Security Analysis of the Octopus System

Andrew Lee, Timothy Lui, and Bryon Leung

***Abstract*— As Octopus Cards Limited deals with a large amount of money and clients everyday, they are greatly concerned with the integrity, availability, and confidentiality of the Octopus System. To achieve these security policies, many security measures are employed by Octopus, i.e. authentication, encryption, etc. It is very important for both the front-end and back-end of the Octopus System to be protected against various possible threats. After conducting a detailed analysis, we conclude that the security policies mentioned above are well enforced by the Octopus System as its security measures are adequate.**

I. INTRODUCTION

IN 1997, Octopus Cards Limited introduced a contact-less smart-card based electronic cash (e-cash) system – the Octopus System. It was originally intended to provide a quick and easy way to pay fares on public transits in Hong Kong. Over the past few years, Octopus has significantly extended its range of applications. Cardholders can use Octopus to make payments at stores and restaurants; gain access to buildings and schools; identify themselves.

Similar to other e-cash systems, such as VisaCash and Mondex, one of the major security concerns of the Octopus system lies with integrity. In particular, only authorized and non-fraudulent cards shall operate with the Octopus System. Moreover, confidentiality is of great concern for Octopus cardholders. Since their personal information is stored in the system, strong security mechanisms are required to ensure data are well protected. Last but not least, availability is another security concern. If the system fails to operate, there will be major losses for both the Octopus Cards Limited and its clients.

To improve the integrity of the Octopus System, the mutual authentication mechanism is employed. This authentication mechanism requires two parties authenticating each other suitably. The card first authenticates itself to the reader and the reader then authenticates itself to the card in such a way that both parties are assured of the others' identity. The triple DES encryption algorithm is used to enforce the confidentiality of the Octopus System. Physical protection mechanisms at both the front-end and back-end assure the availability of the Octopus System.

The security of the Octopus System is especially important since the assets involved are huge. With an average daily transaction of 10 million dollars US, a low-scale security breach would lead to significant loses. Also, successful attacks on the system would reduce the confidence level of its clients, which include 6.5 million end-users and 440 service providers. From the Octopus end users' perspective, a security breach of the Octopus System will put their money and personal information in danger. More importantly, if a stolen card is used for access control purposes, further damages could be induced.

With so many assets at risk, keeping the Octopus System secure is absolutely vital. In this paper, we will analyze both the front-end (Section II) and back-end (Section III) of the system.

II. THE FRONT-END SECURITY OF THE OCTOPUS SYSTEM

A. Description of FeliCa

FeliCa is a radio-frequency identification (RFID) Integrated Circuit (IC) chip smart-card system developed by Sony. It currently meets the ISO 18092 standard which is also known as Near Field Communication. The card is a passive device that is only powered by the electro-magnetic waves emitted from the reader/writer. The operating distance of FeliCa is about 10cm and the hardware allows transactions to be completed in as short as 0.1 second. The card consists of three components: shell, RF antenna, and IC chip.

1) Shell

The shell of the FeliCa card is of the same size as a conventional credit card (ISO/IEC 7810ID-1). The material used is polyethylene terephthalate (PET) plastic.

2) RF Antenna

The RF antenna of the FeliCa card is a film antenna that operates at the frequency of 13.56 MHz. It allows wireless data transfer at the rate of 212 kbps.

3) IC Chip

The IC chip is an 8-BIT SONY reduced instruction set computer (RISC) CPU. It is specifically designed for encryption and random number generation purposes. The memory of the chip provides both volatile and non-volatile memory. Most notably, it has a 4KB EEPROM where user information, such as money value or identification information, is stored.

B. Possible threats on Front-End

Being the most exposed component of the Octopus System, the Octopus cards are vulnerable to two major types of possible threats.

