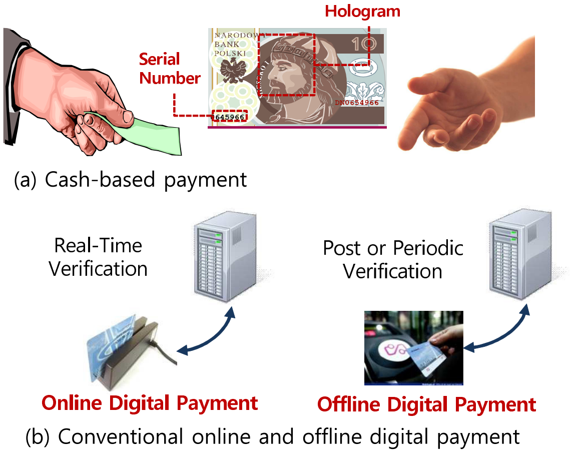
PAPER *Special Section on Security, Privacy and Anonymity in Computation, Communication and Storage Systems*

OPERA: A Complete Offline and Anonymous Digital Cash Transaction System with a One-Time Readable Memory

Ki-Woong PARK[†a)] ***and*** **Sung Hoon BAEK**[††b)], ***Members***

SUMMARY Emerging digital payment services, also known as FinTech, have enabled various types of advanced payment transactions (such as Google Wallet, Apple Pay, Samsung Pay, etc.). However, offline peer-to-peer cash transactions still make up about 25.6% of the overall financial transactions in everyday life. By investigating existing online and offline payment systems, we identify three key challenges for building a digital cash transaction system with core features of the offline cash transactions: self-verifiability of digital cash; user anonymity; atomic cash transfer for double spending/depositing protection. In this paper, we propose OPERA, an offline peer-to-peer digital cash transaction system that addresses the three challenges. It newly introduces a concept of 'one-time-readable memory(ORM)' and 'digital token' which is a unit of self-verifiable digital cash. The one-time readability from ORM and three-stage token exchange protocol enable OPERA to provide uniqueness to digital cash and to allow a complete offline digital payment. OPERA devices are enhanced with TCPA technology to ensure the integrity of the physical device package. To evaluate the feasibility and resilience of the OPERA design, we built a prototype on a customized embedded board.

key words: *offline digital payment, digital cash*



(a) Cash-based payment

(b) Conventional online and offline digital payment

Fig. 1 Overview of conventional payment transactions

1. Introduction

With the rapid expansion of mobile platform and the availability of mobile networks, digital payment systems involving credit/debit cards, mobile payments such as Google Wallet, Apple Pay, and Samsung Pay have become increasingly popular forms of payment [1], [2]. They enable financial transactions to be conducted in a simplified and timely manner. Today, about 66.4% of payment transactions around the world are made with this type of payment. Although digital payment system have become popular nowadays, cash-based payment is still one of the common means of payment; it is used in 25.6% of all payments and it has been used widely for a very long time [3]. The main reason behind the successful circulation of cash is that payers and payees can easily and instantly verify its correctness without relying on system infrastructure.

As shown in Fig. 1 (a), anybody can send and receive cash anonymously without the involvement of a third party. The convenience of cash-based payment is one of the reasons digital payment systems cannot totally replace cash-based payment [4]. As shown Fig. 1 (b), digital payment systems can be categorized into online digital payment systems [5]–[12] and offline digital payment systems [13]–[16]; the difference depends on the authorization technique for the financial transactions. Online digital payments use an authorization server for each transfer. They obviously require more communication and depend on the system infrastructure; it leads to user-obstructive latency or requires additional server components for the verification of legitimate transactions. In contrast, offline payment involves no contact with a third party during the payment process; the transaction involves only a payer and a payee. The obvious problem with offline payment is the difficulty of preventing users from double-spending, double-depositing and monetary forgery in an offline manner. Therefore, most offline payment systems use a post-transaction or periodic verification; that is, they rely on the postmortem detection of problematic transactions.

The main objective of this work is to answer the following questions: what are the most challenging issues for realizing a full cash-like digital payment systems; and how can we deliver core features of cash-based payment in a digital way. By investigating conventional payment schemes, from cash-based payments to digital payment systems, we have identified the following three key challenges for delivering the features of cash transactions in a digital way:

- **Self-verifiability:** If digital cash is to be used ubiquitously in an offline manner, there must be a way for it to be validated without online help from financial authorities. Self-verifiability is essential for protecting the in-

Manuscript received December 13, 2016.
Manuscript revised May 6, 2017.
Manuscript publicized July 21, 2017.

[†] The author is with Department of Computer and Information Security, Sejong University, Korea.
[††] The author is with Jungwon University, Korea.
a) E-mail: woongbak@sejong.ac.kr
b) E-mail: shbaek@jwu.ac.kr (Corresponding author)

DOI: 10.1587/transinf.2016INP0008

tegrity of issued digital cash. That means the payer and payee can verify its correctness without having to rely on system infrastructure.

- **User anonymity:** The anonymity of digital cash must be guaranteed because payer and payee wish to keep their payment activities private for cash-like payments.
- **Atomic transfer for double-spending/depositing protection:** The atomic transfer of digital cash must be guaranteed. Otherwise, a disrupted money transfer can lead to undesirable situations for a payer and a payee. It is therefore crucial to have an atomic transfer mechanism so that both the payer and the payee can agree that the transfer is complete. Beside, any number of valid copies can be made from authentic digital cash in a digital system. There must be a copy protection mechanism for a perfect offline peer-to-peer (P2P) digital payment system.

