# FORCE
## *Fully Off-line secuRe CrEdits for Mobile Micro Payments*

Vanesa Daza$^1$, Roberto Di Pietro$^2$, Flavio Lombardi$^2$ and Matteo Signorini$^1$

$^1$*Department of Information and Communication Technologies - Universitat Pompeu Fabra, Barcelona, Spain*

$^2$*Department of Mathematics and Physics - Roma Tre University, Roma, Italy*

Keywords: Security, Payment, Protocol, Off-line.

Abstract: Payment schemes based on mobile devices are expected to supersede traditional electronic payment approaches in the next few years. However, current solutions are limited in that protocols require at least one of the two parties to be on-line, i.e. connected either to a trusted third party or to a shared database. Indeed, in cases where customer and vendor are persistently or intermittently disconnected from the network, any on-line payment is not possible. This paper introduces FORCE, a novel mobile micro payment approach where all involved parties can be fully off-line. Our solution improves over state-of-the-art approaches in terms of payment flexibility and security. In fact, FORCE relies solely on local data to perform the requested operations. Present paper describes FORCE architecture, components and protocols. Further, a thorough analysis of its functional and security properties is provided showing its effectiveness and viability.

## 1 INTRODUCTION

Market analysts have predicted that mobile payments will overtake the traditional marketplace, thus providing greater convenience to consumers and a new source of revenue to many companies. This scenario can produce a shift in purchase methods from classic credit cards to new approaches such as mobile-based payments, giving new market entrants a further business chance (Lewandowska, 2013).

Widely supported by recent hardware, mobile payment technology is still at its early stages of evolution but it is expected to rise in the near future as demonstrated by the growing interest in crypto-currencies. The first micro payment scheme, named *Payword*, was proposed by Rivest and Shamir (Rivest, 1996) and was based on hash operations. Nowadays, crypto-currencies and decentralized payment systems (e.g. Bitcoin (Martins and Yang, 2011)) are increasingly popular, fostering a shift from physical to digital currencies. However, such payment techniques have yet to become commonplace, due to several unresolved issues, including a lack of widely-accepted standards, limited interoperability among systems and, most importantly, security. A common limitation of present approaches is that the payment protocol either requires at least one of the involved devices to be on-line (i.e. connected to an external trusted third party), or it requires each transaction to be linked to a bank account.

### 1.1 Problem and Objectives

Present digital payment solutions rely on the capability of involved devices to go on-line, i.e. to connect to a remote payment service/gateway. Although many of them claim to provide off-line transactions, they are limited to customer authentication whilst blindly relying on trusting the bank for transactions (as for credit cards). As a matter of fact, for all those cards that do not rely on any bank account, such as prepaid cards, a network connection to the Internet is required in order to check card validity and balance. Unfortunately, a network connection can be unavailable due to either temporary network service disruption or due to a permanent lack of network coverage. Last, but not least, such on-line solutions are not very efficient since remote communication can introduce delays in the payment process. As a consequence, some merchants would rather prefer off-line solutions to take advantage of the low latency of the payment process and of the data plan cost reduction. The lack of fully off-line secure prepaid solutions not linked to bank accounts is mainly due to the difficulty of checking the trustworthiness of a transaction without a trusted third party. This represents an important deficiency

Daza V., Di Pietro R., Lombardi F. and Signorini M..
FORCE - Fully Off-line secuRe CrEdits for Mobile Micro Payments.
DOI: 10.5220/0005053201250136
In *Proceedings of the 11th International Conference on Security and Cryptography* (SECRYPT-2014), pages 125-136
ISBN: 978-989-758-045-1

in nowadays mobile payment ecosystems, in particular for those emerging countries where mobile micro payment solutions are flourishing (thanks to the recent widespread success of low-cost smartphones) despite difficulties in bank pervasiveness.

## 1.2 Contribution

This paper introduces and discusses a novel off-line micro payment solution. The proposed solution, named FORCE, does not require any kind of network connectivity or bank account. In FORCE both the vendor and the customer device can be disconnected from the Internet and from any other trusted third party thus relying only on local data. To the best of our knowledge, this is the first approach that is able to provide secure fully off-line payments while being resilient to different malicious adversaries as described in Section 7.

# 2 RELATED WORK

Solutions proposed so far for mobile payments can be classified according to the following taxonomy:

- **Fully On-line:** solutions such as (Chen et al., 2010; Golovashych, 2005; Vasco et al., 2010) that require the customer's mobile device to be connected to a network (e.g. 3G) in order to communicate with a bank, a payment-gateway or a trusted third party;
- **Semi Off-line:** solutions such as (Kadambi et al., 2009; Sekhar and Mrudula, 2012) that require an active connection only on the vendor side;
- **Weak Off-line:** solutions such as (Dominikus and Aigner, 2007; Nishide and Sakurai, 2011) that require a connection either to a shared dataset or to a peer-to-peer network. Such approach, by allowing access to past transactions, enables vendors to check for customer's account validity, thus preventing malicious behavior such as double spending. Other solutions belonging to the weak-off-line category work with digital cash designed to be accepted either by specific vendors (known as digital vouchers) or within a specific short time window like in (Patil and Shyamasundar, 2004; Aigner et al., 2007);
- **Fully Off-line:** solutions that do not require any external connection but either assume involved devices to be trusted (Juang, 2013; Salama et al., 2011) or are limited to transactions tied to a bank account.

As already introduced, the main issue of a fully off-line solution is that keeping track of past transactions can be hard, as it is difficult for a vendor to check if some digital credits have already been spent. This is the main reason why the solutions proposed so far in the literature (see Table 1) require some kind of Trusted Third Party (TTP) to store past transactions and check such a list each time a new transaction is started (Vasco et al., 2010) (as such they can be considered run-time verified solutions). Alternatively, off-line solutions that do not rely on TTPs either assume a tamper proof/resistant smart card (such as (Juang, 2013; Salama et al., 2011)) or just check for customer identity (Wang et al., 2013) whereas security checks, such as double spending prevention, are verified and validated by the bank at a later time (such solutions are classified as postponed).