1) Physical Attacks

This type of attack is performed directly on the Octopus cards. Since the cards are more portable and readily available than the readers, they are also more vulnerable to the following physical attacks.

a) Stealing and Switching of Card

Also known as “pickpocketing”, this is one of the most common attacks of all. If succeeded, attackers will be able to extract money from the stolen card as if they are the original owners since the Octopus System does not employ any user authentication mechanism.

b) Probing of IC

By probing the IC, attackers might be able to reveal the design and implementation of the Octopus System. In such case, the system will become defenseless to many other attacks.

c) Modifying of IC

In this attack, the IC of the FeliCa card is altered in order to perform fraudulent actions. It requires sophisticated equipments and professional skills.

d) Environmental Stress Attack

The environmental stress attack occurs when the attacker exposes the IC to conditions outside of its specified operating range. This might lead to malfunctioning of the card.

2) Logical Attacks

Besides the possible threats described above, attacks can also be done at the logical level.

a) Brute Force Attack

This attack exhaustively works through all the possibilities in order to decrypt a cryptographic scheme. An example of such attack is demonstrated in the following case study.

C. Case Study: Speedpass

In preparation to assess the security risks of the Octopus System, we have conducted a case study on a previously hacked contact-less e-cash system – Speedpass. With the results of this case study, we will be able to analyze the security strength and weakness of the Octopus System more effectively.

1) Description of Speedpass

Speedpass, developed by ExxonMobil in 1997, is a contact-less payment system which implements the Texas Instruments Radio Identification System (TIRIS) 134.2kHz DST tag system. The system consists of a tag, reader, and central computers at the back-end. Stored on every tag are an ID and a unique key for authentication and encryption purposes. Other information is stored in the central computers for greater security. Speedpass is ISO 14443 compliant as with most RFID payment systems. When a tag is brought into proximity of a reader, the one-way authentication process will be initiated. From this point on, every data transmission is encrypted with a 40-bit encryption key. The encryption algorithm is kept secret by the company, which violates the security principle of open design. After the reader validates the identity of the tag, it then communicates with the back-end to determine further instructions to be executed. [18]

2) Failure of Speedpass

A team of researchers from Johns Hopkins University's Information Security Institute succeeded in hacking Speedpass by employing the reverse engineering techniques. They had to uncover the encryption algorithm and capture a known challenge/response pair in order to run a brute force attack to look for the key that provided the response. [18]

First, the team obtained an evaluation kit and some DST tags from ExxonMobil. They used the software provided with the reader to collect challenges and responses. As this collection grew, clues to the encryption process started to appear. Then the team employed a “black-box” method to figure out the details of the algorithm. For every chosen input, they observed the corresponding output and constructed a process that would produce the same output as the “black-box”. This method avoided any legal issues because the team did not violate any Non-Disclosure Agreements (NDA’s). With a rough diagram of the encryption algorithm, the team was able to fill in the missing parts by mapping out the relationships between the input and output bits. After they have reverse-engineered the internal mathematics of the DST tag, they began to brute force the key for that tag. The team used a Field Programmable Gate Array (FPGA) for this task as its computer processor is reprogrammable. Each FPGA was able to test 32 keys at once in parallel. They programmed and built an array of 16 FPGA’s working in parallel that, given two challenge/response pairs, recovered the key in under an hour.

The attack allowed the team to clone DST tags. With the cloned tags, the team was able to start a DST tag equipped vehicle and purchase fuel at several ExxonMobil gas stations. [18]

D. Security Measures of the FeliCa Card System

The extensive security safeguards that are implemented into the FeliCa card allows it to be certified by ISO/IEC15408 EAL4 – the most reliable criteria to measure security level of a system. In this section, we will discuss the specific security features of the card.

1) Plastic Carrier

The plastic carrier of the card acts as a physical measure of security by encapsulating the delicate internals - the IC chip and RF antenna - from the outside world. The carrier to some extend prevents direct access to the IC chip and therefore hides its physical design. Also, the carrier houses sensors that detect IC removal. Therefore the card will cease to function if the IC has been removed and replaced. Finally, under normal circumstances, the carrier preserves the integrity of the physical structure of the card device.

2) IC Software Support

The IC chip of this card does not support software download. Therefore, the software could not be modified once it was loaded onto the chip during manufacturing. This is an important feature since it will reduce any chance of inserting malicious logic in the software of the IC chip.