We propose an offline P2P digital cash transaction system, called OPERA that addresses the above three challenges. To make a digital cash self-verifiable and anonymous, we introduce a new concept of memory, termed 'one-time-readable memory (ORM)' and 'digital token'. We define ORM to a special type of non-volatile memory, which has following three features: 1) one of the notable features of ORM is that once data has been written into ORM, it permits data be read only once with accompanying a high-security electrical erasure. 2) data transfer of ORM is highly restricted by allowing the transactions only among authenticated ORMs. 3) ORM guarantees for an atomic data transfer and processing, which is to prevent from a duplication of the data stored in the ORM. The above three features are desirable for realizing a cash-like digital payment system because the uniqueness to each data stored in ORM is guaranteed. In order to realize the memory device on the basis of ORM, we devised digital token which is a unit of self-verifiable digital cash signed by a secure cryptographic key [17] and a three-stage token exchange protocol which is enhanced and tightly coupled with TPM [18]. The TCPA technology inside TPM is a hardware-based security extension. It can provide a way to verify authenticity of devices equipped with ORMs, and can protect the devices from any physical attack designed to break into the devices. To evaluate the feasibility and resilience of our design in real hardware, we built a prototype on a customized embedded board called an OPERA-Digital-Wallet (ODW). The evaluation results show that OPERA can perform resiliently when undesirable events occur during a cash transaction.

The remainder of the paper is organized as follows: In Sect. 2, we present the related works and compare them with our newly developed system. In Sect. 3, we present the overall system design, components, and prototype implementation of OPERA. In Sect. 4, we analyze the security strength of OPERA to confirm the safety of the proposed scheme. In Sect. 5, we evaluate the system's operational resilience and efficiency. Finally, in Sect. 6, we present our conclusions and discuss further works on the feasibility of the new system.

2. Related Works

With the advancement of digitization and the availability of communication networks, a large number of digital payment systems have been proposed and developed; and they all provide a new representation of values. There are two types of digital payment schemes: namely, online and offline digital payment. In this section, we present a brief overview of related works on conventional digital payment systems.

2.1 Online Digital Payment Systems

In online digital payment systems, the payment and deposit phases occur in the same transaction. In other words, every payment is verified by a financial authority at the time of payment, and this process requires the financial authority (such as a bank or a credit card company) to be online for every payment performed between the payer and the payee. A wide spectrum of digital payment services has been realized with online digital payment systems. A set of protocols, namely iKP [5], is used for digital payments over open networks. It provides anonymous authentication, payment integrity, and nonrepudiation mechanisms between the payer and the payee. This technology is standardized into SET [6] and is deployed in the PayPal commercial service [7].

Many studies on micropayment schemes have tried to mitigate the operational overhead by focusing on various kinds of computing devices and network environments; examples include E-Cash [8], NetCash [9], Café [10], and MiniPay [11]. These schemes can be broadly deployed in digital payment systems to allow users to perform repeated payments anonymously and efficiently. Many of these schemes use one-way hash functions to generate chains of hash values; that means users can perform billing transactions by releasing a certain number of hashes in a hash chain.

The above-mentioned online payment schemes are widely used. The growing success of auctions, online games, and educational Web sites on the Internet has led to the emergence of payment service providers, which allow person-to-person e-payments over the Internet [12]. Although they promise mature, they are more problematic than cash-based payment. Online systems always require a financial authority to be involved in transactions, but that step is not always possible in many regions of the world. Additionally, online payments are always unidirectional, which means the user is always the payer.

2.2 Offline Digital Payment Systems

In offline digital payment schemes, the digital payment is verified after the transaction at some convenient time for both the payer and the payee, obviating the need for a financial authority to be involved in every payment transaction. There have been many studies on offline digital payment schemes: for example NFC-Voucher [13], Mondex [14], EMV [15], and BitCoin [16]. They enable offline transferability: the payee can use the amount received to make a new payment, without having to go to the financial authority. These schemes offer payers an electronic wallet, which can be used to prevent fake-terminal attacks on the payer's PIN. These offline payments are managed exclusively by the payer and payee; the payment process involves no contact with a third party. The wallets are used for small payments and can support offline transactions.

While not requiring the involvement of a financial authority, existing offline payment systems have significant shortcomings in terms of their support for P2P transactions. First, offline digital payment are not fully offline. In spite of their general independence of online systems, most offline payment systems [13]–[16] use periodic online checking to update a black list or to perform online validation. Second, the self-verifiability function is limited without some online help from financial authorities. This limitation occurs because digital information can often be copied perfectly and arbitrarily. Thus, digital payment systems must enable an honest payer to convince the payee to accept a legitimate payment and, at the same time, prevent a dishonest payer from making unauthorized payments. Third, they do not support the transfer of money between cardholders [14], [15]. Fourth, they have only a limited ability to work autonomously without any additional hardware, such as a card reader or writer. The necessity of card readers is the main limitation. In short, the loading of money onto a card requires direct interaction with additional hardware.

3. OPERA System Architecture

To support digitized P2P cash transactions, we propose an offline P2P digital cash transaction system, namely OPERA. We first present our key features on the three challenges. We then explain the OPERA hardware architecture and a novel token exchange protocol for secure exchange of digital cash.

3.1 Core Features of OPERA

We devised the following three solutions that overcome the challenges of the digitized P2P cash transactions:

- **Self-verifiable token:** To make digital cash self-verifiable, we introduce a concept of token, a unit of digital money signed with a secure cryptographic key. The token can be encrypted by the private key of a financial authority with the power to issue notes (such as the US Federal Reserve Bank). Such a token is valid only when the digital signature of the token matches the context of the token (that is, the serial number, value, and issued date). OPERA supports tokens with different monetary values (such as a $10 token or a $100 token) so that the number of token exchanges required in a P2P transactions can be kept to a minimum.
- **Anonymous authentication:** For the anonymity of digital cash, payer's ODW and payee's ODW perform a TPM-based anonymous authentication. A key feature of the authentication scheme is that each entity see if the counterpart is an authentic ODW via a process, remote attestation. During the attestation process, the identifications of the payer and the payee are concealed by integrating direct anonymous attestation [19] into our system. Consequently, the token can be sent or received with anonymity after the authentication.
- **Three-stage token exchange protocol:** For the atomic transfer and copy protection of digital cash, we developed a three-stage token exchange protocol: it first anonymously authenticates the validity of the counterpart in a P2P transaction (*Stage1*); it transfers tokens re-encrypted with a one-time secure key for the transaction (*Stage2*); and, finally, it hands over the secure key to the counterpart only when the whole transaction is completed without fault (*Stage3*). The transaction is atomic because the transferred tokens are useful only when decrypted with the secure key. The three-stage token exchange protocol leverages TCPA technology to protect tokens from being copied. The movement of tokens is limited exclusively to authentic ODWs. The TCPA technology is used to protect an authentic ODW from any physical attack that attempts to break into the devices; such attacks may involve physical tampering. The tokens are initially charged to an ODW by a financial authority.

