International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

Privacy Preserving In Off_Line System

Student:
E.NAGAVENI,
M.SC. (Computer science),
RIIMS, Tirupati,
Email id:nagavenielluri@gmail.com.

Head:
K.SUNITHA,MCA,M.Tech
Department of Computer Science,
RIIMS, Tirupati
Emailid:kkiran020318@gmail.com

ABSTRACT

Credit and debit card data theft is one of the earliest forms of cybercrime. Still, it is one of the most common nowadays. Attackers often aim at stealing such customer data by targeting the Point of Sale (for short, PoS) system, i.e. the point at which a retailer first acquires customer data. Modern PoS systems are powerful computers equipped with a card reader and running specialized software. Increasingly often, user devices are leveraged as input to the PoS. In these scenarios, malware that can steal card data as soon as they are read by the device has flourished. As such, in cases where customer and vendor are persistently or intermittently disconnected from the network, no secure on-line payment is possible. This paper describes FRoDO, a secure off-line micro-payment solution that is resilient to PoS data breaches. Our solution improves over up to date approaches in terms of flexibility and security. To the best of our knowledge, FRoDO is the first solution that can provide secure fully off-line payments while being resilient to all currently known PoS breaches. In particular, we detail FRoDO architecture, components, and protocols. Further, a thorough analysis of FRoDO functional and security properties is provided, showing its effectiveness and viability.

INTRODUCTION

Market analysts have predicted that mobile payments overtake the traditional marketplace, thus pro-viding greater convenience to consumers and new sources of revenue to many companies . This scenario produces a shift in purchase methods from classic credit cards to new approaches such as mobile-based payments, giving new market entrants novel business chances. 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 pioneering micro-payment scheme, was proposed by Rivest (see Payword) back in 1996. Nowadays, crypto-currencies and decentralized payment systems (e.g., Bitcoin) are increasingly popular, fostering a shift from physical to digital currencies. However, such payment techniques are not yet commonplace, due to several unresolved issues, including a lack of widely-accepted standards, limited interoperability among systems and, most importantly, security.

Over the last years, several retail organizations have been victims of information security breaches and payment data theft targeting consumer payment card data and personally identifiable information (PII). Although PoS breaches are declining, they still remain an extremely lucrative endeavor for criminals. Customer data can be used by cybercriminals for fraudulent operations, and this led the payment card industry security standards council to establish data security standards for all those organizations that handle credit, debit, and ATM cardholder information. Regardless of the structure of the electronic payment system, PoS systems always handle critical information and, oftentimes, they also require remote management. Usually, as depicted in Fig. 1, PoS systems act as gateways and require some sort of network connection in order to con-tact external credit card processors. This is mandatory to validate transactions. However, larger businesses that wish to tie their PoSes with other back-end systems may connect the former to their own internal networks. In addition, to reduce cost and simplify administration and maintenance, PoS devices may be remotely managed over these internal networks. However, a network connection might not be available due to either a 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. Most PoS attacks can be attributed to organized criminal groups. Brute forcing remote access connections and using stolen credentials remain the primary vectors for PoS intrusions. However, recent developments show the resur-gence of RAM scraping malware. Such attacks, once such malware is installed on a PoS terminal, can monitor the system and look for transaction data in plain-text, i.e., before it is encrypted. This paper introduces and discusses FRoDO, a secure off-line micro-payment approach using multiple physical unclonable functions (PUFs). FRoDO features an identity element to authenticate the customer, and a coin element where coins are not locally stored, but are computed on-the-fly when needed.

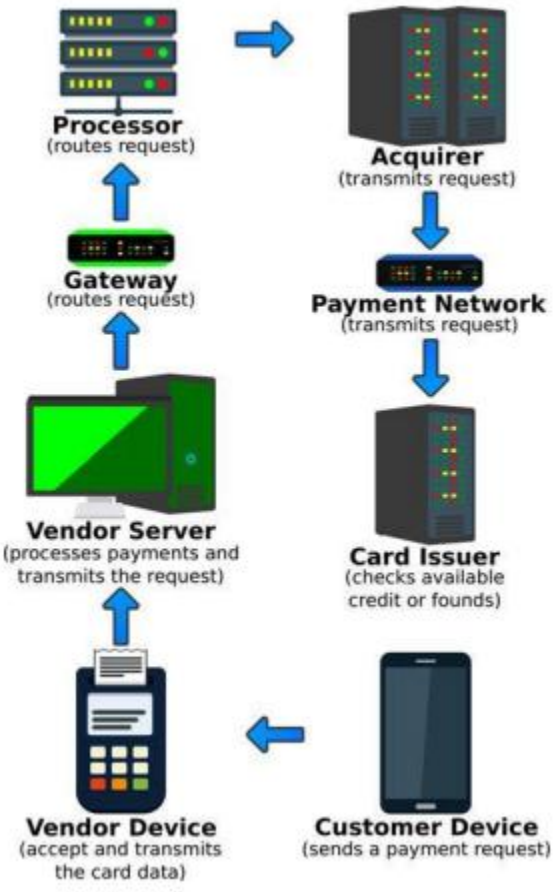

Fig 1. Payment authorization stages

The communication protocol used for the payment transaction does not directly read

customer coins. Instead, the vendor only communicates with the identity element in order to identify the user. This simplification alleviates the communication burden with the coin element that affected our previous approach . The main benefit is a simpler, faster, and more secure interaction between the involved actors/entities. Among other proper-ties, this two-steps protocol allows the bank or the coin element issuer to design digital coins to be read only by a certain identity element, i.e., by a specific user. Furthermore, the identity element used to improve the security of the users can also be used to thwart malicious users. To the best of our knowledge, this is the first solution that can provide secure fully off-line payments while being resilient to all currently known PoS breaches.