3) Data Protection

The data on the card is very important since the cash value is stored directly in the memory of the card. Before any transaction the card makes with the reader, the data stored on the card is always written to a non-volatile backup memory block. If a transaction is interrupted during writing, the card will first remove all changes made by the write command. Then, the backup data is immediately used to restore the data previously on all system and user memory blocks of the card. This feature prevents malicious attempts of interfering with transactions and thus inducing inconsistency in the data on the card. Also, it ensures that every transaction is either cancelled or successfully completed. Another feature that protects the data is that the card utilizes check sums for the detection of corruption of data stored in the memory [7]. This will prevent any chance of changing the data on the card as a result of data being corrupted due to intentionally or accidentally interrupted transactions. The above features are especially important since the IC chip and antenna relies on the power of the reader and occasional power loss during writing transactions is unavoidable.

4) *Card Information Protection*

The information regarding the settings of the card is always protected by a special key from the time it is manufactured to the time it reaches end users. An IC manufacturing key must be supplied any time information in cards needs to be changed [7] during the production of the card. Before the card is shipped to the Octopus, the manufacturing key is replaced by an IC card shipping key that protects information on the card during transportation. When the cards reach Octopus, it can then replace this key with their own key before configuring the card for its usages. Any subsequent modifications to the core information and settings on the card must be done after supplying this key.

5) *Memory Structure*

The memory structure of the card is built with security in mind. The memory structure provides secure storage for user data and associates security attributes with the stored data [7]. These attributes limit both the accessibility and the method of access to a particular block of data on the card. There are three types of files: Random Access files, Cyclic Access files and Purse Access files. Random Access files allow users to randomly access any block by specifying the location of the block. Cyclic Access files have blocks that are arranged in chronological order [7], therefore, users can only access them in a certain order. When accessing Purse Access files, users are limited to a small set of special functions such as adding a value to a specific block that contains a data value. On top of the different access modes, every file or file block can be specified to be secure, insecure, read-only or read/write-able. Finally, every block can be protected by an access key. For the Octopus application of the FeliCa card, the data and instruction sets used by Octopus-related transactions are stored in secure Purse Access files that are protected by a special Octopus key.

E. *Security of Communication between Card and Reader/Writer*

Besides assessing the countermeasures on the card itself, it is also important to analyze the security of the communication between the card and the reader/writers.

1) *Overview*

The communication protocol between the Octopus cards and readers employs an improved system derived from the 3-way handshake defined in ISO9798 [7]. The wireless communication path between the reader/writer and the card is also protected by a pair of transaction ID and transaction key that are randomly generated at the beginning of every session.

2) *Encryption Method*

The encryption method used during the mutual authentication is the triple DES system. This system greatly improves the security of the data and data transfer of the system. Basically, in this encryption system, DES is applied to the same plaintext three times using three different keys. As a result, the overall key, which is the combination of all three keys, is much longer and the effective security is also much greater than DES. The following diagram depicts the triple DES system graphically.

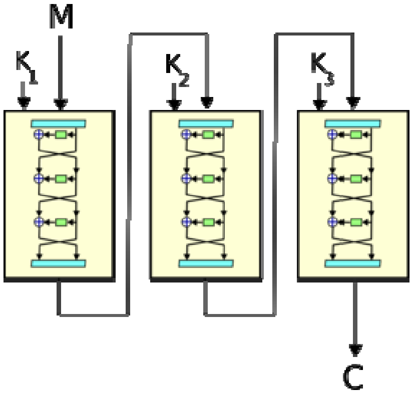


Fig. 1. Triple DES Encryption

In this diagram, M is the plaintext and C is the cipher-text. K1,K2,K3 are the three keys and the blocks represents a DES encryption block. All the encryptions, including both the data on the card and the data packets sent during the communication between the card and the readers, employ this method of encryption.

3) *Communication Protocol*

Since the actual communication protocol between the card and the reader/writer has not disclosed by Sony or Octopus, we have decided to produce a reconstruction of the actual communication protocol based on the information presented in section 2.1 and 2.2. The protocol that we constructed is shown in the following diagram.