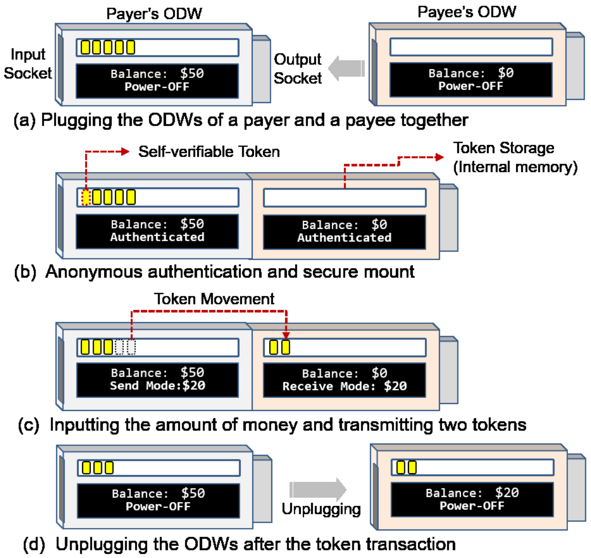
In the rest of the section, we explain how these core features come together to realize a secure form of offline P2P digital cash transaction.

3.2 ODW Internal on the Basis of OPERA

An ODW acts as a digital wallet. The goal of an ODW is to enable ODW holders to use an OPERA-based digital cash transactions anywhere in an offline and person-to-person manner. Figure 2 shows a digital cash transaction between a payer's ODW and a payee's ODW. If a payer transmits $20 to a payee and one token is mapped to $10, an OPERA-based digital cash transaction is performed as follows:

1. The ODWs of a payer and a payee are plugged together (Fig. 2 (a)).
2. The physical mount between the ODWs is authenticated by the anonymous authentication (Fig. 2 (b)).
3. The amount of money(tokens) to be transmitted is inputted by the payer (Fig. 2 (c)).
4. Two tokens from the payer's ODW are transmitted to the payee's ODW in a token-by-token manner (Fig. 2 (c)).
5. After the verification for the performed transaction, the ODWs are unmounted (Fig. 2 (d)).

Figure 3 shows the internal architecture of an ODW. It consists of a one-time-readable memory (ORM) as a token storage for storing the user's tokens, and a transaction



(a) Plugging the ODWs of a payer and a payee together

(b) Anonymous authentication and secure mount

(c) Inputting the amount of money and transmitting two tokens

(d) Unplugging the ODWs after the token transaction

Fig. 2 Digital payment transaction with OPERA

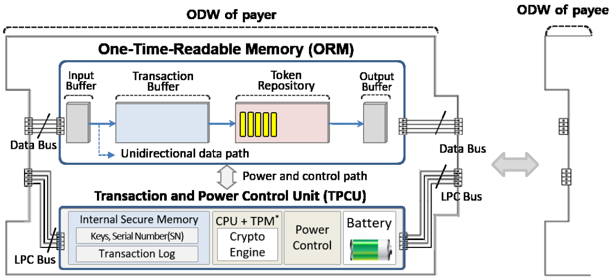


Fig. 3 Overall architecture of OPERA Digital Wallet (ODW)

Table 1 Notations of the entities and messages

Definition of the entity symbols
· Payer: A
· Payee: B
Definition of the Message symbols
· SN_α: Serial number of α
· PK_α: Public key of α
· SK_α: Private key of α
· $SEED$: Seed value for generating One-Time Keys
· OTK_n: Generated n^{th} One-Time-Key
· PTK_α: Key for encrypting/decrypting tokens
· { Message } SK_α: Digital signature by SK_α + Message
· { Message } PK_α: Message encrypted by PK_α
· { Message } OTK_n: Message encrypted by OTK_n
· N_α: Nonce value by α
· $N'_\alpha = N_\alpha$ XOR $0{\times}FFFFFF$
· C_α: The number of token transmission
· AMT: Amount of money to be transmitted
· ACK: Acknowledgement Message
· FIN: Payment Complete
· $Token$: {SN‖value‖Issue Date}SK_c
· TR_k: k^{th} transaction message
· FLG: Acknowledge reception flag (received:1, not:0)

and power control unit (TPCU) that takes care of the authentication and control of the user's tokens, and a data bus for secure communication with other ODWs. The ODW is protected and encapsulated by TPM [18] technology so that the internal data pertaining to the keys, log, and tokens have greater security against external software attacks and physical breakage. The details of the major components of an ODW are as follows:

- **ORM as a token storage:** This part consists of token input and output buffers (volatile memory), a transaction buffer (nonvolatile memory), and a token repository (nonvolatile memory). The subcomponents of ORM are linked each other with unidirectional links, which means every tokens can be moved in a unidirectional path. All entry and exit of ORM are being controlled by TPCU so that any transmission of the tokens are enabled only after completion of the anonymous authentication.
- **TPCU:** The TPCU is responsible for the secure P2P cash transaction among ODWs, and power control. The TPCU contains an internal secure memory that stores private and public keys and serial numbers; it also contains a transaction log that stores the history of payment transactions to guarantee the atomic properties in the event of crashes or hardware failures. It serves to ensure that an incomplete transaction can be completed seamlessly after a crash or hardware failure. Finally, the power control module is designed to supply power to the token storage only when the ODWs of the payer and payee are mounted together after the anonymous authentication.
- **Low pin count and data bus:** An ODW has two bus interfaces for anonymous authentication and payment transactions between a payer and payee. The low pin count (LPC) [20] bus is used for anonymous authentication and transaction logging. The data bus is used for the token transfer between a payer and payee. Note that this design assumes that two ODW devices are physically mounted. As an alternative, the ODWs can communicate through other means, such as a wireless near-field communication network [21]. Consequently, the benefits of OPERA can be integrated into mobile computing platforms as a digital wallet module.