LITERATURE REVIEW

Mobile payment solutions proposed so far can be classified as fully on-line semi off-line weak off-line or fully off-line. The main issue with a fully off-line approach is the difficulty of checking the trustworthiness of a transaction without a trusted third party. In fact, keeping track of past transactions with no available connection to external parties or shared databases can be quite difficult, as it is difficult for a vendor to check if some digital coins have already been spent. This is the main reason why during last few years, many different approaches have been proposed to provide a reliable off-line payment scheme. Although many works have been published, they all focused on transaction anonymity and coin unforgeability. However, previous solutions lack a thorough security analysis. While they focus on theoretical attacks, discussion on real world attacks such as skimmers, scrapers and data vulnerabilities is missing. As regards physical unclonable functions, a key component of our solution, other applications on banking scenarios have already been proposed in the past. However such strong functions are generally used for authentication purposes only. As such, they only guarantee that data has been computed on the right device but they cannot provide any proof about the trustworthiness of the data itself.

One of the most relevant differences between [8] and FRoDO is the technology used to compute digital coins. FORCE used a read-once memory to randomly store digital coins and a physical unclonable function to recover their layout. This approach has been proven resilient against casual fraudsters. However, FORCE is vulnerable to advanced attacks based on the exfiltration of sensitive data when they are in transit or at rest . To mitigate such threats FRoDO does not use persistent memories at all but an erasable physical unclonable function. (details in Section 5.1); Protocol. While in [8] the vendor had to directly interact with the coin card, in FRoDO the vendor only interacts with the identity element. Such an element identifies a user (i.e., his device) and has the burden to communicate with the coin element. This new approach provides a number of advantages with respect to FORCE. On the one hand, customers' pri-vacy protection is enhanced as the vendor device is not aware of the amount and size of the digital coins written into the coin element. The vendor just sends a payment request message

containing the required amount of money. It is the identity element that will locally and internally interact with the coin element to check for fund availability. On the other hand, this new design provides seamless and faster transac-tions. In fact, just one message is sent from the ven-dor to the customer and another one is sent back from the customer to the vendor containing all the required digital coins, if available. All other mes-sages exchanged during the payment protocol will be managed internally inside the customer device. Furthermore, differently from our preliminary work , in FRoDO digital coins are directly com-puted in hardware by challenging the erasable PUF rather than being built in software. This avoids the usage of memories in the coin reconstruction pro-cess, thus mitigating any chance of attacks based on data vulnerabilities; Security properties. Differently from , the double-step communication protocol between the identity and the coin element allows, on the one hand, a bank/coin element issuer to design digital coins that can be read only by a certain identity element, i.e., by a specific user/device. This means that even though the coin element is lost or stolen by an attacker, such an element will not work without the associated identity element—hence providing a two-factor authentication for each transaction. On the other hand, the identity element can be used to thwart fraudsters. If an identity element is considered malicious and it is blacklisted, no matter which is the coin element used in the transaction, all payment requests will be rejected. Whilst in the physical unclonable function was used only to authenticate core elements of the architecture, in this improved version multiple physical unclonable functions are also used to allow all the elements to interact in a secure way.

BACKGROUND

Payment transactions are usually processed by an electronic payment system (for short, EPS). The EPS is a separate function from the typical point of sale function, although the EPS and the PoS system could be co-located on the same machine. In general, the EPS performs all payment process-ing, while the PoS system is the tool used by the cashier or consumer .

PoS System Breaches

Attacks against PoS systems in mature environments are typically multi-staged . First, the attacker must gain access to the victim's network (this step is called infiltration). Usually, they gain access to an associated network and not directly to the cardholder data environment. They must then traverse the network (this step is called propagation), ultimately gaining access to the PoS systems. Next, they install malicious software in order to steal data from the compromised systems (this step is called aggregation). As the PoS system is unlikely to have external network access, the stolen data is then typically sent to an internal back office server waiting for the attacker to be back (this step is called exfiltration). PoS system network-level hacking can be rendered possible by exploiting shared connections, open networks, or by cracking the password of the merchant's network. How-ever, networks can be monitored and protected against malicious activities [23].

Network infiltration is just one of the many sophisticated attack methods. In addition, a successful server breach will give attackers access not only to a single PoS system or to a network of PoS systems in a single location but, depending on the architecture, possibly to all PoS systems controlled by the retailer, even in multiple locations. Regardless of the adopted EPS model, the payment process is composed of two main processing phases, the authorization and the settlement.

verified and finalized. The settlement comprises all actions happening after the authorization stage. Even though the data processed at this stage is not as valuable as the data processed during the authorization stage, it still contains sensitive data such as the amount of money spent within the transaction. Such information is relevant to customer privacy and thus it has to be protected.

PoS Device Breaches

PoS devices can be considered the most important entities in an electronic payment system and are normally "guarded" by employees during operating hours. However, it is still possible for an attacker to inject malware into the PoS or even to replace it with a fake/malicious device. In fact, many all-in-one PoS systems are based on general purpose operating systems. As such, they are susceptible to a wide variety of attack scenarios which could lead to large scale data breaches. All the attacks described so far require the PoS to be connected to a network in order for the attacker to break into the payment system and infect either the PoS itself or a specific component within the EPS. However, as already introduced in Section 2, EPS can also be fully off-line. In this scenario, no data is going to leave the PoS and there is no way to infect the PoS. As such, breaches based on network-level hacking cannot be unleashed. However, data processed by the PoS can still be eavesdropped by having physical access to the PoS itself or by exploiting device vulnerabilities. In Section 4a description of the possible breaches threatening PoS systems will be provided.

