Cosmos Bank SWIFT/ATM US\$13.5 Million Cyber Attack Detection Using Security Analytics," August 2018. [Online]. Available:
<https://www.securonix.com/securonix-threat-research-cosmos-bank-swift-atm-us13-5-million-cyber-attack-detection-using-security-analytics>.
- [29] The National Cybersecurity and Communications Integration Center (NCCIC), "Alert (TA18-275A) HIDDEN COBRA – FASTCash Campaign," December 2018. [Online]. Available:
<https://www.us-cert.gov/ncas/alerts/TA18-275A>.
- [30] PCI Security Standards Council, "Payment Card Industry (PCI) Point-to-Point Encryption: Solution Requirements and Testing Procedures Version 3.0," December 2019. [Online]. Available:
https://www.pcisecuritystandards.org/documents/P2PE_v3.0_Standard.pdf.
- [31] S. Kai, T. Ishikawa, H. Ogata and T. Sanada, "Accelerating Global Business through ATM Security Practices," Information Processing Society of Japan, IPSJ Digital Practice Vol.9 No.3, pp.700-715, 2018.
- [32] "The Common Criteria," [Online]. Available:
<https://www.commoncriteriaportal.org/>. [Accessed 30 12 2019].

- [33] "Bull-Dat-Diebold-NCR-Siemens-Nixdorf-Wang-Global: Automatic Cash Dispensers/Teller Machines Protection Profile Version 1.0, PP/9907," 02 March 1999. [Online]. Available:
<https://www.commoncriteriaportal.org/files/ppfiles/PP9907.pdf>. [Accessed 30 12 2019].
- [34] "Common Approval Scheme: Point of Interaction Protection Profile Version 2.0," 26 November 2010. [Online]. Available:
https://www.ssi.gouv.fr/uploads/IMG/certificat/ANSSI-CC-cible_PP-2010-10en.pdf. [Accessed 30 12 2019].
- [35] APICS, "SCOR Framework," [Online]. Available:
<http://www.apics.org/apics-for-business/frameworks/scor>. [Accessed 30 12 2019].
- [36] International Organization for Standardization, "ISO 31000:2018(en) Risk management — Guidelines," 2018. [Online]. Available:
<https://www.iso.org/obp/ui/#iso:std:iso:31000:ed-2:v1:en>. [Accessed 30 12 2019].
- [37] H. Ogata, T. Ishikawa, N. Miyamoto and T. Matsumoto, "An ATM security measure for smart card transactions to prevent unauthorized cash withdrawal," IEICE TRANSACTIONS on Information and Systems, Vol.E102-D, No.3, pp.559-567, 2019.
- [38] International Organization for Standardization, "ISO 11568-1:2005, Banking -- Key management (retail) -- Part 1: Principles," June 2015. [Online]. Available:
<https://www.iso.org/standard/34937.html>.
- [39] International Organization for Standardization, "ISO 11568-2:2012, Financial services -- Key management (retail) -- Part 2: Symmetric ciphers, their key management and life cycle," February 2012. [Online]. Available:
<https://www.iso.org/standard/53568.html>.
- [40] International Organization for Standardization, "ISO 11568-4:2007, Banking -- Key management (retail) -- Part 4: Asymmetric cryptosystems -- Key management and life cycle," July 2007. [Online]. Available:
<https://www.iso.org/standard/39666.html>.
- [41] American National Standards Institute, "ANSI X9.24-1-2017: Retail Financial Services Symmetric Key Management Part 1: Using Symmetric Techniques," June 2017. [Online]. Available:
<https://webstore.ansi.org/Standards/ASCX9/ANSIX9242017?source=blog>.
- [42] American National Standards Institute, "ANSI X9.24-2-2016, Retail Financial

Services Symmetric Key Management Part 2: Using Asymmetric Techniques for the Distribution of Symmetric Keys," November 2016. [Online]. Available:
<https://webstore.ansi.org/Standards/ASCX9/ANSIX9242016>.