# 3 PROPOSED MODEL

FORCE is the first solution that neither requires TTPs, nor bank accounts, nor trusted devices to mitigate the attacks that usually affect fully off-line payment schemes (details in Section 6). To achieve such a goal, FORCE leverages physically unclonable functions (for short PUF, details in Section 4) and proposes a novel, fully off-line system based on digital credit, i.e. prepaid coins that can be spent only once.

Furthermore, by allowing FORCE customers to be free from having a bank account, makes it particularly interesting as regards privacy. In fact, unlike all other solutions, anyone can buy a FORCE scratch card (e.g. at a local reseller) without disclosing her identity. Digital credits used in FORCE are just a digital version of real cash and, as such, they are not linked to anybody else than the holder.

Differently from other payment solutions based on tamper-proof hardware, FORCE assumes that only the chips built upon PUFs can exploit the tamper evidence feature provided by the PUFs themselves. As a consequence, our assumptions are much less restrictive and more realistic than other approaches.

## 3.1 FORCE Model

FORCE can be applied to any scenario composed by a payer/customer device, a payee/vendor device, a scratch card (i.e. a digital credit physical wallet) and a payee/vendor local storage device. In its current version, as depicted in Figure 1, FORCE has been designed using a smartphone as the customer device (for short CD), a Point Of Sale as the vendor device (for short VD), and a Near Field Communication (Coskun et al., 2012) (for short NFC). The rationale behind the

Table 1: Some payment schemes with double spending attack prevention technique being adopted.

| Scheme | TTP-Free | TTP Type | Check Type |
|---|---|---|---|
| (Dai et al., 2006) | ✗ | P2P Network | Realtime |
| (Popescu and Oros, 2007) | ✗ | Database | Postponed |
| (Zhou, 2008) | ✗ | P2P Network | Postponed |
| (Srivastava et al., 2008) | ✗ | Database | Realtime |
| (Wang and Lu, 2008) | ✗ | Database | Postponed |
| (Zhan-gang and Zhen-kai, 2009) | ✗ | Database | Postponed |
| (Vasco et al., 2010) | ✗ | Issuing Authority | Realtime |
| (Salama et al., 2011) | ✗ | Database | Realtime |
| (Wang et al., 2013) | ✗ | Database | Postponed |
| (Juang, 2013) | ✗ | Database | Postponed |
| (Chaurasia and Verma, 2014) | ✗ | Database | Realtime |
| FORCE | ✓ | None | Realtime |

choice of NFC is that it is much easier to use compared to other wireless communication technologies like Bluetooth or WiFi.

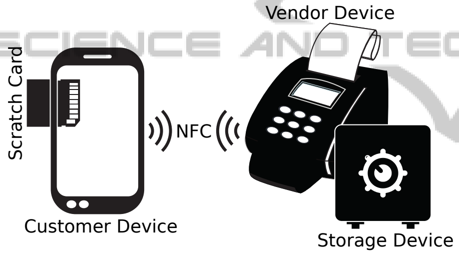

Figure 1: FORCE model.

In FORCE, as detailed in Section 6, all involved devices can be tweaked by an attacker and are considered untrusted except from the storage device, that we assume is kept physically secure by the vendor. It is important to highlight that such an assumption does not affect the security of the proposed system. In fact, similarly to physical wallets, bank's safety vaults or crypto-currency digital wallets, the storage device is not involved in the payment transaction and represents a secure, write-only, place where collected money are stored.

Furthermore, FORCE, rather than being an e-cash system, has been designed to be a secure and reliable encapsulation scheme of digital coins into digital credits. This makes FORCE also applicable to multiple-bank scenarios. Indeed, as for credit and debit cards where TTPs (i.e. card issuers) guarantee the validity of the cards, some common standard convention can be used in FORCE to make banks able to produce and sell their own scratch card. Any bank will then be capable of verifying digital credits of scratch cards issued by other banks, by requiring banks and vendors to agree on the standard used for the digital credits within the scratch card (see Section 4).

In contrast to all other solutions proposed so far, in FORCE, vendors are able to verify digital credit validity at run-time. This means that once a digital credit has been verified, it can be directly and immediately re-used (details in Section 5.4) and there is no way such a credit could be refused or reclaimed.

FORCE does not require any special hardware component apart from the scratch card that can be plugged into any device able to read SD cards. Similarly to a secure element (e.g MasterCard PayPass chip), our scratch card is a tamper proof device that provides a secure storage and execution environment for sensitive data. Thus, as defined in the ISO7816-4 standard, our scratch can be accessed via some APIs while maintaining the desired security and privacy level. Such software components (i.e. APIs) are not central to the security of the scratch card system and can be easily updated. This renders infrastructure maintenance easier.

## 3.2 Threat Model

In order to better describe all the possible threats a fully off-line environment is subject to, a detailed description of both attacks and attackers is introduced in this section. The first important distinction that has to be made is about the position of the adversary:

- **Internal Attacker:** this adversary is directly involved in the payment as the customer/vendor. As such, he is capable of tweaking any device either by injecting malicious code or by having physical access to it;
- **External Attacker:** this adversary is not directly involved in the payment. As such, he can only access/alter the data being exchanged between the vendor and the customer over the NFC channel.

The second classification is based upon the number of tweaked devices as follows:

- **Collector:** this is an external adversary able to eavesdrop and alter messages being exchanged between the CD and the VD;
- **Malicious Customer:** this is an internal adversary that can either physically open the CD to eavesdrop sensitive information or inject malicious code within the CD in order to alter its behavior;
- **Malicious Vendor:** (for short, M. Vendor) is an internal adversary that can either eavesdrop information from the VD or inject malicious code within the VD in order to alter its behavior;
- **Ubiquitous:** this is an internal adversary with complete access to both CD and VD.

In the proposed payment scheme, as shown in Section 6, no restrictions are made on the capabilities of the adversary, always considered as ubiquitous.

## 4 FORCE: ARCHITECTURE

In this section, the architecture of our mobile payment solution will be described (see Figure 2). The core element of the whole payment system architecture is a scratch card that can be built within the CD or used as a separate element, such as Secure Digital cards (for short SD). A scratch card is composed of:

- **Scratch Memory:** special read once memory used to store digital credits;
- **Authenticator:** used to compute, on the fly, all the cryptographic keys required for the payment protocol;
- **Memory Mapping Unit:** used to retrieve the digital credit layout and to detect malicious attacks based on guessing the memory layout.