PROPOSED METHOD

The solution proposed in this work, FRoDO, is based on strong physical unclonable functions but does not TABLE 2 Most Relevant Acronyms Used in This Paper Acronym Meaning APD Avalanche PhotoDiode API Application Programming Interface CRP Challenge-Response Pair EPS Electronic Payment System FIB Focused Ion Beam IC Integrated Circuit IPsec Internet Protocol Security PII Personal Identifiable Information PoS Point of Sale PUF Physical Unclonable Function SPP Simple Pairing Protocol SSL Secure Sockets Layer TLS Transport Layer Security TPM Trusted Platform Module TTP Trusted Third Party require any precomputed challenge-response pair . Physical unclonable functions (for short, PUFs) were introduced by Ravikanth 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. FRoDO is the first solution that neither requires trusted third parties, nor bank accounts, nor trusted devices to pro-vide resiliency against frauds based on data breaches in a fully off-line electronic payment systems. Furthermore, by allowing FRoDO customers to be free from having a bank account, makes it also particularly interesting as regards to privacy. In fact, digital coins used in FRoDO are just a digital version of real cash and, as such, they are not linked to anybody else than the holder of both the identity and the coin element.

Differently from other payment solutions based on tamperproof hardware, FRoDO assumes that only the chips built upon PUFs can take advantage from the tamper evidence feature. As a consequence, our assumptions are much less restrictive than other approaches. As depicted in Fig. 2, FRoDO can be applied to any scenario composed of a payer/customer device and a payee/ vendor device. All involved devices can be tweaked by an attacker and are considered untrusted except from a storage device, that we assume is kept physically secure by the vendor.

Furthermore, it is important to highlight that FRoDO has been designed to be a secure and reliable encapsulation scheme of digital coins. This makes FRoDO also applicable to multiple bank scenarios. Indeed, as for credit and debit cards where trusted third parties (for short, TTPs) such as card issuers guarantee the validity of the cards, some common standard convention can be used in FRoDO to make banks able to produce and sell their own coin element. Any bank will then be capable of verifying digital coins issued by other banks by requiring banks and vendors to agree on the same standard formats.

Fig2. FRoDO model

FRoDO does not require any special hardware component apart from the identity and the coin element that can be either plugged into the customer device or directly embedded into the device. Similarly to secure elements, both the identity and the coin element can be considered tamper-proof devices with a secure storage and execution environ-ment for sensitive data. Thus, as defined in the ISO7816-4 standard, both of them can be accessed via some APIs while maintaining the desired security and privacy level. Such soft-ware components (i.e., APIs) are not central to the security of our solution and can be easily and constantly updated. This renders infrastructure maintenance easier.

FRoDO: The Architecture

As depicted in Fig. 3, the architecture of FRoDO is composed of two main elements: an identity element and a coin element. The coin element, depicted in Fig. 4, can be any

hardware built upon a physical unclonable function (such as an SD card or a USB drive) and it is used to read digital coins in a trusted way. The identity element has to be embedded into the customer device (such as a secure element) and it is used to tie a specific coin element to a specific device. This new design provides a two factor authentication to the customer. In fact, the relationship between a coin element and an identity element prevents an attacker from stealing coin elements that belong to other users. A specific coin element can be read only by a specific identity element (i.e., by a specific device). Furthermore, this approach still provides anonymous transactions as each identity element is tied to a device and not to a user. The whole FRoDO architecture can be decomposed as follows:

Identity element:

Key Generator: used to compute on-the-fly the private key of the identity element;

- Cryptographic Element: used for symmetric and asymmetric cryptographic algorithms applied to data received in input and sent as output by the identity element;

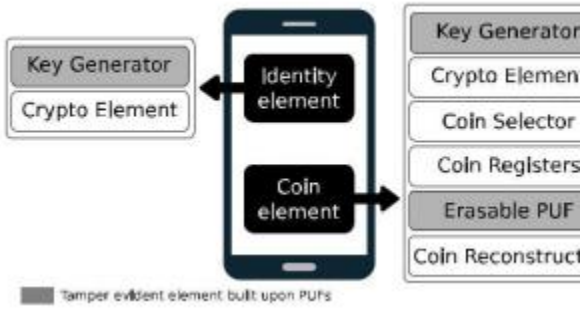

Fig 3.FRoDO main architecture

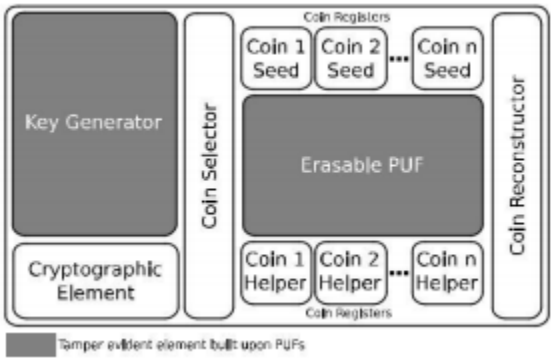

Fig 4. Coin element architecture

Coin Element.

- Key Generator: used to compute on-the-fly the private key of the coin element;

- Cryptographic Element: used for symmetric and asymmetric cryptographic algorithms applied to data received in input and send as output by the coin element;

- Coin Selector: is responsible for the selection of the right registers used together with the output value computed by the coin element PUF in order to obtain the final coin value;

- Coin Registers: used to store both PUF input and output values required to reconstruct original coin values. Coin registers contain coin seed and coin helper data. Coin seeds are used as input to the PUF whilst coin helpers are used in order to reconstruct stable coin values when the PUF is challenged;

- Erasable PUF[30]: is a read-once PUF . After the first challenge, even if the same input is used, the output will be random;

- Coin Reconstructor: responsible to use the out-put coming from the PUF together with a coin helper in order to reconstruct the original value of the coin. The reconstructor uses helper data stored into coin registers to extract the original output from the PUF. Both the identity element and the coin element are built upon physically unclonable functions. As such, both of them inherits the following features: Clone resiliency. It must be extremely hard to physically clone a strong PUF, i.e., to build another system which has the same challenge-response behavior as the original PUF. This restriction must hold even for the original manufacturer of the PUF; Emulation resiliency. Due to the very large number of possible challenges and the PUF's finite read-out rate, a complete measurement of all challenge-response pairs (for short, CRPs) within a limited time frame must be extremely hard to achieve; Unpredictability. It must be difficult to numerically predict the response of a strong PUF to a randomly selected challenge even if many other challenge-response pairs are known. In the remainder of this section, each element of FRoDO will be described. Furth the transaction protocol will be depicted.