3.3 Digital Cash Transaction Protocol

ODWs use a three-stage digital cash exchange protocol to transfer tokens securely by means of the hardware components explained in the previous section. The protocol consists of three stages. In the beginning, the payer and payee perform an anonymous authentication and mount operation by entering *Stage1*. After that, a certain number of tokens on the payer's ODW are moved to the payee's ODW in *Stage2*. The payer and payee then perform verification and unmounting operations in *Stage3*. This section describes the three stages of the proposed payment protocols. All the notations are summarized in Table 1.

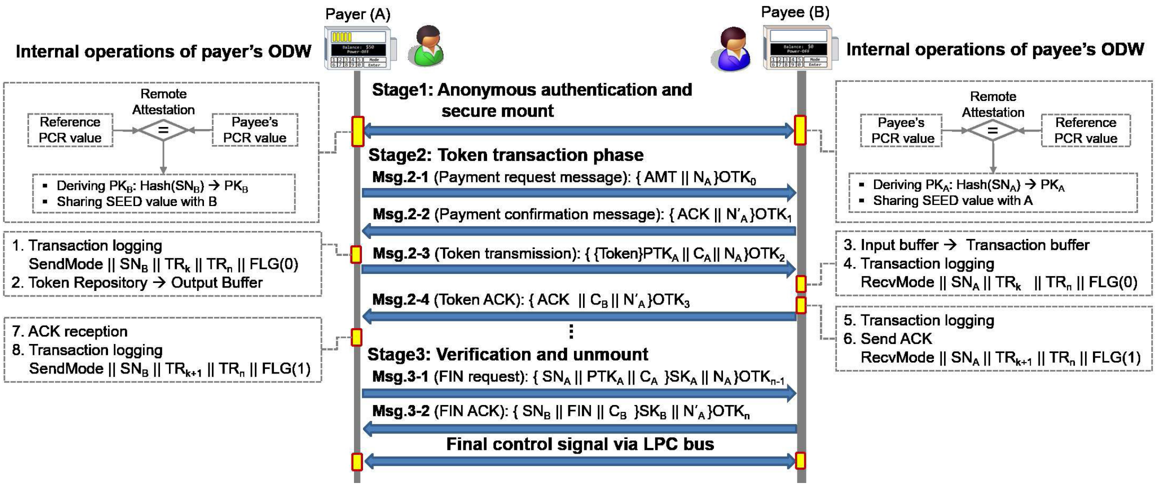

Payer (A)
Payee (B)
Internal operations of payer's ODW
Internal operations of payee's ODW
Remote Attestation
Reference PCR value
=
Payee's PCR value
Deriving PK_B: Hash(SN_B) → PK_B
Sharing SEED value with B
Payee's PCR value
Reference PCR value
Deriving PK_A: Hash(SN_A) → PK_A
Sharing SEED value with A
Stage1: Anonymous authentication and secure mount
Stage2: Token transaction phase
Msg.2-1 (Payment request message): { AMT || N_A }OTK_0
Msg.2-2 (Payment confirmation message): { ACK || N'_A }OTK_1
Msg.2-3 (Token transmission): { {Token}PTK_A || C_A || N_A }OTK_2
Msg.2-4 (Token ACK): { ACK || C_B || N'_A }OTK_3
⋮
Stage3: Verification and unmount
Msg.3-1 (FIN request): { SN_A || PTK_A || C_A }SK_A || N_A }OTK_{n-1}
Msg.3-2 (FIN ACK): { SN_B || FIN || C_B }SK_B || N'_A }OTK_n
Final control signal via LPC bus
1. Transaction logging
SendMode || SN_B || TR_k || TR_n || FLG(0)
2. Token Repository → Output Buffer
7. ACK reception
8. Transaction logging
SendMode || SN_B || TR_{k+1} || TR_n || FLG(1)
3. Input buffer → Transaction buffer
4. Transaction logging
RecvMode || SN_A || TR_k || TR_n || FLG(0)
5. Transaction logging
6. Send ACK
RecvMode || SN_A || TR_{k+1} || TR_n || FLG(1)

Fig. 4 Three-stage digital cash exchange protocol of OPERA: Stage1 (anonymous authentication and secure mount protocol); Stage2 (token transaction phase); and Stage3 (verification and unmount)

3.3.1 Stage1: Anonymous Authentication and Secure Mount Protocol

The first step involves anonymous authentication between the payer's ODW and the payee's ODW to see if each device is an authentic ODW. For anonymous authentication, we integrated an anonymous attestation scheme, called direct anonymous attestation [22] which has been adopted in the latest version of TPM specification [18] to address privacy concerns. This scheme is for remote attestation of a computing device, while not revealing the privacy of the user of the device.

As described in Fig. 4, the authenticity of partner's ODW can be verified by comparing the value of platform configuration register (PCR) of the partner's ODW with the predefined PCR value stored inside a secure memory region of ODW. The PCR is a built-in register that can be used to store the 160-bit hash values obtained from the SHA-1 [23] hashing algorithm. The PCR values can only be changed by the *Extend()* function, which is an internal function of the TPM. Whenever the ODW is powered on, the *Extend()* is executed, and outputs the hash result of hardware configuration and firmware inside ODW. To enable an ODW to verify the authenticity of partner's ODW, *Quote()* function is used, which uses a TPM private key called an attestation identity (*AIK*) to return a digital signature of the current PCR values. The *AIK* is created inside the TPM and protected by the TPM so that *Quote()* provides proof that the output of *Quote()* was generated inside ODW in a secure manner. Consequently, user can check the correctness of ODW being used and refuse to use it if the PCR value is not matched to the predefined PCR value. The predefined PCR value is opened to the users as a reference PCR value of ODW by issuing authority. If the partner's ODW is correct, the PCR value must be exactly same with the predefined PCR value.