Both the authenticator and the memory mapping unit elements are built upon physically unclonable functions. PUFs were introduced by Ravikanth (Ravikanth, 2001) in 2001. He showed that, due to manufacturing process variations, every transistor in an integrated circuit has slightly different physical properties that lead to measurable differences in terms of electronic properties. Since these process variations are not controllable during manufacturing, the physical properties of a device cannot be copied or cloned. As such, they are unique to that device and can be used for authentication purposes. However, creating a device with a given electronic fingerprint is difficult and expensive, whereas implementing a PUF requires an electronic circuit that is able to produce hardware outputs (i.e. responses) to given inputs (i.e. challenges). These responses depend on the unique physical properties of the device. As such, PUFs are functions that are easy to challenge and whose response is easy to measure, but very hard to reproduce. PUFs have been proposed in banking systems in the past but so far they have only been used to provide stronger customer authentication. One of the most important features about PUFs is their tamper-evidence capability.

In the remainder of this section, each element of the scratch card will be described. Further, in Section 5 the transaction protocol will be depicted.

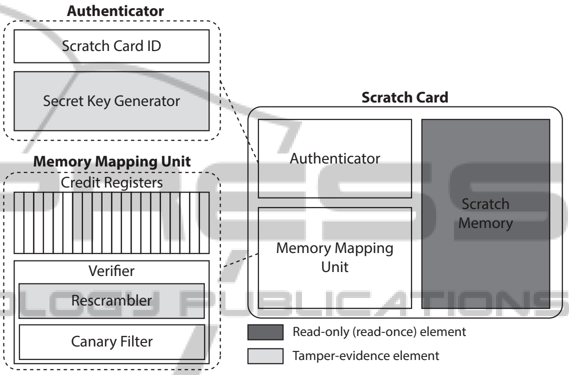

Figure 2: *FORCE* scratch card architecture. Elements in grey contain a physically unclonable function.

### 4.1 Scratch Memory

At the hearth of the scratch card lies a read-once memory(Rens, 2006) named scratch memory. Such memory, used to store digital credits, has the property that reading one value destroys/erases the original content.

FORCE is not tied/limited to any static digital credit format. It just requires each digital credit to be composed of at least two fields, namely the value of the digital credit and an integrity verification value. This value is used to guarantee that a specific credit is created to be spent by a specific scratch card only. Such value is computed at manufacturing time by first encrypting the credit value with the public key of the scratch card and then additionally signing it with the private key of the card issuer. This is to avoid forgery attacks. Once a digital credit has been created, it is stored within the scratch memory in a non contiguous way. During this step, the card issuer creates unique random sequences, one for each credit, where unique means that taken two credits $C_a$ and $C_b$ and given $S_a$ (the sequence of $C_a$) and $S_b$ (the sequence of $C_b$) then $S_a \cap S_b = \emptyset, \forall (a,b)$ with $a \neq b$. Such sequences represent the layout of each credit within the scratch memory.

FORCE does not rely either on a specific scratch memory size or on a specific number of digital credits. It is the card issuer that has the responsibility of managing the scratch memory layout as regards both the size and the credit number. As such, FORCE can work with scratch memories of any size and with any number of digital credits. It is also important to highlight that the scrambled layout of digital credits within the scratch memory is not the core security element of the solution proposed in this work. Digital credit layout is only meant to prevent a subset of attacks based on the guessing of the scratch memory (see Section 7 for details).

## 4.2 Authenticator

The authenticator is used to on-the-fly compute the scratch card's private key used to decrypt vendor requests. In fact, rather than embodying a single cryptographic key within the device, thus potentially allowing an adversary to steal it, PUFs have been used in FORCE to implement strong challenge-response authentication. The challenge used as input for the PUF is a publicly known scratch card identifier hard-coded within the card and used in the payment protocol as the public key of the card. Each scratch card is indeed shipped with a public key, signed by the bank/card issuer to avoid forgery attacks and hard-coded into the card itself. This allows the customer to broadcast the public key of the card to vendors. As such they are not required to know all the public keys of all the active scratch cards.

As detailed in Section 5, vendors can encrypt payment requests with the public key of a scratch card with the guarantee that such requests will be read only by that card. Further, the tamper-evidence feature of PUFs ensures that any attempt to open on the fly the authenticator element to read the computed private key will alter the behavior of the PUF causing a different key to be produced and thus the loss of the original key. Changing the original private key leads to the impossibility to read vendor requests thus rendering the whole scratch card useless.

## 4.3 Memory Mapping Unit

The memory mapping unit (MMU) is composed by a set of credit registers and by a verifier element as shown in Figure 2. Credit registers are hard-coded into the MMU and each of them is given as input to the rescrambler-PUF to compute the actual layout of each digital credit within the scratch memory (see Figure 3). Again, actual layout values of digital credits are not stored anywhere within the card but are computed on the fly each time, making it hard for an adversary to eavesdrop them.

The latest element of the MMU is the canary filter, embedded into the verifier and used to protect the scratch card from memory guess-based attacks by using special bits (canary bits). These bits have the main goal of keeping track of scratch memory malicious accesses and, as depicted in Figure 3, they are designed as a mapping function between input and output. If a bit given as input to the canary filter matches a canary bit, the output is multiplexed to the whole scratch memory. This guarantees that any attempt to read a canary bit will automatically cause the entire scratch memory to be read and, as such, erased. As for the authenticator, the MMU takes advantage of the tamper-evidence feature of its embedded rescrambler-PUF.