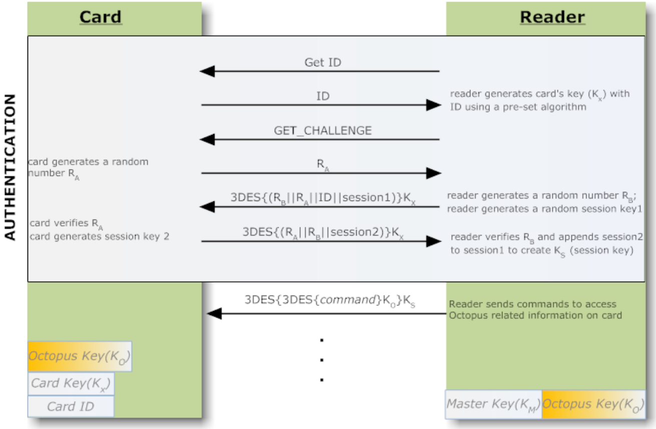
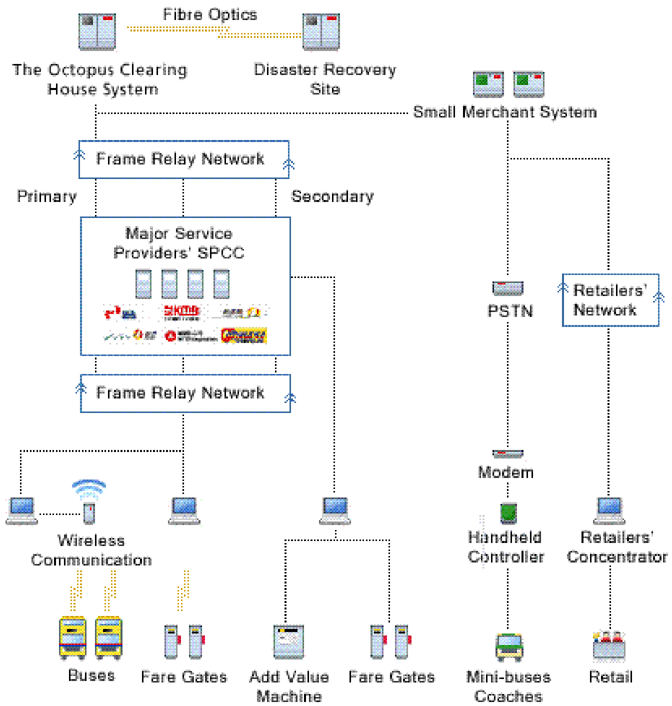

Card
Reader
AUTHENTICATION
Get ID
ID
reader generates card's key (K_x) with ID using a pre-set algorithm
GET_CHALLENGE
card generates a random number R_A
R_A
3DES{(R_B||R_A||ID||session1)}K_x
reader generates a random number R_B; reader generates a random session key1
card verifies R_A
card generates session key 2
3DES{(R_A||R_B||session2)}K_x
reader verifies R_B and appends session2 to session1 to create K_S (session key)
3DES{3DES{*command*}K_O}K_S
Reader sends commands to access Octopus related information on card
Octopus Key(K_O)
Card Key(K_x)
Card ID
Master Key(K_M)
Octopus Key(K_O)