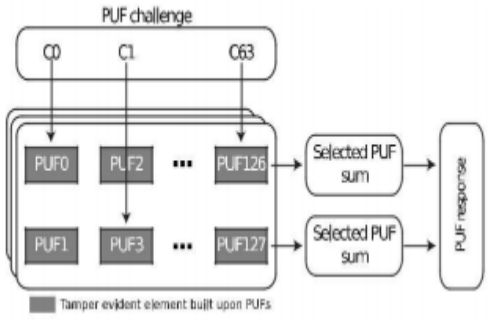

Fig 5. Stable PUF based private key generation

Key Generator

As depicted in Fig. 3, the key generator element is used both within the identity element and within the coin element. The main responsibility of such an element is to compute on-the-fly the private key. Such keys are used by the cryptographic elements to decrypt the requests and encrypt the replies. PUFs have been used in FRoDO to implement strong challenge-response authentication. In particular, multiple physical unclonable functions are used to authenticate both the identity element and the coin element and last, but not least, to allow them to interact in a secure way . In order to compute each private key, a publicly known ID (respectively the identity element ID and the coin element ID) is used as input to the PUF. Thus, both the identity and the coin element are shipped with such a hard-coded ID signed by the element issuer in order to avoid forgery attacks.

This allows the customer to broadcast the public key of both the identity and the coin element to vendors that are not required to know all the public keys of all the active identity/coin elements in the world. Furthermore vendors can encrypt payment requests with pub-lic keys of the customer's device identity element, thus ensuring that such requests will be read only by that customer. However, given a fixed input, PUFs can produce a response that is unique to the manufacturing instance of the PUF circuit but that is not bitwise-identical when reiterated multiple times. As such, in order to use PUFs in algorithms where stable

values are required, an intermediate step is needed. This problem is usually faced in cryptographic algorithms (known as "secret key extraction") . It can be solved using a two-steps algorithm. In the first step the PUF is challenged, 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-steps 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 and the cost of actual implementations thereof is assessed in.

While this approach is feasible for the coin element that is based upon an erasable PUF , this is not feasible for the identity element. In fact, storing PUF helper data within the device could allow an attacker to reconstruct the private key of the device. However, a number of solutions have been proposed to correct PUF out-put on-the-fly thus allowing the generation of stable secret values within the device, without the need of any helper data. FRoDO adopts a similar approach by using a light-weight error correction algorithm (see Fig. 7) to generate stable cryptographic keys from PUFs within both the identity element and the coin element. The basic 64-sum PUF block first introduced in [36] measures the difference between two delay terms, each produced by the sum of 64 PUF values. Then, given a challenge, its ith bit (called Ci) determines, for each of the 64 stages, which PUF is used to compute the top delay term, and which one is used to compute the bottom delay term. The sign bit of the difference between the two delay terms deter-mines whether the PUF outputs a 1 or a 0 bit-value for the 64-bit challenge C0 . . . C63. The remaining bits of the differ-ence 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 with a real-valued output that contains both the output bit as well as its confidence level. This information is then used by the downstream lightweight error correction block that is able to output a stable value. By using such on-the-fly stable value generation process, the identity/coin element private keys are not stored any-where within the customer device. Hence, they are much better protected from attackers trying to steal them .

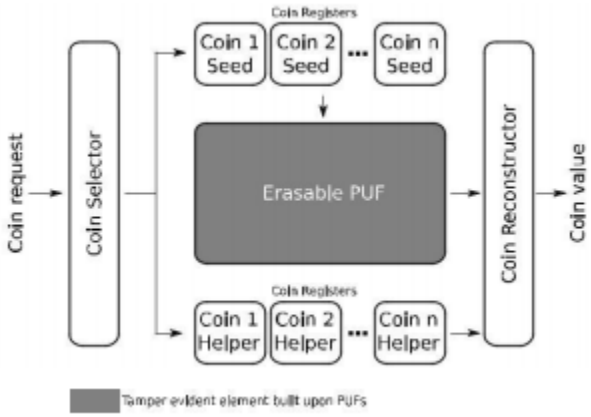

Fig 6. Coin reconstruction based on an erasable strong PUF

Erasable Coins

At the heart of FRoDO proposal lies a read-once strong physical unclonable function [30]. Such PUF, used to compute on-the-fly each coin, has the property that reading one value destroys the original content by changing the behavior of the PUF that will

response with random data in fur-ther challenges. FRoDO is not tied to any specific digital coin format. Furthermore, it does not directly write digital coins within the customer's coin element but uses special hardware to reconstruct them on-the-fly when needed. As depicted in Fig. 6