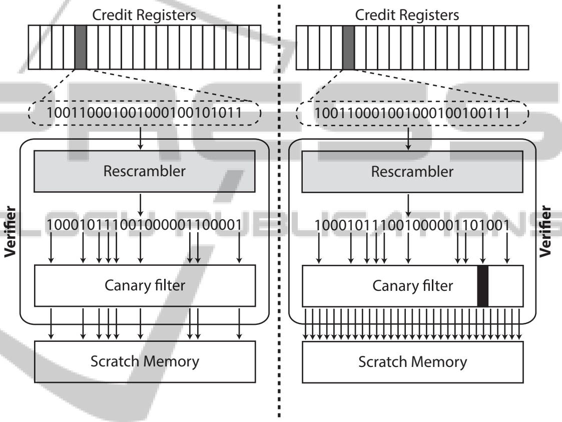

Figure 3: An authorized read on the left and a malicious read on the right. On the right, the canary bit is multiplexed to all the addresses of the scratch memory.

## 4.4 Stable PUF Extraction

As described in Section 4.2 and Section 4.3, physically unclonable functions have been used in FORCE to compute the private key of the scratch card and the actual layout value of each digital credit. However, given a fixed input, PUFs can produce a responses that is unique to the manufacturing instance of the PUF circuit but that it is not bitwise-identical when regenerated multiple times. As such, in order to use PUFs in algorithms where stable values are required, an intermediate step is required. This problem, usually faced in cryptographic algorithms, is known as "secret key extraction" and it can be solved using a two-step algorithm. In the first step the PUF is queried, thus producing an output together with some additional information called *helper data*. In the second step, the helper data is used to extract the same output as in the first step thus making the PUF able to build stable values. It is also possible to construct a two-step algorithm guaranteeing that the computed value is perfectly secret, even if the helper data is publicly known. Practical instances of such kind of algorithm have been proposed in (Dodis et al., 2008) and the cost of actual implementations thereof is assessed in (Maes et al., 2009).

Recently, some solutions have been proposed to correct PUF output on the fly thus providing the generation of secret stable values within the device. FORCE uses this approach for the design of both the key generator element (embedded in the authenticator) and for the verifier element (embedded in the MMU). Such special PUFs are built upon a lightweight error correction algorithm proposed in (Yu et al., 2011) and described in this section.

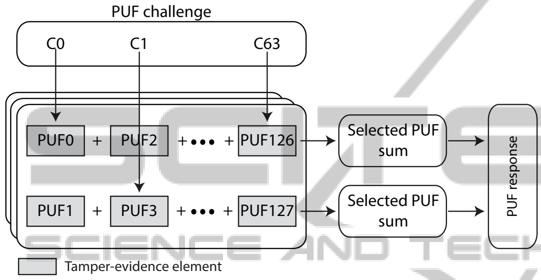

Figure 4: Stable PUF response computation approach used for the key generator and the MMU verifier.

As depicted in Figure 4, the basic 64-sum PUF looks at the difference between two delay terms, each produced by the sum of 64 PUF values. Given a challenge, its $i^{th}$ bit called $C_i$ determines, for each of the 64 stages, which PUF is used to compute the top delay term, and which is used to compute the bottom delay term. The sign bit of the difference between the two delay terms determines whether the PUF outputs a '1' or '0' bit for the 64-bit challenge $C_0 \cdots C_{63}$. The remaining bits of the difference determine the confidence level of the '1' or the '0' output bit. The k-sum PUF can be thought of as a k-stage Arbiter PUF (Lim et al., 2005) with a real-valued output that contains both the output bit as well as its confidence level. This information is used by the downstream lightweight error correction block that is able to produce in output a stable value within the scratch card.

By using such on the fly stable value generation process, FORCE does not store either private keys or digital credit actual layout within the customer device thus protecting them from malicious customers and ensuring that only the right scratch card can compute its own private key with a single step each time it is needed.

# 5 FORCE: PROTOCOL

This section describes all phases of the FORCE protocol. For completeness, the *Redemption*, *Transaction Dispute* and the *Rollover* phases will be analyzed even though they are not part of the payment procedure that is composed by the *Pairing* and the *Payment* phases only.

## 5.1 Pairing Phase

The current version of FORCE uses the NFC technology for all the communications between CDs and VDs. Even though NFC requires both the involved devices to be very close to each other, an adversary could still be able to unleash man-in-the-middle attacks (for short MITM) by using NFC boosters. As such, a pairing setup process has being used as the first step of each new transaction request. For the pairing phase, FORCE relies on standard and well known pairing protocols such as the Passkey Entry of the Bluetooth Simple Pairing Process (for short SSP). At the end of the pairing protocol, both the devices will share their public keys used to guarantee integrity and authenticity of messages being exchanged. Furthermore, in order to avoid brute force attacks on the pairing protocol, FORCE adopts a "fail-to-ban" approach based upon a failure threshold value. In this case, if a malicious customer consecutively fails the Passkey Entry procedure the system stops for few seconds, usually 20 or 30 seconds. If the number of consecutive fail-to-ban reaches a security threshold value, the vendor can decide to refuse the pairing request.

For the sake of simplicity, all the encryption operations involved in the SSP Passkey Entry protocol and used in the FORCE pairing process will be omitted.

## 5.2 Payment Phase

The FORCE payment phase is depicted in Figure 5 and it is composed by the following steps (symbols in Table 2):

1. The customer sends a purchase request to the VD asking for some goods;
2. The vendor computes the total amount and sends it back to the customer;
3. The customer checks for the amount and either confirms or denies the transaction. If the transaction is confirmed, the CD creates a reply for the VD with the indexes of all the credits that are still available in the card. If the $i^{th}$ index number is present in the reply, it means that the $i^{th}$ credit register can be read in order to retrieve the $i^{th}$ digital credit within the card;

Table 2: Symbols used in all the phases of the transaction protocol

| Symbol | Meaning |
|---|---|
| Enc()/Dec() | Symmetric encryption/decryption |
| $\underline{\text{Enc}}$()/$\underline{\text{Dec}}$() | Asymmetric encryption/decryption |
| Salt | Salt value |
| CreditIdx | Credit memory addresses |
| CreditVal | Credit memory content |
| Req | Credit request built by VD |
| Res | Response built by CD |
| CPK | Card public key |
| CSK | Card secret (private) key |
| BPK | Bank/Card Issuer public key |
| BSK | Bank/Card Issuers secret (private) key |
| VPK | Vendor public key |
| VSK | Vendor secret (private) key |
| EReq | Encrypted request |
| ERes | Encrypted response |
| FRes | Final response |
| RReq | Redemption request |
| Log | Log entry |
| ELog | Encrypted log entry |