After the authenticity verification of partner's ODW, the payer's ODW and the payee's ODW can construct secure channel between payer's TPM and payee's TPM which allows anonymous interactions. The anonymity can be guaranteed because all the message between the payer and payee is concealed by on-chip encryption inside an ODW package protected with TCPA technology. It guarantees that the keys and all the messages of the ODW cannot be accessed by any malicious attacks or physical breakage. Therefore, any attempts to gain authentication secrets or serial number (SN) are destined to fail. Through the anonymous attestation scheme, the payer and payee not only can check the authenticity of partner's ODW but also can share *SEED* and *SN* of each partner.

The *SEED* is used as a seed value to generate one-time keys (OTKs) for secure communication between the payer and payee. The public key of a partner can be derived locally by using the *SN* of the partner and the hash function; this process means that the internal secure memory acts as a certificate authority and key storage area. For instance, if the size of *SN* is 16 bits and the size of the hash output is N bits, all possible 2^{16} *SN*s can be mapped to one public key (PK) from among 2^N PKs. The upcoming transaction(*Stage2,3*) can be performed only upon the successful completion of *Stage1*; thus, the exchange of tokens can be performed only among authentic ODWs (secure mount). Finally, the *SEED* is used to generate an OTK , which is used to encrypt messages of *Stage2* and *Stage3*. The OTK is derived as follows:

$$OTK_n = \begin{cases} Hash(OTK_{n-1} || Msg^{\dagger}), & \text{if } n \geq 1 \\ Hash(SEED), & \text{if } n = 0. \end{cases} \quad (1)$$

$Msg^{\dagger}$: Context of previous message

The OTK makes the upcoming message transmissions stronger in terms of the message integrity and atomicity because all the messages are chained to each other by the set of *OTK*s and the context of the previous message. In our implementation, we used SHA-1 [23] as a secure hash function. The more secure hash functions can be deployed into OPERA in order to provide collision resistance.

3.3.2 Stage2: Token Transaction Phase

Stage2 is the token transmission phase. The payer generates a payment request message by sending the amount of digital cash (tokens) to be transmitted (*Msg.2-1*). Upon receiving the message, the payee transmits an acknowledgement (*Msg.2-2*). The confirmed amount of tokens is then transmitted through *Msg.2-3* and *Msg.2-4*. The specific message protocol is as follows:

- *Msg.2-1* $A \Rightarrow B$: The payer generates a payment request message by sending the number of tokens to be transmitted (AMT) and N_A. They are encrypted with OTK_0, which is derived from Eq. (1).
- *Msg.2-2* $B \Rightarrow A$: When the message arrives, the payee verifies *Msg.2-1* and transmits *Msg.2-2* as a confirmation. It contains $N_A^{'}$, which is encrypted with OTK_1.
- *Msg.2-3* $A \Rightarrow B$: The payer transmits $\{Token\}PTK_A$, C_A, and N_A, all of which are encrypted with OTK_2. A token consists of SN, value, the date of issue, and the digital signature of issuing authority, using SK_c. The payee can use the public key of the issuer (PK_c) to verify that the tokens have been received. The PTK prevents the received tokens from being decrypted before the completion of the payment transaction. The incremental counter (C) stores the number of times a token transmission has occurred, and the value of the counter is stored on a secure memory region of TPCU. The counter is used to verify the completeness of the payment transaction in a comparison of the payer's counter (C_A) and the payee's counter (C_B).
- *Msg.2-4* $B \Rightarrow A$: When the *Msg.2-3* is received, the payee sends an acknowledgement message, *Msg.2-4*, which contains C_B and $N_A^{'}$. Note that the payee still cannot acquire the original token because the payee does not have PTK_A. Therefore, the received encrypted tokens are temporarily stored in the transaction buffer.

In short, the payer and payee can confirm the amount of money to be transmitted through *Msg.2-1* and *Msg.2-2*. The number of tokens to be transmitted is determined by the formula $AMT/token$ value. For example, if $800 is to be transmitted and the value of a token is $10, the payer transmits 80 tokens.

3.3.3 Stage3: Verification and Unmount

Stage3 is for the verification and finalization of payment transactions. It is launched by sending PTK_A, a key used to encrypt the tokens from the payer to the payee. The payee can then decrypt the tokens and stores them a token queue. The verification can be achieved by checking the correctness of the received tokens and by comparing C_A with C_B. If the payment transaction is performed correctly, the two values should be identical to each other. More specifically, *Stage3* performs the following operations:

- *Msg.3-1* $A \Rightarrow B$: After the transmission of the tokens through *Stage2*, the payer sends PTK_A, C_A, SN_A, and a digital signature after the encryption by OTK_{n-1}.
- *Msg.3-2* $B \Rightarrow A$: When the message is received, the payee can verify the received tokens by checking the validity of C_B and SN.