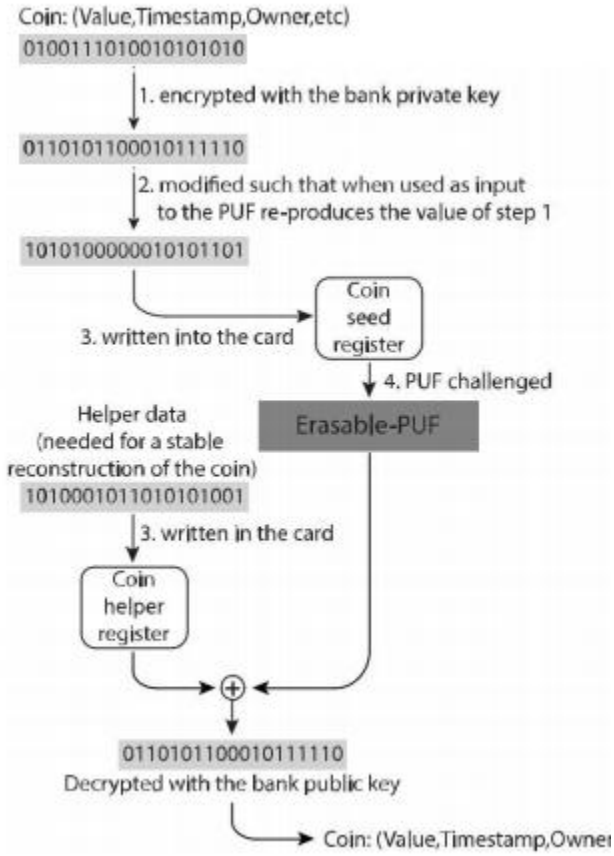

Fig 7. Coin reconstruction

vendor's coin requests do not contain the erasable-PUF challenge by themselves, but they are used as input to the coin selector. This latter one has information about avail-able funds for each register and it has the burden of selecting the coin registers (one or more) that will be involved in the transaction. The selected coin seed register is then used as input to the erasable PUF, while the coin helper register is combined to the PUF output in order to reconstruct the final value of the coin. The scheme of a coin reconstruction is given in Fig. 7. Coin raw data is first encrypted by the bank with its private key and then modified in order to create a chunk of bytes that are written into the coin seed register. Further, helper data are written in the coin helper register in order to provide stable PUF output [33]. The coin seed register is then used at transacion time to challenge the erasable PUF. The obtained response is combined with the coin helper register data in order to obtain the original encrypted coin again. Finally, as depicted in Fig. 9, the original coin data is computed using the public key of the bank. Last but not least, FRoDO does not rely on any specific number or type of coins. As such, it can work with coin elements of any size and with any number of coins.

FRoDO: The Protocol

This section describes the payment protocol being used in FRoDO. For completeness' sake, the Transaction Dispute and the Redemption phases will be introduced in this section, even though they are not part of the payment procedure (composed of the Pairing and of the Payment phases).

Payment Phase

For the sake of clarity and completeness, the FRoDO pay-ment protocol will be described from two different points of view. From the first one (depicted in Fig. 10 where by Enc (X,Y1; . . . ; Yn) we mean that data Y1 . . . Yn is encrypted using key X), messages exchanged between the vendor and the customer device will be described. Then, from the second one (depicted in Fig.

11), customer device internal messages exchanged between the identity element and the coin element will be described. The protocol depicted in Fig. 10 is composed of the following steps 1) The customer sends a purchase request to the vendor asking for some goods;

2) The vendor first creates a random salt value. Then, it encrypts the coin request three times. The first time with the salt itself. The second time with the public key of the identity element (i.e., the public key of the customer device that is going to receive this request), and the last time with the private key of the vendor itself. Thus, operations performed by the vendor are the following:

$$Enc_{Salt}(Req) = CReq$$

$$Enc_{IePK}(CReq; Salt) = EncReq$$

$$Enc_{VSK}(EncReq) = PrivateReq:$$

3) Once the private request has been built, it is sent to the customer;

4) When the customer receives such a request, first the private key of the identity element is computed by the identity element key generator. Then, all the encryption layers computed by the vendor are removed. As such, the customer computes three decryption operations. The first one with the private key of the identity element and the last one with the salt value.

$$Dec_{VPK}(PrivateReq) = EncReq$$

$$Dec_{IeSK}(EncReq) = (CReq; Salt)$$

$$Dec_{Salt}(CReq) = Req:$$

5) Once the coin request is in plain-text, the value of the coin is retrieved from the coin element. Then, such a value computed by the eras-able PUF and the coin reconstructor is first encrypted with the salt, then with the private key of the identity element (in order to prove the authenticity of the response) and at the end with the public key of the vendor—to ensure that only the right vendor device can decrypt it. That is:

$$Enc_{Salt}(CoinValue) = CValue$$

$$Enc_{IeSK}(CValue) = EncValue$$

$$Enc_{VPK}(EncValue) = PrivateResponse:$$

6) When the vendor finally receives the Private Response, the last step only requires the coin just read to be validated. Then, the whole payment transaction can be authorized and committed. Main steps are as follows: first, the received response is decrypted with the private key of the vendor. Second, the obtained value is decrypted with the public key of the identity element. Then, the salt is used to obtain the value read from the erasable PUF. As a final step, the public key of the bank/coin element issuer is used to decrypt the Coin Value that was encrypted (at manufacturing time) by the bank/ coin element issuer, with its private key. This way, it is possible to obtain and

verify the raw coin data built by the bank/card issuer.

$$\underline{\mathrm{Dec}}_{\mathrm{VPK}}(\mathrm{PrivateResponse}) = \mathrm{EncValue}$$
$$\underline{\mathrm{Dec}}_{\mathrm{IePK}}(\mathrm{EncValue}) = \mathrm{CValue}$$
$$\mathrm{Dec}_{\mathrm{Salt}}(\mathrm{CValue}) = \mathrm{CoinValue}$$
$$\underline{\mathrm{Dec}}_{\mathrm{BPK}}(\mathrm{CoinValue}) = \mathrm{RawValue}:$$

7) If the raw value of the just read coin is correct, a new entry is stored in the storage device of the vendor after being encrypted with the vendor's private key. It is important to stress that the Coin Value value is not a raw representation of the coin, but it is encrypted at manufacturing time by the bank with its private key. This means that it is not possible to forge digital coins. Indeed, the whole transaction will be validated if and only if the decryption of the Coin Value with the public key of the bank is successful.