4. The vendor first creates a random salt value. Then, for each credit that will be involved in the transaction, a request is created by encrypting the credit index with the random salt obtaining *Req*

$$Enc_{Salt}(CreditIdx) = Req \tag{1}$$

5. Such encrypted request along with the salt just created are encrypted once again with the public key of the scratch card, thus rendering the customer the only party able to read it

$$\underline{Enc}_{CPK}(Req, Salt) = EReq \tag{2}$$

6. When the customer receives such a request, the private key is computed by the authenticator as shown in Section 4.2 and it is used to decrypt the message received thus obtaining the salt value and the request

$$\underline{Dec}_{CSK}(EReq) = (Req, Salt) \tag{3}$$

7. The salt is then used to decrypt the request *Req*

$$Dec_{Salt}(Req) = CreditIdx \tag{4}$$

8. CreditIdx is used by the MMU to read the scratch card digital credit value (details in Section 4.3);
9. The credit value is sent back to the authenticator;
10. The salt is used once again to create an encrypted response for the vendor

$$Enc_{Salt}(CreditVal) = Res \tag{5}$$

11. The response is encrypted with the private key of the card thus providing authenticity and integrity

$$\underline{Enc}_{CSK}(Res) = ERes \tag{6}$$

12. The encrypted response is then sent back to the vendor;

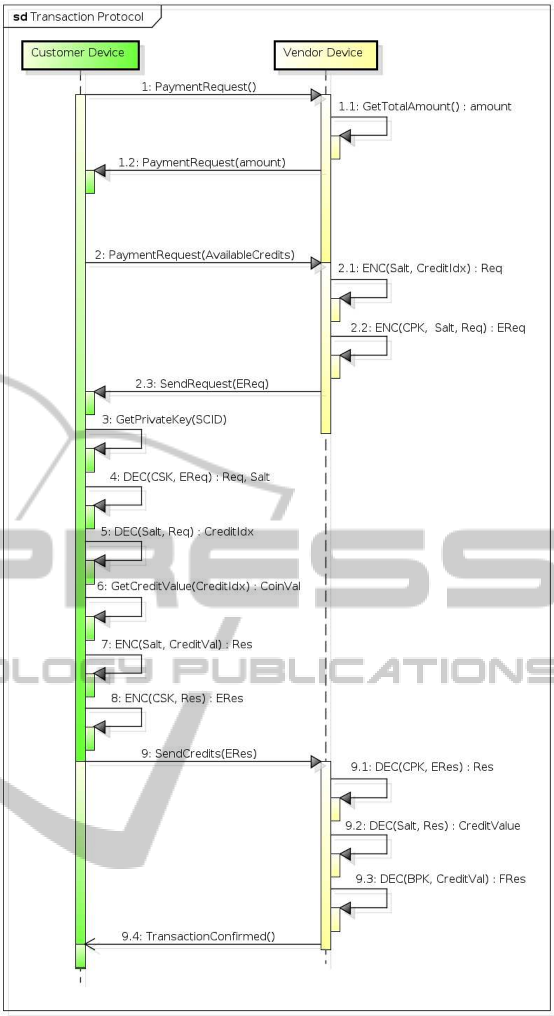
sd Transaction Protocol
Customer Device
Vendor Device
1: PaymentRequest()
1.1: GetTotalAmount() : amount
1.2: PaymentRequest(amount)
2: PaymentRequest(AvailableCredits)
2.1: ENC(Salt, CreditIdx) : Req
2.2: ENC(CPK, Salt, Req) : EReq
2.3: SendRequest(EReq)
3: GetPrivateKey(SCID)
4: DEC(CSK, EReq) : Req, Salt
5: DEC(Salt, Req) : CreditIdx
6: GetCreditValue(CreditIdx) : CoinVal
7: ENC(Salt, CreditVal) : Res
8: ENC(CSK, Res) : ERes
9: SendCredits(ERes)
9.1: DEC(CPK, ERes) : Res
9.2: DEC(Salt, Res) : CreditValue
9.3: DEC(BPK, CreditVal) : FRes
9.4: TransactionConfirmed()

Figure 5: Payment protocol. Operations from 3 to 8 are executed within the scratch card but the whole process is invoked by the customer device.

13. The vendor decrypts the *ERes* in two steps

$$\underline{Dec}_{CPK}(ERes) = Res \tag{7}$$

$$Dec_{Salt}(Res) = CreditVal \tag{8}$$

14. Finally the content of the credit is decrypted with the public key of the bank/card issuer

$$\underline{Dec}_{BPK}(CreditVal) = FRes \tag{9}$$

15. If the credit value is correct, a new entry is stored in the storage device of the vendor after having being encrypted with the private key of the vendor.

If all the steps are accomplished without errors (see Section 5.3) the transaction is authorized and the purchase is allowed. It is important to highlight that, as already described in Section 1, FORCE has been designed as a secure and reliable encapsulation scheme rather than as an e-cash system. As such, problems affecting digital currencies, such as digital change, are beyond the scope of the proposed solution and will not be analyzed here.

### 5.3 Transaction Dispute

Due to its truly off-line nature, FORCE does not provide a transaction dispute protocol phase to better protect both the customer and the vendor. Indeed, a malicious customer could simulate an error in the transaction, thus requesting a direct refund to the vendor, while a malicious vendor could simulate an invalid transaction, even if digital credits were successfully read from the customer's scratch card.

As such, direct transaction disputes between vendors and customers are avoided while on-line transaction disputes are allowed. In fact, since the redemption phase is on-line (see Section 5.4), the correctness and completeness of each off-line transaction can be easily verified by the bank/card issuer thus rendering a fake transaction dispute attempt too risky and unfeasible to the malicious party.

### 5.4 Redemption Phase

Vendors accepting FORCE scratch cards from their customers can verify digital credits at run-time without relying on any TTP. This is thanks to the fact that what is actually exchanged between the customer and the vendor is not a promissory note (as with credit cards and all other postponed payment schemes that claim to be off-line) but it is a digital value, representing real money, signed by the bank/card issuer. As such, each FORCE payment transaction just needs the pairing and the payment phases in order to be accomplished and evaluated by the vendor. However, for the sake of completeness, the redemption phase will also be briefly discussed.