- [43] H. Ogata, T. Ishikawa, N. Miyamoto, T. Matsumoto, "An ATM security measure to prevent unauthorized deposit with a smart card," IEICE TRANSACTIONS on Information and Systems, Vol.E103-D, No.03, 2019.
- [44] H. Ogata, T. Ishikawa, N. Miyamoto, T. Matsumoto, "Secure ATM Device Design by Control Command Verification," In: Shankar Sriram V., Subramaniaswamy V., Sasikaladevi N., Zhang L., Batten L., Li G. (eds) Applications and Techniques in Information Security. ATIS 2019. Communications in Computer and Information Science, vol 1116. pp.32-50, Springer, 2019.
- [45] 緒方日佐男, 石川智祥, 宮本範親, 松本勉, "コマンド真正性検証を用いたセキュアな ATM 設計法," 情報処理学会論文誌, Vol.61, 2020.
- [46] NCR Corporation, "Transaction Reversal Fraud - Global," 9 July 2018. [Online]. Available:
<https://www.ncr.com/content/dam/ncrcom/content-type/brochures/NCR%20Security%20Alert%20-%202018-06%20Transaction%20Reversal%20Fraud.pdf>.
- [47] NCR Corporation, "Cash Trapping "Type 1" Attacks in Spain," December 2016. [Online]. Available:
https://www.ncr.com/content/dam/ncrcom/content-type/brochures/ncr_security_alert_-_2016-14_cash_trapping_in_spain_0.pdf.
- [48] THE EUROPEAN ATM SECURITY TEAM, "EUROPEAN ATM CRIME REPORT 2014," 08 April 2015. [Online]. Available:
<https://www.association-secure-transactions.eu/files/EAST-ATM-Crime-Report-2014.pdf>.
- [49] Retail Banking Research Ltd., Deposit Automation and Recycling 2016, Retail Banking Research Ltd, 2016.
- [50] IOActive, Inc., "IOActive Security Advisory, Physical and Authentication Bypass in Diebold Opteva ATM," 26 July 2017. [Online]. Available:
https://ioactive.com/pdfs/ATM_security-advisory_FINAL_v4-davis_cm.pdf.
- [51] S. E. S. O. B. Daniel Regalado, "Backdoor.Ploutus Reloaded – Ploutus Leaves Mexico," October 2013. [Online]. Available:
<https://www.symantec.com/connect/blogs/backdoorploutus-reloaded-ploutus-leaves-mexico>.

- [52] American National Standards Institute, "ANSI X9.24-3-2017, Retail Financial Services Symmetric Key Management Part 3: Derived Unique Key Per Transaction," October 2017. [Online]. Available:
<https://webstore.ansi.org/Standards/ASCX9/ANSIX9242017-1665702>.
- [53] American National Standards Institute, "ANSI X9.63-2011 (R2017), Public Key Cryptography for the Financial Services Industry - Key Agreement and Key Transport Using Elliptic Curve Cryptography," February 2017. [Online]. Available: <https://webstore.ansi.org/Standards/ASCX9/ANSIX9632011R2017>.
- [54] PCI Security Standards Council, "Payment Card Industry (PCI) Data Security Standard Requirements and Security Assessment Procedures Version 3.2.1," May 2018. [Online]. Available:
https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2-1.pdf.
- [55] European Association for Secure Transactions Ltd (EAST), "TERMINAL PHYSICAL ATTACK DEFINITIONS & TERMINOLOGY (ATM & ATS)," 08 March 2019. [Online]. Available:
<https://www.association-secure-transactions.eu/files/EAST-Terminal-Physical-Attack-Definitions-Terminology-ATM-ATS-.pdf>. [Accessed 29 12 2019].

List of Papers

Reviewed Papers in Journals

1. S. Kai, T. Ishikawa, H. Ogata, N. Miyamoto, T. Sanada, "Accelerating Global Business through ATM Security Practices", IPSJ Digital Practice Vol.9 No.3, pp.700-715, July 2018
https://ipsj.ixsq.nii.ac.jp/ej/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=190389&item_no=1&page_id=13&block_id=8
2. H. Ogata, T. Ishikawa, N. Miyamoto and T. Matsumoto, "An ATM security measure for smart card transactions to prevent unauthorized cash withdrawal", IEICE TRANSACTIONS on Information and Systems, Vol.E102-D, No.3, pp.559-567, 2019,
DOI: 10.1587/transinf.2018EDP7136

3. H. Ogata, T. Ishikawa, N. Miyamoto and T. Matsumoto, “An ATM security measure to prevent unauthorized deposit with a smart card”, IEICE TRANSACTIONS on Information and Systems, Vol.E103-D, No.03, pp.590-601, Mar. 2020.
DOI: 10.1587/transinf.2019EDP7143
4. 緒方日佐男, 石川智祥, 宮本範親, 松本勉, “コマンド真正性検証を用いたセキュアな ATM 設計法”, 情報処理学会論文誌, Vol.61, No.4, 1–12, Apr. 2020.

Reviewed Papers in International Conference Proceedings

5. Ogata H., Ishikawa T., Miyamoto N., Matsumoto T. “Secure ATM Device Design by Control Command Verification”. In: Shankar Sriram V., Subramaniaswamy V., Sasikaladevi N., Zhang L., Batten L., Li G. (eds) Applications and Techniques in Information Security. ATIS 2019. Communications in Computer and Information Science, vol 1116. pp.32-50, Springer, Singapore, 2019,
https://doi.org/10.1007/978-981-15-0871-4_3

Technical Reports

6. 緒方 日佐男, 石川 智祥, 宮本 範親, 松本 勉, ”デバイス間暗号通信を用いた ATM 不正入金・送金取引防止対策”, 電子情報通信学会 ハードウェアセキュリティ研究会 (HWS 研究会) 2018 年度 3 月研究会, 2018 年
7. 緒方 日佐男, 石川 智祥, 宮本 範親, 松本 勉, ”被制御デバイスによる制御コマンドの真正性検証方式の提案”, 電子情報通信学会 2019 年 暗号と情報セキュリティシンポジウム SCIS2019, セッション 2E4-1, 2019 年

Award

ATIS 2019 Best Paper Award

“Secure ATM Device Design by Control Command Verification”, November 2019