Fig. 2. Octopus System Communication Protocol

In this diagram, we have a valid Octopus card and a valid Octopus reader. The card has a unique card ID (ID) and also a unique card key (K_X). Furthermore, the files containing the data and instructions that are used in Octopus-related transactions are protected by an Octopus key (K_O). The reader has a master key (K_M) and it also has the Octopus key. The reader is continuously polling for a card and when a card is in range, the authentication process begins. The reader will first request for the card ID of the card. When it receives the ID, it will generate K_X from ID using a pre-set algorithm. Then, the reader will send a request for a challenge from the card. The card will generate a random number, R_A, and send that to the reader unencrypted. The reader will then generate two more random numbers, R_B and S_1. These numbers along with R_A will be encrypted with K_X and be sent back to the card. The card will then decrypt the block and it will first retrieve R_A and verify that R_A is correct. Then it will generate another random number S_2 and send $R_A \| R_B \| S_2$ encrypted with K_X back to the reader. Likewise, the reader will decrypt this block and verify that R_B is correct. When both the card and the reader finish verifying that each other correctly encrypted and decrypted the random numbers R_A and R_B, the mutual authentication process is completed. S_1 and S_2 can then be used as the transaction ID and transaction key for any future communication in this transaction. The transaction ID is appended to the beginning of every message and the transaction key is used to encrypt the message. The transaction ID can ensure that messages in one session can never be reused in a different session. Besides transaction key, the Octopus key must also be used for every Octopus related transactions since as mentioned before, the Octopus data and instructions stored on the card are protected by this key. This key distinguishes valid Octopus cards and readers from any other FeliCa cards or readers. At the end of every communication sessions, in order to improve the security level of the system, the transaction key is used as a key disposable [7].

III. The Back-End Security of the Octopus System

Behind Octopus cards and readers, there is a wide range of communication in different forms. In addition to the security of the transport of the transaction data, an extensive audit system is built for detection of human errors or possible attacks.



A. Description of the Octopus Clearing House System

Fig. 3. Architecture of the Octopus Clearing House System

The back-end of the Octopus Card System divides into four levels. The lowest level is the Front-End Smart Card Processor. This level includes all online and offline smart card readers. Next level is Local Data Processor. This level includes devices such as handheld controllers, modems and personal computers. These devices collect transaction data in readers and serve as transmitting-end of the back-end communication. Next level is Service Provider Central Computer. For major service providers, the volume of data collected each day is huge. This level serves as an intermediate point for organizing data within the same service provider before the data are sent to the final level. The last level is Octopus Clearing House System. This level is the receiving-end of all data transmission. All transaction data are checked in the audit and reconciliation process. A standard set of reports is then sent to each participating service provider on a daily basis. [14]

B. Data Communication and vulnerabilities between Front-end Smart-Card-Processor and Local-Data-Processor

There are three types of communication after data reaches a card reader. They are in form of wireless communication, local area network and manual transmission via portable devices.

1) Wireless communication

Buses from major service providers are equipped with online Octopus Card readers with wireless data transmission utility. They are activated by trained personnel when the reader arrives at a location equipped with the receiving-end of the wireless communication, such as a terminal bus stop. The routine of uploading data are automatic once it is activated. [14] The communication protocol information is restricted to Octopus Card Limited and its service providers, but it is believed that it complies with the 802.11 legacy standard. It operates at 2.4-2.5 Frequency at a transmission rate of 1Mb/s with a range of approximately 75 meters. [17]

This method of communication is highly vulnerable to possibility of sniffing attacks, thus its security depends on the encryption of the data transmission.

2) Local Area Network

Online Octopus Card readers in railway stations are connected by Local Area Network inlaid within walls or underground of the local structure. The locations of these readers are public and crowded and, thus, they are nearly impossible to attack without drawing significant attention. Data read by these readers are transmitted through Local Data Processor and reaches the next level, Service Provider Central Computer, in real time. Service Provider Central Computer is physically secured and can only be accessed by restricted personnel. [14] This makes online Octopus Card reader in railway stations the most secured of all 3 kinds of readers.

3) Offline Readers and Other Related Devices

Offline readers can store transaction data locally. The data are collected manually by a hand held controller supplied by Octopus Card Limited. The device is then connected to a modem that can be located at personal space to transmit transaction data to the Octopus Clearing House System through public switch telephone network. [14] This method is the most vulnerable of all 3 types of communication as all of the involved devices – the offline reader, handheld controller, and modem – are subject to modification.

C. Data Communication and Vulnerabilities between Local Data Processor and Octopus Clearing House System

There are two form of communication between each Local Data Processor and Octopus Clearing House System: Frame Relay Network and Public Switch Telephone Network (PSTN).