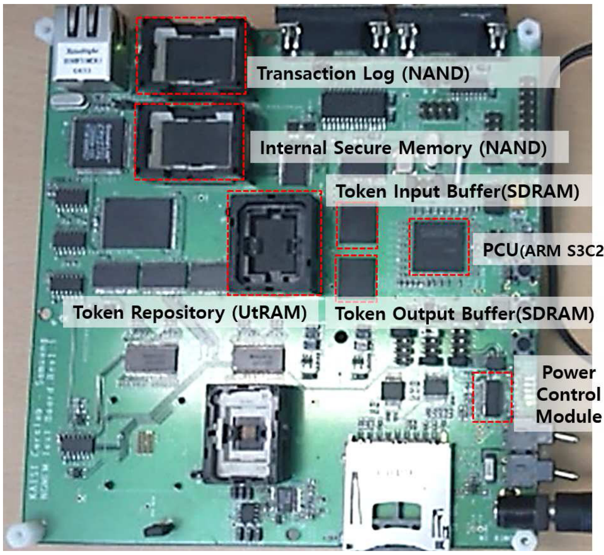


Fig. 5 OPERA-based offline digital payment hardware platform

The payment transaction is finalized by the transmission of a FIN message. Upon receiving the FIN message, the TPCU of the payer generates a final control signal to payer and payee via the LPC bus. The signal is to cut off the power supply to the ORM and to clear the transaction log of ODWs.

3.4 Prototyping OPERA

To evaluate the feasibility and resilience of our design in a real hardware, we built an OPERA prototype on a customized embedded board. As shown in Fig. 5, we equipped the embedded board with volatile and nonvolatile memory to evaluate the diverse kinds of memory chips. In this work, we mapped the nonvolatile memory (NAND flash memory) to the transaction log and the internal secure memory of the PCU; we also mapped the volatile memory (SDRAM) to the token buffer of the token storage. For the transaction buffer and token repository, we used UtRam [24]. We used an ARM-based CPU and a TPM chip as the TPCU that performs the authentication and cryptographic operations and ORM control. In order to implement the authenticated power control, we deployed a power control module to selectively supply power to the ODWs. Only when the ODWs of the payer and payee are mounted together after the anonymous authentication, the power control module is designed to supply power to the ORM. Finally, when the token exchange is finished (*Stage3*), all the data in the token buffer are cleared completely by cutting off the power supply to the token storage.

4. Case Study on Safety of OPERA

In this section, we present our analysis is based on consideration of resilience, security functionality, and safety against potential attacks to confirm the safety of OPERA.

4.1 Resiliency of OPERA

The fact that a connection can be broken as a result of hardware failure before the completion of a transaction is a problematic. OPERA uses the transaction log to handle this type of situations. The log is used to ensure that an incomplete transaction can be resumed and completed after a crash or hardware failure. Through the payment protocol, the overall transactions are logged in transaction logs in case that a fault (such as a battery outage or disruptive unmounting) disrupts the payment transaction before the transaction is completed. The logging data enables a disrupted transaction to be resumed when the payer's ODW and the payee's ODW are mounted again. As shown in Fig. 4, the transaction log contains data that indicates the send or receive mode, the serial number of the partner ($SN_{A/B}$), the current transaction number (TR_k), and the total number of transactions (TR_n); it also contains a completion flag that checks whether the corresponding ACK message has been delivered. If the completion flag is zero and TR_k is k in the payer's ODW, the ACK messages of the kth token transmission are not delivered. If the completion flag is zero and TR_k is k in the payee's ODW, the ACK of the kth token transmission is not transmitted to the payer's ODW. Thus, the payer and payee can successfully complete a disrupted transaction in a sustainable manner. A more comprehensive evaluation of OPERA in terms of resilience is given in Sect. 5.1.

4.2 Safety against Potential Attacks

Our analysis is based on consideration of external software attacks, replay attacks, MITM attacks. We assume that any principle can place or inject a message on any link at any time. In addition, any principle can see, delete, alter, and redirect all exchanged messages that are passed along any link or replay messages recorded in past transactions.

- **Against copied token attacks:** A malicious user may try to falsify the token by double-reading the internal input buffer of ODW. However, the falsification of the malicious user cannot succeed because the TPCU performs the token transfer atomically which means that the payee can acquire the usable tokens from the input buffer only after *Stage3*. Even if the token transition is blocked at a middle point, the payee can acquire the usable tokens from the input buffer only after completion of the blocked transaction by mounting the payer's ODW and the payee's ODW again. In addition, the internal memory of an ODW is protected by TCPA technology so that stored keys and tokens cannot be accessed from an outside package.
- **Against falsified ODW attacks:** An attacker can develop a falsified ODW and try to connect it to an authentic ODW to break the OPERA protocol. However, the attack cannot succeed because the ODWs of the payer and payee can only be connected to each other after the mutual authentication and secure mount.
- **Against MITM and replay attacks:** In the three-stage token exchange protocol, the nonce ($N_{A/B}$) is used to prevent attacks such as replay attacks and man-in-the-middle (MITM) attacks. An MITM attack is an active eavesdropping attack where a falsified ODW attack attempts to intercept, read or alter information transmitted between two ODWs. Thus, N_A is included in all interactions in the three-stage token exchange protocol to verify the freshness of the data and the value is changed per every message transaction, which is encrypted by one-way generated one-time-key (OTK_n).

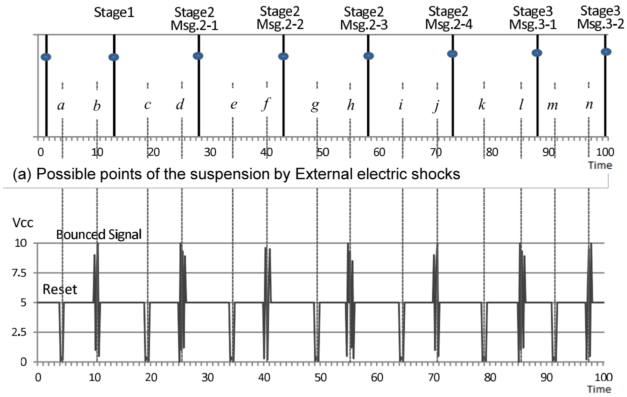
5. Evaluation

In this section, we evaluate the operational resilience and efficiency of the OPERA digital cash transaction protocol.