Now that all messages exchanged between the customer and the vendor device have been introduced, it is possible to show how the identity and the coin elements interact:

1) Once the identity element has decrypted the coin request received by the vendor, it has to start a customer device internal protocol that allows the identity element to read a coin from the coin element. The first operation is the encryption of the coin request with the private key of the identity element. This provides authenticity for the message that will be received by the coin element. Then, such a private request (for short, PrReq) is encrypted with the pub-lic key of the coin element in order to mitigate Man in The Middle (for short, MITM) attacks between the identity element and the coin element:

Symbol	Meaning
Enc()/Dec()	Symmetric encryption/decryption
Enc()/Dec()	Asymmetric encryption/decryption
Salt	Salt value
IePK	Identity element public key
IeSK	Identity element secret (private) key
CePK	Coin element public key
CeSK	Coin element secret (private) key
BPK	Bank/Element Issuer public key
BSK	Bank/Element Issuers secret (private) key
VPK	Vendor public key
VSK	Vendor secret (private) key

Table1. Symbols Used in the Transaction Protocol

$$\underline{\mathrm{Enc}}_{\mathrm{IeSK}}(\mathrm{PrReq}) = \mathrm{PrReq}$$
$$\underline{\mathrm{Enc}}_{\mathrm{CePK}}(\mathrm{PrReq}) = \mathrm{SecureRequest}:$$

2) The newly encrypted coin request is sent to the coin element;

3) Once the coin request is received by the coin element, the first operation is the retrieval of the private key of the coin element. As for the identity element, the coin element uses its embedded ID as a challenge to the PUF that will response with the private key as output;

4) When the private key of the coin element has been computed, it is possible to first decrypt the request received by the identity element and then decrypt the obtained output using the public key of the identity element. This ensures message authenticity and integrity:

$$\text{Dec}_{CeSK}(\text{SecureRequest}) = PrReq$$

$$\text{Dec}_{IePK}(PrReq) = Req.$$

5) Such a request is then used to challenge the erasable PUF embedded into the coin element. All the involved operations are as follow :

$$\text{SelectCoinSeed}(Req) = PUFChallenge$$

$$\text{ReadCoin}(PUFChallenge) = PartialCoin$$

$$\text{Reconstruct}(PartialCoin, CoinHelper) = CoinValue.$$

6) The coin value has now to be encrypted twice. The first encryption layer is needed in order to prove the authenticity of the coin. The second encryption layer is needed such that only the right identity element will be able to read it:

$$\text{Enc}_{CeSK}(CoinValue) = EncCoin$$

$$\text{Enc}_{IePK}(EncCoin) = FinalCoin.$$

7) When the encrypted coin has been received by the identity element, these two encryption layers are removed:

$$\text{Dec}_{IeSK}(FinalCoin) = EncCoin$$

$$\text{Dec}_{CePK}(EncCoin) = CoinValue.$$

8) Now the identity element has the coin value read from the erasable PUF. In order to complete the transaction, this value will be sent back to the vendor device as shown in the previous description of the protocol .

If all the steps are accomplished without errors the transaction is authorized and the purchase is allowed. It is important to highlight that FRoDO 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 pro-posed solution.

SECURITY ANALYSIS

In this section the robustness of FRoDO is discussed. FRoDO uses both symmetric and asymmetric cryptographic primi-tives in order to guarantee the following security principles: Authenticity. It is guaranteed in FRoDO by the onthefly computation of private keys. In fact, both the identity and the coin element use the key generator to compute their private key needed to encrypt and decrypt all the messages exchanged in the protocol. Furthermore, each public key used by both the ven-dor and the identity/coin element is signed by the bank. As such, its authenticity can always be verified by the vendor; Non-repudiation.

The storage device that is kept physi-cally safe by the vendor prevents the adversary from being able to delete past transactions, thus protecting against malicious repudiation requests. Furthermore, the content of the storage device can be backed up and exported to a secondary equipment, such as pen drives, in order to make it even harder for an adversary to tamper with the transaction history; Integrity. It is ensured with the encryption of each dig-ital coin by the bank or identity/coin element issuer. Coin seeds and coin helpers are written into the coin element registers by either the bank or coin element issuer such that the final

coin value given as output corresponds to an encrypted version of the real digital coin. As such, by using the public key of the bank or identity/coin element issuer, it is always possible to verify the integrity of each coin. Furthermore, the integrity of each message exchanged in the protocol is provided as well. In fact, both the identity and the coin element use their private/public keys. The private key is not stored anywhere within the identity/ coin element but it is computed each time as needed; Confidentiality. Both the communications between the customer and the vendor and those between the identity element and the coin element leverage asymmetric encryption primitives to achieve message confidentiality; Availability the availability of the proposed solution is guaranteed mainly by the fully off-line scenario that completely removes any type of external communication requirement and makes it possible to use off-line digital coins also in extreme situations with no network coverage.

Furthermore, the lack of any registration or withdrawal phase, makes FRoDO able to be used by different devices. As in FRoDO shares the assumption that each element built on top of a PUF is tamper-evident. This assumption is based on the size of nowadays integrated circuits (for short, IC) and on the impossibility for a casual attacker to open the device without causing an alteration in PUF behavior. This assumption is no longer valid if an expert attacker 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 cost thousands of dollars and applying this kind of attack on each single device to steal a few dollars does not provide an incentive to attack the system.

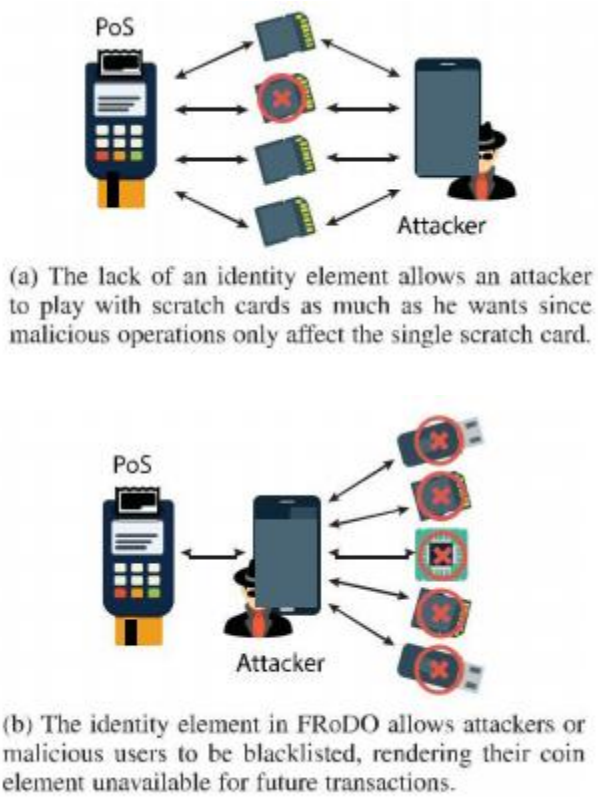

Fig 8. Attacks over the coin element

Blacklists

FRoDO uses two different elements: an identity element and a coin element, in order to improve the security of the whole payment system (see Fig. 12). In fact, the vendor device does not directly communicate with the coin element but has to go through the identity element. On the one hand this allows either the bank or the coin element issuer to design all the digital coins belong to a specific coin element to be read only by a certain identity element, i.e., by a specific user. This means that even though the coin element is lost or it is stolen