As shown in Figure 6, once the off-line transaction has been completed, the vendor owns the digital credit just received from the customer. Such credit is encrypted by the bank/card issuer and, as such, it can be easily verified by everyone using the public key of the bank/card issuer. Thus, once the credit has been verified, the vendor can use the digital coin (encapsulated within the credit) either to send it back to the bank/card issuer in exchange for real money or to use it as a common crypto/digital currency. If the vendor chooses to send it back to the bank/card issuer, the credit and the coins will be stored in the bank database. On the contrary, if the vendor decides to use the credit as an e-cash digital coin, the credit will be broadcast over the network depending on the payment scheme being used.

This "second-step" payment process relies on common on-line payment protocols. Thus, its security and reliability features are not discussed here.

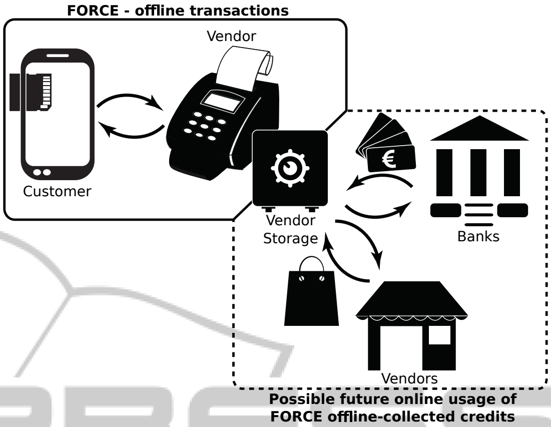
FORCE - offline transactions
Vendor
Customer
Vendor
Storage
Banks
Vendors
Possible future online usage of
FORCE offline-collected credits

Figure 6: Possible uses of digital credit obtained in past transactions.

## 6 FORCE: SECURITY ANALYSIS

In this section the robustness of FORCE is discussed. FORCE uses both symmetric and asymmetric crypto schemes in order to guarantee the most important security principles as follows:

- **Authenticity:** it is guaranteed by FORCE both in the pairing and in the payment transaction phase. In the pairing phase the authenticity is ensured by the PassKey Entry standard protocol while in the payment phase the authenticity is ensured by the authenticator element embedded in the scratch card;
- **Non Repudiation:** the storage device kept physically safe by the vendor prevents the adversary from deleting past transactions thus avoiding malicious repudiation requests. In addition, the content of the storage device can be exported to an external storage, such as a pen drive, on a timely basis;
- **Integrity:** digital credit integrity is ensured by encrypting each digital credit with the private key of the bank/card issuer. Furthermore, message integrity is ensured by FORCE thanks to the on the fly computation of the private key of the scratch card that is not stored anywhere within the scratch card but it is computed each time as needed;
- **Confidentiality:** each response sent to the vendor by the customer has a double layer encryption. Responses are first encrypted with the random salt generated by the vendor at the beginning of the payment phase and then encrypted

with the private key of the scratch card. This second layer of encryption ensures that the response was originated by that card as described in Section 4.2 while the encryption layer built upon the salt, guarantees confidentiality and freshness of the response generated by the card;

- **Availability:** the availability of the proposed solution is guaranteed mainly by: first, the fully off-line scenario completely removes any type of external communication requirement and makes it possible to use off-line digital credits also in extreme situations with no network coverage. Second, the implementation with a passive card and the lack of any registration phase, makes the proposed scratch card able to be used by different devices.

FORCE shares (Choi and Kim, 2012) the assumption that the scratch card is tamper-evident. This assumption is based on the size of nowadays IC and on the impossibility for a casual adversary to open the device without causing an alteration in PUF behavior. This assumption is no longer valid if an expert adversary with access to highly sophisticated and expensive tools, such as scanning electron microscopes or focused ion beams, is taken into account. However, such tools can be worth thousands of dollars and applying this kind of attack on each single device to steal a few dollars will not be convenient to the attacker.

### 6.1 Key Rollover

As for all the real-world payment schemes based on smart cards such as credit, debit and prepaid cards, FORCE assumes that, in case of bank/card issuer private key renewal a time-window is adequately chosen to let customers decide whether to spend their last credits or to exchange the current card with a new one. These standard procedures are widely accepted in the real world and, as such, no custom key rollover protocol has been designed in FORCE.

## 7 FORCE: ATTACK MITIGATION

In this section, the resiliency of FORCE to all the attacks listed in Section 3.2 is discussed:

- **Double Spending:** the read once property of the scratch memory prevents an adversary from reading the same digital credit twice. Even if a malicious customer creates a fake vendor device and reads all digital credits (as described in Section 4.1) it will not be able to spend such credit due to the inability to decrypt the request of other vendors (see the payment protocol in Section 5.2). Indeed, as described in Section 4.2 the private key of the card is needed to decrypt the request of the vendor and can be computed only within the device. The fake vendor could then try to forge a new emulated scratch card with private/public key pair. However, scratch card public keys are valid only if signed by the bank/card issuer. As such, any message received by an unconfirmed scratch card will be immediately rejected;
- **Credit Forgery:** each credit is encrypted with the private key of the bank/card issuer and thus it is not possible for an adversary to forge new credit;
- **Memory Poisoning:** each completed transaction is kept in the vendor storage device. If a digital credit has been corrupted by a memory poisoning attack, such credit will not be accepted. Such corrupted and unused credits can be claimed back to the bank/card issuer that will check for both vendor logs and on-line payment circuit databases and if such credit is not present in any of them a refund will be given back to the victim;
- **Memory Deletion:** this is a special case of the Memory Poisoning attack in which all credits are corrupted;
- **Memory Dump:** as shown in Sections 4.2 and 4.3 opening the scratch card to copy the content of the scratch memory will alter the behavior of the PUF, thus invalidating the whole scratch card;
- **Memory Reconstruction:** this is a special case of the memory dump. By attempting a memory reconstruction the adversary could be able to reconstruct each digital credit and then use them in future transactions. However, reading the memory for dumping will change the PUF behavior, thus preventing the authenticator from computing the CD private key required to decrypt the vendor request;
- **Hardware Emulation:** PUFs, by design, cannot be neither dumped nor forged as in this case computed responses will be different from the original ones;
- **Software Emulation:** it is not possible, by design, to emulate PUFs without opening them and, thus, corrupting them;
- **Postponed Transaction:** the only way to either forcibly access or eavesdrop clear-text information is by physically opening the scratch card. Again, doing so will alter PUFs behavior thus invalidating the whole card;
- **Information Stealing:** as shown in Section 6 the private key of the CD and the real layout of each credit is computed on the fly as needed. No sensitive information is kept in the scratch card;
- **Replay:** each challenge, even if related to the

same digital credit, is different due to the random salt generated each time by the vendor;