1) Frame Relay Network

Frame relay is a technology offering virtual private-line replacement. It has evolved into a network interface with its own specified set of ANSI and CCITT standards. Between major service providers and Octopus Clearing House System, the network employed is a wide area network (WAN) with transmission rates at 64 Kbps. The frame relay wide area network delivers highly reliable, fast, and efficient data transmissions. [9]

2) The Frame Relay Packet

Each frame relay access station is responsible for transforming the data into frame relay for transport over the network. Each frame has the following format:

Flag – Indicates the start and end of a frame relay packet

Frame Relay Header – Contains the destination of the user data packet, defined by a data link control identifier, and management information

User Data – Contains the data to be transported across the frame relay network

FCS – Frame Check Sequence allows the integrity of the data to be validated

The frame relay network receives transports and delivers variable-length frames. Figure 1 shows the frame relay packet and the frame handling by switches.

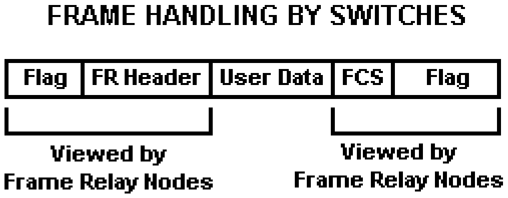

FRAME HANDLING BY SWITCHES
Flag
FR Header
User Data
FCS
Flag
Viewed by Frame Relay Nodes
Viewed by Frame Relay Nodes

Fig. 4. Frame Handling by Switches

There are two major advantages in frame relay network over other network connections, such as internet protocol (IP). Frame relay network offers predictable, reliable and measurable performance. Frame relay network operators pre-program logical circuits in the network and it is clear how much bandwidth is used between connections. When transmitted data are in known format and length, the data will be less vulnerable to modification attacks. Another advantage is that frame relay connections can be connected to a network management system. If a circuit goes down, an alarm goes off. This will prevent most Denial of Service attacks.

However, frame relay network is still vulnerable to sniffing and intercepting attacks. Authentication or other sensitive information can be intercepted.

3) Public Switch Telephone Network

The Public Switch Telephone Network is probably the most often “tapped” means of communication in the world. The risk of exposing sensitive data is further increased by private possession of handheld devices and modems, which is used to transmit offline data directly to the Octopus house clearing system. The security here relies heavily on data encryption.

D. The Audit and Reconciliation Process

According to Octopus Card Limited, the Octopus Clearing House System uses a complex set of business rules to validate each transaction prior to authorising settlement. A standard set of reports is then sent to each participating service provider on a daily basis. The process is strictly confidential to their service providers. But through a known successful attack incident, the audit system proves to be adequate in achieving the security goals of detection.

In November, 2005 a cashier working in a convenient store discovered a “bug” in the process of recharging value to an Octopus card. She first recharged for an anonymous customer any amount (as requested) to the customer's card. Then, she scanned a free shop gifts with the price scanner, which was connected to the cashier and the Octopus card reader. And immediately after that, she recharged a different Octopus card (her own). The second card that was recharged would have both the first and the second recharging value added to the same card.

Her co-worker followed the procedure and together they stole a total of HKD$63,700 between October 27, 2005 and November 17, 2005. The amount were stolen and stored in 91 separate Octopus cards. They were caught by the store manager while he was investigating lost of the store by the Octopus report.

Although the incident does not classify as a security breach, it can be viewed as an example of possible lost in an actual security breach incident. The daily transaction amount flowing to and from Octopus Cards Limited is closely audited. This proves that the system is successfully achieving the security goal of detection.[13]

E. Other Vulnerabilities to the Back-End System

The security of any encryption system consists of three parts – the encryption algorithm, the encryption key, and the cipher-text. It is apparent that most communication protocol employed within the Octopus Card system relies on the secrecy of encryption key and the data sent (random numbers). If the format of the data sent, the range of the random number or the random number generation method is known, the encrypted communication, either between the card and the reader or the reader and the House Clearing System, will be at high risk. This vulnerability can be exploited through “Social Engineering”. Social engineering is a collection of techniques used to manipulate people into performing actions or divulging confidential information. Attacker may harvest useful information mentioned above through this method. [13]