by an attacker, such element will not work without the associated identity element. As such, the identity element can be considered as a second factor aimed at improving the security of customer coins. On the other hand, the identity element can be used to fight against attackers as well. In fact, as depicted in Fig. 12, if an identity element is considered malicious and is black-listed, no matter what is the device used by the user, any coin will not be accepted and processed by the vendor.

Attack Mitigation

In this section, the resiliency of FRoDO to the attacks listed in Section 4 is discussed: Double spending. The read-once property of the eras-able PUF used in this solution prevents an attacker from computing the same coin twice. Even if a malicious customer creates a fake vendor device and reads all the coins, it will not be able to spend any of these coins due to the inability to decrypt the request of other vendors. Indeed, as described in Section 5.1, the private keys of both the identity and coin elements are needed to decrypt the request of the vendor and can be computed only within the customer device. The fake vendor could then try to forge a new emulated identity/coin element with private/ public key pair. However, identity/coin element public keys are valid only if signed by the bank. As such, any message received by an unconfirmed identity/coin element will be immediately rejected; Coin forgery. Each coin is encrypted by either the bank or the coin element issuer and thus it is not possible for an attacker to forge new coins; Emulation.

Physical unclonable functions, by design, can be neither dumped nor forged, either in hard-ware or software. Responses computed by emulated/fake PUFs will be different from the original ones; Postponed transaction. The only way to understand data obtained as output from the identity/coin element is by having access to their private key. How-ever, physically opening these elements will alter their PUFs behavior thus invalidating the elements itself. However, no information is kept within the elements, either in plaintext or in the encrypted form. As such, an attacker will not be able to steal any information; Information stealing. The private key of each element is computed on-the-fly as needed. No sensitive information is kept in either the identity or the coin element. Coin seeds and coin helpers do not provide by themselves any information about coins and physical access to the hardware will cause the PUFs to change their behavior as already described in Section 5.1; Replay. Each transaction, even if related to the same coin, is different due to the random salt generated each time by the vendor; Man in the middle. Digital coins are encrypted by either the bank or the coin element issuer and contain, among all other things, the ID of the coin element.

Furthermore, as in FRoDO digital coins are computed at run-time rather than being written into the memory, an attacker cannot dump coins from another customers. Last but not least, an attacker cannot pretend to be another customer with a different ID because it will not be able to compute his private key; Reverse engineering. By design, any attempt to tweak and steal any useful information from either the identity or the

coin element will alter the behavior of the PUFs thus rendering the elements no longer usable; Denial of services. FRoDO uses an initial pairing process. Such step cannot be accomplished by an attacker as it requires a security code to be manually typed on the customer's device. As such, DoS and DDoS attacks are mitigated. Even if the attacker is a malicious vendor, each transaction has to be con-firmed by the customer thus preventing batch attacks where either the identity or the coin element are repeatedly challenged; HW modification.

Physical Access Protection

As regards physical attacks to PUFs, integrated circuits and hardware in general, some relevant results are discussed. 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 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 crypto-graphic 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. As such, the problem of limiting data access in a physical device is extremely difficult. Attacks that try to infer infor-mation from a device can be categorized as passive or intru-sive attacks. In passive attacks the system interface is probed for either timing or electrical differences. In intrusive attacks the attacker is able to breach the physical boundary of the package and can scan, probe or alter the hardware itself. Passive attacks can be further subdivided into powered and un-powered attacks. In powered attacks the device is monitored while running whilst in unpowered attacks, information is extracted from the device while the hardware is not powered on. In FRoDO digital coins are directly computed in hard-ware by challenging the erasable PUF rather than being built in software. As such, no memory is involved in the reconstruction process. This mitigates the chances of un-powered attacks. Whereas, stealing information on-the-fly at run-time would require extremely expensive instrumentation.

CONCLUSION

In this paper we have introduced FRoDO that is, to the best of our knowledge, the first data-breach-resilient fully off-line micropayment approach. The security analysis shows that FRoDO does not impose trustworthiness assumptions. Further, FRoDO is also the first solution in the literature where no customer device data attacks can be exploited to compromise the system. This has been achieved mainly by leveraging a novel erasable PUF architecture

and a novel protocol design. Furthermore, our proposal has been thoroughly discussed and compared against the state of the art. Our analysis shows that FRoDO is the only proposal that enjoys all the properties required to a secure micro-payment solution, while also introducing flexibility when considering the payment medium (types of digital coins). Finally, some open issues have been identified that are left as future work. In particular, we are investigating the possibility to allow digital change to be spent over multiple off-line transactions while maintaining the same level of security and usability.

REFERENCES

[1] Near Field Communication Forum. http://www.nfc-forum.org/, 2008.

[2] M. Abe. A secure three-move blind signature scheme for polynomially many signatures. In B. Pfitzmann, editor, Advances in Cryptology — EUROCRYPT 2001, volume 2045 of Lecture Notes in Computer Science, pages 136–151. Springer Berlin Heidelberg, 2001.

[3] R. Abelson and M. Goldstein. Millions of anthem customers targeted in cyberattack. http://www.nytimes.com/2015/02/05/business/ hackers-breached-data-of-millions-insurer-says.html?_r=1, 2015.