5.1 Operational Resilience of OPERA

The operational resilience should be guaranteed so that tokens can be transferred safely between the payer and payee. As described in Sect. 3, we subject the implemented ODW devices to electric shocks in order to evaluate the capability of resuming a payment transaction suspended due to external factors. The process of transferring tokens consists of multiple message communications and memory read/write operations. To verify the operational resilience, we subjected OPERA to multiple electric shocks. The evaluation in this section deals with the operational resilience of all message transmissions.

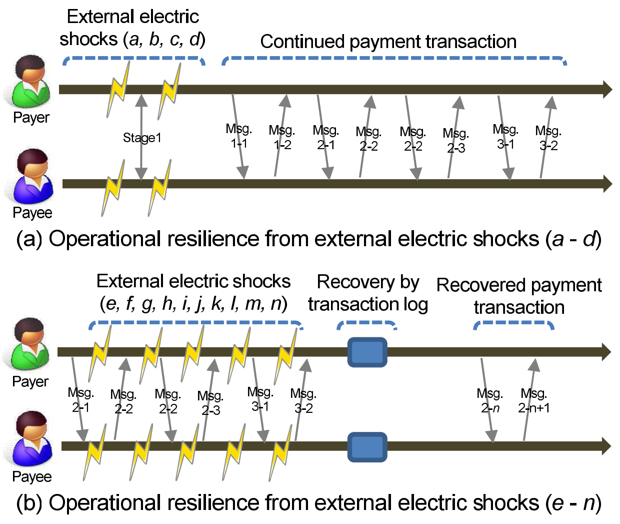
Figure 6 (a) shows possible points of suspension. Figure 6 (b) illustrates the external electric shocks inflicted at possible suspension points of the message protocol. The events a, c, e, g, i, k, and m appear when the reset signal of



(a) Possible points of the suspension by External electric shocks

(b) External electric shocks (reset, bounced signal) inflicted to the atomic transactions

Fig. 6 Operational resilience evaluation of ODW


External electric shocks (a, b, c, d)
Continued payment transaction
Payer
Stage1
Msg. 1-1
Msg. 1-2
Msg. 2-1
Msg. 2-2
Msg. 2-2
Msg. 2-3
Msg. 3-1
Msg. 3-2
Payee
External electric shocks (e, f, g, h, i, j, k, l, m, n)
Recovery by transaction log
Recovered payment transaction
Payer
Msg. 2-1
Msg. 2-2
Msg. 2-2
Msg. 2-3
Msg. 3-1
Msg. 3-2
Msg. 2-n
Msg. 2-n+1
Payee

(a) Operational resilience from external electric shocks (*a* - *d*)

(b) Operational resilience from external electric shocks (*e* - *n*)

Fig. 7 Operational resilience of OPERA from injecting electric shocks

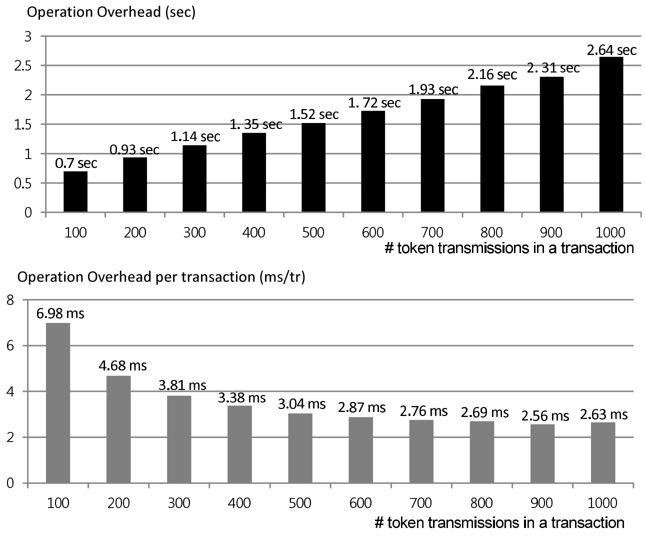


Fig. 8 Computation overhead with varying the number of transactions

the ODW is set. These events correspond to real events such as a battery burnout or the user's wrong power down. The events *b*, *d*, *f*, *h*, *j*, *l*, and *n* are generated when an external electric shock is applied to the ODW. These events describe a suspension that occurs during the protocol processing due to poor contact between ODW devices or in the event of an unexpected electric shock from an external source. The signal shown in Fig. 6 (b) was applied with a voltage regulator during the protocol processing. We examine whether the operational status of the protocol is maintained or the previous protocol status shows up correctly after the link between the ODW devices has been reconnected. As shown in Fig. 7 (a), we have confirmed that when events *a*, *b*, *c*, and *d* occur the authentication operation (*Stage1*) restarts from the beginning after the link between the devices has been reconnected. As shown in Fig. 7 (b), the other events (*e*, *f*, *g*, *h*, *i*, *j*, *k*, *l*, *m*, and *n*) are located in the payment transactions after the authentication operation. Our results confirm that for all these events the operation works correctly and is completed successfully because the protocol uses the transaction log inside the ODW for a successful resumption of the transaction.

5.2 Operational Efficiency of OPERA

Since the OPERA system is expected to be implemented on a single silicon die, we used a board-based prototype to assess the system's overall performance trend. Figure 8 (a) shows the total time taken for a digital cash transfer, and Fig. 8 (b) illustrates the time taken for each token transmission. In these experiments, the total time taken is measured by varying the number of token transmissions from 100 to 1000 (x-axis); and the time taken for each token transmission (y-axis) is calculated by dividing the total time by the number of token transmissions. We used tokens with the least monetary value to test the prototype under the stress of a high number of token transmissions. Figure 8 (a) clearly shows that the total time taken to complete transactions increases as the number of token transmissions increases. The transfer of each token transmission in *Stage2* requires a symmetric key computation and a hash computation. The operation overhead in *Stage2* increases proportionally as the number of token transmissions increases. This performance trend would be a drawback in comparison to conventional offline digital payment systems which has a static operation overhead. However, the association of a financial authority existing offline payment systems have some drawbacks under the P2P transactions. Usually, offline digital payment systems do not satisfy the characteristic of full offline. On the other hand, Fig. 8 (b) shows that the time taken per token transmission is reduced when the number of transactions increases. This trend occurs because more token transmissions amortize the overhead of the authentication in *Stage1* and the verification in *Stage3*

6. Conclusion

In this paper, we propose OPERA, a complete offline digital cash transaction system for P2P transactions. We thoroughly investigate cash-based payment systems and digital payment systems and identify the main challenges of making cash-based payments in a digital way. From our design of the OPERA hardware architecture and a three-stage token exchange protocol, we built a prototype on a custom board. The benefits of OPERA are not limited to bidirectional digital cash transactions. The benefits of OPERA can be integrated into mobile computing platforms as a digital wallet module. They also can be extended to diverse secure transaction systems, such as a digital voucher, one-time authentication, digital right management, video on demand services, and e-book content systems.