IV. Future Enhancements

After analyzing the security of both the front-end and back-end of the Octopus system, we came up with an enhancement that could be adopted to increase the security of the front-end system. Instead of encrypting with the 3DES system, we can adopt the AES encryption method for both the communication protocol and the information stored on the card. Although 3DES should be sufficient in preventing most attempts of brute force attack, AES encryption method will totally eliminate any chance of such attacks succeeding. Also, the AES algorithm is more efficient and the complexity of implementing this algorithm is comparable to 3DES. Therefore, we believe that switching to AES would be the greatest enhancement to the security of the front-end. As for the back-end of the system, we discovered that much of the vulnerabilities are rooted from the human factors involved instead of the technologies applied. The only enhancement we could come up with is that more automated checking algorithms can be implemented into the system to monitor the usage and return of cards. These algorithms would improve the detection of malicious attempts at an earlier stage before the auditing process.

V.DISCUSSION

There are five goals in computer security: prevention, deterrence, detection, recovery, and investigation. The complete Octopus Card system is adequate in achieving each goal. The security goal of prevention is achieved by the encryption systems employed both in front-end and back-end communications, for example, the mutual authentication that checks for the authenticity of cards and readers. Deterrence and detection is accomplished by the audit and reconciliation system. Recovery and investigation are also covered by different hardware specifications, such as the frame relay network. In addition to the system architecture and hardware and software specifications, there is also a list of policies protecting both users and the company itself that enhances the security of the overall system. (They are not discussed as the technicality is insignificant to the analysis of this paper) Except for minor flaws and bugs over the last 10 years of its usage, the Octopus System has yet to fail to any attack. Despite of the system's age, its security measures are comparable to modern online payment systems employed by the most prestige corporation on the Internet.

REFERENCES

[1] I. Attali, “Smart Card Programming and Security”, International Conference on Research in Smart Cards, E-smart 2001, September 2001.

[2] R. Bright, “Smart Cards: Principles, Practice, Applications”, Ellis Horwood Ltd, 1988.

[3] D. Chaum, “Smart Card 2000”, Elsevier Science Publishers B.V., 1991.

[4] M. Devargas, “Smart Cards & Memory Cards”, National Computing Centre Ltd, 1992.

[5] J. Domingo-Ferrer, “Smart Card Research and Advanced Applications”, 7th IFIP WG 8.8/11.2 International Conference,CARDIS 2006, April 2006.

[6] H. Dreifus, and J. T. Monk, “Smart Cards: A guide to Building and Managing Smart Card Applications,” John Wiley & Sons (UK), 1998.

[7] FeliCa RC-S860 Contactless Smart Card Security Target: http://www.commoncriteriaportal.org/public/files/epfiles/
FeliCaRC.pdf

[8] K. Finkenzeller, “RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition,” John Wiley & Sons (UK), 2003.

[9] Frame Relay Security Guide: http://www.tccsecure.com/fr_Security_Guide.html

[10] A. Haddad, “A New Way to Pay”, Gower Publishing Ltd, 1997.

[11] U. Hansmann, M. S. Nicklous, T. Schack, A. Schneider, and F. Seliger, “Smart Card Application Development Using Java”, Springer-Verlag Berlin Heidelberg, 2002.

[12] M. Hendry , “Smart card security and applications,” Norwood, Mass.: Artech House, 1997.

[13] Hong Kong Judiciary Case # TWCC 1046-2006: http://www.judiciary.gov.hk/tc/index/index.htm

[14] Octopus Card Limited Website: http://www.octopuscards.com/

[15] Overview of FeliCa: http://www.sony.net/Products/felica/abt/dvs.html

[16] W. Rankl, and W. Effing, “Smart Card Handbook”, John Wiley & Sons Ltd, 1997.

[17] Tech Guide: IEEE 802.11: http://www.computerplug.com/tech_guide.php?topic=IEEE+802.11

[18] F. Thornton, “RFID Security”, Syngress Publishing Inc, 2006.

[19] UK IT Security Evaluation and Certification Scheme: http://www.commoncriteriaportal.org/public/files/epfiles/
CRP165.pdf