- **Man In the Middle:** digital credits are encrypted by the bank/card issuer and contain, among all other things, the scratch card ID. As a consequence, an adversary cannot spend digital credits of other customers by simply copying them from the scratch card of the victim. Even changing the content of the victim's digital credit by replacing the original ID with the ID of the adversary is not possible. After such alteration of the credit, the adversary would not be able to encrypt the credit with the private key of the bank/card issuer, thus rendering the malicious credit useless. Last but not least, the adversary cannot pretend to be another customer with a different ID because it will not be able to compute the private key linked to that ID;
- **Reverse Engineering:** by design, any attempt to tweak the scratch card in order to try and steal any useful information will alter the behavior of the PUF thus rendering the whole scratch card no longer usable;
- **Denial of Services:** the pairing process, based on the Bluetooth Passkey Enter standard protocol, cannot be accomplished by an adversary because it requires a security code to be manually and physically typed on the customer's device. As such, DoS and DDoS attacks where the adversary wipes the credits on the SC are mitigated. Even if the adversary is a malicious vendor, each transaction has to be confirmed by the customer thus preventing batch attacks where the SC is repeatedly challenged;
- **HW Modification:** again, by design, it is not possible for an adversary to add/modify/remove any element belonging to the scratch card without changing its behavior;
- **HW Eavesdropping:** it is well known that nowadays photon counting APD modules and photon emission microscope with InGaAs image sensors are used with Focused Ion Beam (FIB) systems in order to locate faults within integrated circuits. However, as explained at the beginning of this section, we consider this kind of attack overkill.
- **Repudiation:** as described in Section 5.3, FORCE does not provide a transaction dispute protocol phase. However, while the payment transaction is accomplished in a fully-offline scenario, any additional operation (e.g. disputes or refund requests) can be accomplished on-line. This way, the customer cannot repudiate a valid transaction (the log entry for that transaction will be notified on-line by the vendor) and the same applies for the vendor (a repudiated valid transaction cannot be spent).

So far, we have discussed the resilience of our payment scheme to the attacks introduced in Section 3.2. In the following, other considerations are shared based on the different adversary models introduced in the same Section above:

- **Malicious Customer:** as shown at the beginning of this Section, forgery, dump and reply attacks are mitigated by the architecture and physical nature of the core elements of the scratch card.
- **Malicious Vendor:** the only feasible attack for a malicious customer is the deletion of past transaction entries from the storage device. However, this is not possible as the storage device is assumed to be kept physically secure by the vendor;
- **Ubiquitous:** the smarter attack that can be unleashed by such an adversary is the stealing of information from both the VD and the CD to reconstruct the semantics of the scratch card memory content. However, in order to steal such information the adversary has to physically tweak the scratch card, thus invalidating it.

The robustness of FORCE is mainly based on PUFs features but also on the high unpredictability of digital credit layout within the scratch memory. As regards physical attacks to PUFs, Integrated Circuits (ICs) and hardware in general, some relevant results are discussed in (Griffin et al., 2012) and (Choi and Kim, 2012). The first one aims at protecting IC integrity as each manufactured IC is rendered inoperative unless a unique per-chip unlocking key is applied. After manufacturing, the response of each chip to specially generated test vectors is used to construct the correct per-chip unlocking key. As concerns (Choi and Kim, 2012), Choi and Kim aimed to protect the keys inside TPMs using a PUF. In fact, when the keys are stored in memory and when they are moved through the bus, their value is changed with the PUF, thus rendering eavesdropping out of the PUF IC useless. When the keys are needed for the cryptographic module, they are retrieved from outside the PUF IC and decrypted by the same PUF. However, the values of the keys could be revealed through side-channel attacks, e.g. non-invasive forms of physical attack measuring timings, power consumption, and electromagnetic radiation. Most cryptographic modules are known to be vulnerable to side-channel attacks, and these attacks would be effective against the TPM; thus, countermeasures against side-channel attacks are necessary.

## 8 DISCUSSION

The problem of limiting data access in a physical device is extremely difficult. Attacks that try to infer information from a device can be categorized as *passive* or *intrusive* attacks. In *passive* attacks the system interface is probed for either timing or electrical differences. In *intrusive* attacks the adversary is able to breach the physical boundary of the package and can scan, probe or alter the hardware itself.

In FORCE, on the one hand, intrusive attacks are not feasible as they alter the functionality of the scratch card. On the other hand, passive attacks have been analyzed by subdividing them into *powered* and *un-powered* attacks. In *powered* attacks the device is monitored while running whilst in *un-powered* attacks, information is extracted from the device while the hardware is not powered on. In FORCE no value used by the protocol is permanently stored in the CD. As such, un-powered attacks are mitigated. On the contrary, a run-time attack using extremely complex monitoring tools could have access to the values being computed during each step of the protocol. However, stealing information on the fly at run-time would require extremely expensive instrumentation whose cost is well beyond the relatively small amount of money that can be stored in a scratch card. Further, a successful extraction of data from a scratch card will not reveal any useful information about other scratch cards, even if they are shipped by the same card issuer. As such, as already discussed in Section 6 we can safely assume that this kind of attack is not worth the effort and, as such, it is considered overkill.