[4] O. Acıı¸cmez, B. Brumley, and P. Grabher. New results on instruction cache attacks. In S. Mangard and F.-X. Standaert, editors, Cryptographic Hardware and Embedded Systems, CHES 2010, volume 6225 of Lecture Notes in Computer Science, pages 110–124. Springer Berlin Heidelberg, 2010.

[5] N. Asokan, P. A. Janson, M. Steiner, and M. Waidner. The state of the art in electronic payment systems. IEEE Computer, 30(9):28–35, 1997.

[6] F. Baldimtsi, M. Chase, G. Fuchsbauer, and M. Kohlweiss. Anonymous transferable e-cash. In J. Katz, editor, Public-Key Cryptography – PKC 2015, volume 9020 of Lecture Notes in Computer Science, pages 101–124. Springer Berlin Heidelberg, 2015.

[7] F. Baldimtsi and A. Lysyanskaya. Anonymous credentials light. In Proceedings of the 2013 ACM SIGSAC conference on Computer communications security, CCS ’13, pages 1087–1098, New York, NY, USA, 2013. ACM.

[8] P. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In B. Preneel and S. Tavares, editors, Selected Areas in Cryptography, volume 3897 of Lecture Notes in Computer Science, pages 319–331. Springer Berlin Heidelberg, 2006.

[9] L. Batina, J.-H. Hoepman, B. Jacobs, W. Mostowski, and P. Vullers. Developing efficient blinded attribute certificates on smart cards via pairings. In D. Gollmann, J.-L. Lanet, and J. Iguchi-Cartigny, editors, Smart Card Research and Advanced Application, volume 6035 of Lecture Notes in Computer Science, pages 209–222. Springer Berlin Heidelberg, 2010

[10] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. Compact e-cash and simulatable VRFs revisited. In H. Shacham and B. Waters, editors, PairingBased

Cryptography – Pairing 2009, volume 5671 of Lecture Notes in Computer Science, pages 114–131. Springer Berlin Heidelberg, 2009.

[11] D. Bernstein. Batch binary edwards. In S. Halevi, editor, Advances in Cryptology - CRYPTO 2009, volume 5677 of Lecture Notes in Computer Science, pages 317–336. Springer Berlin Heidelberg, 2009.

[12] D. Bernstein and P. Schwabe. NEON crypto. In E. Prouff and P. Schaumont, editors, Cryptographic Hardware and Embedded Systems – CHES 2012, volume 7428 of Lecture Notes in Computer Science, pages 320–339. Springer Berlin Heidelberg, 2012.

[13] D. J. Bernstein. Curve25519: new Diffie-Hellman speed records. In M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, editors, Public Key Cryptography – PKC 2006, volume 3958 of Lecture Notes in Computer Science, pages 207– 228. Springer-Verlag Berlin Heidelberg, 2006. http://cr.yp.to/papers.html# curve25519.

[14] D. J. Bernstein. Cryptography in NaCl. http://cr.yp.to/highspeed/ naclcrypto-20090310.pdf, 2009.

[15] D. J. Bernstein, B. van Gastel, W. Janssen, T. Lange, P. Schwabe, and S. Smetsers. TweetNaCl: A crypto library in 100 tweets, to appear. Document ID: c74b5bbf605ba02ad8d9e49f04aca9a2, http://cryptojedi.org/ papers/#tweetnacl.

[16] P. Bichsel, J. Camenisch, T. Groß, and V. Shoup. Anonymous credentials on a standard java card. In Proceedings of the 16th ACM Conference on Computer and Communications Security – ACM CCS 2009, CCS '09, pages 600–610, New York, NY, USA, 2009. ACM.

[17] E.-O. Blass, A. Kurmus, R. Molva, and T. Strufe. PSP: Private and secure payment with rfid. In Proceedings of the 8th ACM Workshop on Privacy in the Electronic Society, WPES '09, pages 51–60, New York, NY, USA, 2009. ACM.

[18] A. J. Blumberg. On locational privacy, and how to avoid losing it forever. 2009.

[19] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. Journal of Cryptology, 17(4):297–319, 2004.

[20] J. W. Bos, M. E. Kaihara, T. Kleinjung, A. K. Lenstra, and P. L. Montgomery. On the security of 1024-bit rsa and 160-bit elliptic curve cryptography. IACR Cryptology ePrint Archive, 2009:389, 2009. 134

21] S. Brands. Untraceable off-line cash in wallets with observers (extended abstract). In Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '93, pages 302–318, London, UK, UK, 1994. Springer-Verlag

. [22] J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In R. Cramer, editor, Advances in Cryptology – EUROCRYPT 2005, volume 3494 of Lecture Notes in Computer Science, pages 302–321. Springer Berlin Heidelberg, 2005.

[23] A. Chan, Y. Frankel, and Y. Tsiounis. Easy come — easy go divisible cash. In K.

Nyberg, editor, Advances in Cryptology — EUROCRYPT 1998, volume 1403 of Lecture Notes in Computer Science, pages 561–575. Springer Berlin Heidelberg, 1998.

[24] D. Chaum. Blind signatures for untraceable payments. In D. Chaum, R. Rivest, and A. Sherman, editors, Advances in Cryptology, pages 199–203. Springer US, 1983.

[25] D. Chaum. Achieving electronic privacy. Scientific American, pages 96–101, August 1992. [26] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In S. Goldwasser, editor, Advances in Cryptology — CRYPTO' 88, volume 403 of Lecture Notes in Computer Science, pages 319–327. Springer New York, 1990.

[27] E. Clemente-Cuervo, F. Rodr´ıguez-Henr´ıquez, D. O. Arroyo, and L. Ertaul. A PDA implementation of an off-line e-cash protocol. In Proceedings of the 2007 International Conference on Security & Management, SAM 2007, Las Vegas, Nevada, USA, June 25-28, 2007, pages 452–458, 2007.