Acknowledgements

This work was supported by the National Research

Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2017R1C1B2003957, NRF-2016R1A4A1011761).

References

[1] E. Taylor, "Mobile payment technologies in retail: a review of potential benefits and risks," International Journal of Retail & Distribution Management, vol.44, no.2, pp.159–177, 2016.

[2] R. Ali, J. Barrdear, R. Clews, and J. Southgate, "Innovations in payment technologies and the emergence of digital currencies," Bank of England Quarterly Bulletin, p.Q3, 2014.

[3] S.S. Claire Greene and J. Stavins, "The survey of consumer payment choice," Aug. 2016.

[4] E. Gannamaneni, J. Ondrus, and K. Lyytinen, "A post-failure analysis of mobile payment platforms," 48th International Conference on System Sciences (HICSS), pp.1159–1168, IEEE, 2015.

[5] M. Bellare, J.A. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, E. Van Herreweghen, and M. Waidner, "Design, implementation, and deployment of the ikp secure electronic payment system," IEEE J. Sel. Areas Commun., vol.18, no.4, pp.611–627, April 2000.

[6] S. Kungpisdan, B. Srinivasan, and P.D. Le, "A secure account-based mobile payment protocol," ITCC, vol.1, pp.35–39, 2004.

[7] P. Co.Ltd, "On-line paypal payment service," 2016.

[8] M. Scheir, J. Balasch, A. Rial, B. Preneel, and I. Verbauwhede, "Anonymous split e-cash-toward mobile anonymous payments," ACM Transactions on Embedded Computing Systems (TECS), vol.14, no.4, p.85, 2015.

[9] G. Medvinsky and C. Neuman, "Netcash: a design for practical electronic currency on the internet," Proceedings of the 1st ACM conference on Computer and communications security, CCS '93, New York, NY, USA, pp.102–106, ACM, 1993.

[10] T.P. Pedersen, "Electronic payments of small amounts," Proceedings of the International Workshop on Security Protocols, London, UK, pp.59–68, Springer-Verlag, 1997.

[11] A. Herzberg and H. Yochai, "Minipay: charging per click on the web," Selected papers from the sixth international conference on World Wide Web, Essex, UK, pp.939–951, Elsevier Science Publishers Ltd., 1997.

[12] M.E. Steurer and F. Kappe, "A micropayment enabled webshop for digital assets in virtual worlds," Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek '10, 2010.

[13] G. Van Damme, K.M. Wouters, H. Karahan, and B. Preneel, "Offline nfc payments with electronic vouchers," 1st ACM workshop on Networking, systems, and applications for mobile handhelds (MobiHeld), 2009.

[14] Mondex, "Mondex digital payment service," 2016.

[15] S.C.A. Press, "Smart card applications: Emv," 2016.

[16] K. Crary and M.J. Sullivan, "Peer-to-peer affine commitment using bitcoin," ACM SIGPLAN Notices, vol.50, no.6, pp.479–488, 2015.

[17] R. Kikuchi, L.T. Phong, and W. Ogata, "A Framework for Constructing Convertible Undeniable Signatures," vol.6402, pp.70–86, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[18] M.A. Bouazzouni, E. Conchon, and F. Peyrard, "Trusted mobile computing: An overview of existing solutions," Future Generation Computer Systems, 2016.

[19] E. Brickell, L. Chen, and J. Li, "A new direct anonymous attestation scheme from bilinear maps," International Conference on Trusted Computing, pp.166–178, Springer, 2008.

[20] I.P. Release, "Low pin count interface specification document," tech. rep., INTEL Development Center, 2010.

[21] N. Forum, "Introduction to nfc," 2010.

[22] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," Proceedings of the 11th ACM conference on Computer and communications security, CCS '04, New York, NY, USA, pp.132–145, ACM, 2004.

[23] D. Eastlake and P. Jones, "Rfc 3174: Us secure hash algorithm 1 (sha1), 2001," URL http://www.ietf.org/rfc/rfc3174.txt, 2012.

[24] S.M. Division, "Datasheet: K1s161611a (uni-transistor random access memory)," datasheet, Samsung Electronics, 2016.

Ki-Woong Park received the B.S. degree in computer science from Yonsei University, Korea, in 2005, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 2007, the Ph.D. degree in electrical engineering from KAIST in 2012. He received a 2009-2010 Microsoft Graduate Research Fellowship. He worked for National Security Research Institute as a senior researcher. He has been a professor in the department of computer and information security at Sejong University. His research interests include security issues for cloud and mobile computing systems as well as the actual system implementation and subsequent evaluation in a real computing system.

Sung Hoon Baek received the B.S. degree in electronics engineering from Kyungpook National University, Korea, in 1997, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 1999, the Ph.D. degree in electrical engineering from KAIST in 2008. He worked for Electronics Telecommunication Research Institute (ETRI) as an R&D staff from 1999 to 2005 and for Samsung Electronics as a senior R&D staff from 2008 to 2011. He has been an assistant professor in the department of computer system engineering at Jungwon University since 2011. His research interests include storage system, operating system, and parallel processing.