## 9 CONCLUSION AND FUTURE WORK

In this paper we have presented the first fully off-line approach for micro-mobile payments. We have described how our solution provides a higher security level without any trustworthiness assumption over the devices involved in the payment protocol. This has mainly been achieved by leveraging PUF properties and a special read-once memory where our digital credits have been stored using a highly unpredictable layout. Our proposal has been thoroughly discussed with reference to state of the art solutions. Features such as feasibility and convenience have been shown.

Finally, some open issues that will require further investigation have been identified. In particular, present FORCE only allows each off-line credit to be spent once. We are working on an enhanced version of FORCE that will allow digital credit to be spent in multiple off-line transactions while maintaining the same level of security and usability.

## REFERENCES

Aigner, M., Dominikus, S., and Feldhofer, M. (2007). A System of Secure Virtual Coupons Using NFC Technology. In *IEEE PerComW'07*, pages 362–366. IEEE.

Chaurasia, B. K. and Verma, S. (2014). Secure pay while on move toll collection using {VANET}. *Computer Standards & Interfaces*, 36(2):403–411.

Chen, W., Hancke, G., Mayes, K., Lien, Y., and Chiu, J.-H. (2010). Using 3G network components to enable NFC mobile transactions and authentication. In *IEEE PIC '10*, volume 1, pages 441 –448.

Choi, P. and Kim, D. K. (2012). Design of security enhanced TPM chip against invasive physical attacks. In *IEEE ISCAS '12*, pages 1787–1790.

Coskun, V., Ok, K., and Ozdenizci, B. (2012). *Near Field Communication: From Theory to Practice*. Wiley Publishing, 1st edition.

Dai, X., Ayoade, O., and Grundy, J. (2006). Off-line micro-payment protocol for multiple vendors in mobile commerce. PDCAT '06, pages 197–202, Washington, DC, USA. IEEE Computer Society.

Dodis, Y., Ostrovsky, R., Reyzin, L., and Smith, A. (2008). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139.

Dominikus, S. and Aigner, M. (2007). mcoupons: An application for near field communication (nfc). AINAW '07, pages 421–428, Washington, DC, USA. IEEE Computer Society.

Golovashych, S. (2005). The technology of identification and authentication of financial transactions. from smart cards to NFC-terminals. In *IEEE IDAACS '05*, pages 407–412.

Griffin, W. P., Raghunathan, A., and Roy, K. (2012). Clip: Circuit level ic protection through direct injection of process variations. *IEEE Trans. Very Large Scale Integr. Syst.*, 20(5):791–803.

Juang, W.-S. (2013). An efficient and practical fair buyer-anonymity exchange scheme using bilinear pairings. In *Asia JCIS*, pages 19–26.

Kadambi, K. S., Li, J., and Karp, A. H. (2009). Near-field communication-based secure mobile payment service. In *ICEC '09*. ACM.

Lewandowska, J. (2013). http://www.frost.com/prod/servlet/press-release.pag?docid=274238535.

Lim, D., Lee, J. W., Gassend, B., Suh, G. E., van Dijk, M., and Devadas, S. (2005). Extracting secret keys from integrated circuits. *IEEE Trans. Very Large Scale Integr. Syst.*, 13(10):1200–1205.

Maes, R., Tuyls, P., and Verbauwhede, I. (2009). Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs. CHES '09, pages 332–347, Berlin, Heidelberg. Springer-Verlag.

Martins, S. and Yang, Y. (2011). Introduction to bitcoins: a pseudo-anonymous electronic currency system. CASCON '11, pages 349–350, Riverton, NJ, USA. IBM Corp.

Nishide, T. and Sakurai, K. (2011). Security of offline anonymous electronic cash systems against insider attacks by untrusted authorities revisited. INCOS '11, pages 656–661, Washington, DC, USA. IEEE Computer Society.

Patil, V. and Shyamasundar, R. K. (2004). An efficient, secure and delegable micro-payment system. EEE '04, pages 394–404, Washington, DC, USA. IEEE Computer Society.

Popescu, C. and Oros, H. (2007). An off-line electronic cash system based on bilinear pairings. In *EURASIP '07*, pages 438–440.

Ravikanth, P. S. (2001). *Physical one-way functions*. PhD thesis, Massachusetts Institute of Technology.

Rens, B. J. E. V. (2006). Authentication using a read-once memory. http://www.google.com/patents/US7059533. Accessed: 2013-07-30.

Rivest, R. L. (1996). Payword and micromint: two simple micropayment schemes. In *CryptoBytes*, pages 69–87.

Salama, M. A., El-Bendary, N., and Hassanien, A. E. (2011). Towards secure mobile agent based e-cash system. In *1st Intl. Workshop on Security and Privacy Preserving in e-Societies*, pages 1–6, New York, NY, USA. ACM.

Sekhar, V. C. and Mrudula, S. (2012). A complete secure customer centric anonymous payment in a digital ecosystem. *ICCEET '12*.

Srivastava, A., Kundu, A., Sural, S., and Majumdar, A. (2008). Credit card fraud detection using hidden markov model. *IEEE Transactions on Dependable and Secure Computing*, 5(1):37–48.

Vasco, M. G., Heidarvand, S., and Villar, J. (2010). Anonymous subscription schemes: A flexible construction for on-line services access. In *SECRYPT '10*, pages 1–12.

Wang, C. and Lu, R. (2008). An ID-based transferable off-line e-cash system with revokable anonymity. In *Intl. Symp. on Electronic Commerce and Security '08*, pages 758–762.

Wang, C., Sun, H., Zhang, H., and Jin, Z. (2013). An improved off-line electronic cash scheme. In *ICCIS '13*, pages 438–441.

Yu, M.-D. M., M'Raihi, D., Sowell, R., and Devadas, S. (2011). Lightweight and secure PUF key storage using limits of machine learning. CHES'11, pages 358–373, Berlin, Heidelberg. Springer-Verlag.

Zhan-gang, W. and Zhen-kai, W. (2009). A secure off-line electronic cash scheme based on ECDLP. In *ETCS '09*, volume 2, pages 30–33.

Zhou, X. (2008). Threshold cryptosystem based fair off-line e-cash. In *IITA '08*, volume 3, pages 692–696.