FRIENDLY OBSERVERS EASE OFF-LINE E-CASH

Shouhuai Xu
Dept. of Computer Science, Fudan Uni., P. R. China
shxu@fudan.edu.cn

Moti Yung
CertCo, New York, NY, USA
moti@cs.columbia.edu

Gendu Zhang
Dept. of Computer Science, Fudan Uni., P. R. China
gdzhang@fudan.edu.cn

Abstract

Double (or over) spending seems rather difficult to absolutly prevent in both perfectly and conditionally anonymous e-cash systems. In the case of *on-line* checking there is the potential for efficiency bottlenecks, whereas in *off-line* cash, the prevention which is currently available depends on the "mere tamper-resistance" of the hardware. Here, we introduce the concept of a *friendly* observer, and based on it we propose a robustness mechanism, namely a *watermark for wallets* which enhances the *mere* tamper-resistance of a wallet. We also present two concrete e-cash schemes based on the original wallet architecture (i.e., wallet=observer+computer) and the substantially cheaper one (i.e., wallet=observer=computer) introduced in this paper, respectively. Using the friendly observer which is assumed to be a circuit which follows its instructions, and to be aware of the application semantics, we are assured that a not so powerful tampering adversary will be captured in real time, whereas a versatile attacker (e.g., a well funded organization) is still traceable after the fact.

Keywords: Smart cards, e-cash, tamper-resistance, observer, security, reliability.

1. INTRODUCTION

Numerous e-cash schemes have been proposed since the invention of the technical enabler, namely *blind signatures*. The schemes' intention is to emulate electronically the payer-anonymity property of paper money [C82]. A basic concern for both perfectly and conditionally anonymous e-cash systems is *double-spending* (over-spending) because everybody can easily "clone" an e-coin (which is a bit string with certain semantics). Though the after-the-fact *detection* approach entitles us (e.g., via the challenged payment in perfectly anonymous e-cash of [CFN88], or the conditional anonymity in [FTY96]) to reveal the identity of the double-spender, it still does not prevent potential financial losses [BGJY98]. Therefore, double-spending *prevention*, which can be traced back to the very birth of e-cash in [C82], seems to be an important issue still. Indeed, we have already had two candidates: *on-line* prevention (i.e., the freshness of an e-coin has to be checked by the bank before being accepted) and *off-line* prevention (i.e., based on the tamper-resistance of certain hardware). As far as *on-line* checking is concerned, it may impose an efficiency bottleneck. Even worse side effects like the so-called "reputation lost" [Ba98] for the bank is possible: the customers are unable to complete payments though they have enough fresh e-coins [GoT97], due to a denial-of-service vulnerability of all networks/servers (e.g., by industrial spies employed by some dishonest competitors). Such problems may still remain, even if the bank periodically broadcasts to the shops the list of e-coins having been spent already because perfect synchronization is impossible. Therefore, tamper-resistant hardware based solutions (i.e., *off-line* prevention) are still attractive candidates, though the techniques currently available are not satisfactory. In this paper we propose a new robustness mechanism to improve the *mere* tamper-resistance of electronic wallets via the incorporation of certain application semantics. The robustness is based on the observer (a small computer circuit which is in a well protected portion of a smart card) being faithful to its task, while its working environment can be somewhat tampered with. Based on it, we construct two practical e-cash schemes.

1.1. OUTLINE

Section 2 will give the background. In section 3 we present the model of our system including the new architecture (i.e., wallet= observer= computer, wallet= computer + observer). In section 4 we describe our basic e-cash proof schemes based on the two different architectures, respectively. While the two constructions are practical, the new architecture based solution may be preferred in practice since the cost of a wallet

is much cheaper (i.e., a decrease of at least 50%). In section 5 we extend the basic schemes to realize various preferred properties (e.g., scalability and control as in [XYZZ99b]).

2. BACKGROUND AND CONTRIBUTION

2.1. CATEGORIES OF TAMPER RESISTANCE

We adopt the taxonomy of attackers proposed by IBM to guide designers of security systems that rely to some extent on tamper resistance [AK97]:

1 Class I (*Clever Outsiders*): They are often very intelligent but may have insufficient knowledge of the system. They may have access to only moderately sophisticated equipment. They often try to take advantage of an existing weakness in the system, rather than try to create one.

2 Class II (*knowledgeable insiders*): They have substantial specialized technical education and experience. They have varying degrees of understanding of parts of the system but potential access to most of it. They often have highly sophisticated tools and instruments for analysis.

3 Class III (*funded organizations*): They are able to assemble teams of specialists with related and complementary skills backed by great funding resources. They are capable of in-depth analysis of the system, designing sophisticated attacks, and using the most advanced analysis tools. They may use Class II adversaries as part of the attack team.

In [AK97], the authors presented and developed certain intrusion techniques able to tamper with smart-cards and other tamper-resistant devices, which points out the vulnerability of existing technology with respect to Class II and even Class I attacks. Therefore, a motivating issue for this paper is making tamper-resistance devices immune to at least Class II adversaries.

2.2. E-CASH BASED ON "WALLETS WITH OBSERVERS"

The idea to prevent double-spending with tamper-resistant hardware was introduced by Even and Goldreich [EG84] with the intention to

bypass the efficiency bottleneck resulting from *on-line* freshness checking [C82]. However, in the context of [C82, EG84], it is obviously too risky to make the security and the robustness of the whole system rely on the tamper-resistance of the hardware only. If the hardware is ever compromised, the attacker can double-spend perfectly without getting captured. On the other hand, another implicit concern (i.e., not explicitly claimed before) against this approach is the potential threat to the privacy of the wallet holder, since the device is produced/initialized by the issuer or the bank. This is even more of a concern if the wallet is occasionally under the control of the bank.

The latter concern (i.e., potential threat to the privacy of the holder) was first addressed by Chaum and Pederson in [CP92] with the newly introduced wallet architecture called a "wallet with observer", in which a wallet consists of an observer issued and trusted by a special authority (e.g., the bank) and a computer on behalf of the customer. Cramer and Pederson [CP93] further enhanced the privacy of the implementation in [CP92] to be immune against the so-called *shared information* attack where even if a wallet is temporarily in the hands of the issuer (e.g., for maintenance), the privacy of the customer is still preserved. The best result, up to now, is due to Brands [B93] in which the dependence on the hardware tamper-resistance is weakened because the identity of a double-spender with a tampered wallet is still revokable.

Technically, the solutions in [CP92, CP93, B93] are based on the *mere* (as is) tamper-resistance of the wallets. In the wallet, blind signatures [C82] and its variants are adopted to prevent subliminal channels (in an information theoretic sense) which have the potential to corrupt the customers' privacy (via so-called *inflow* or *outflow*). Therefore, heavy computational overhead is imposed on the customer's *computer* as well as on the bank's *observer* in the payment and/or the withdrawal transactions.

2.3. OUR CONTRIBUTIONS

The contribution in regards to definitions of this paper is the concept of a *friendly* observer, which in turn entitles us to enhance the *as is* tamper-resistance of a wallet by incorporating certain application semantics. An observer is said to be *friendly* if it has no capability (in an information theoretic sense) to corrupt the privacy of the wallet holder (we trust its limited computing capabilities, which is reasonable for a small hardware circuit as opposed to a generic computer architecture. Thus, assuming friendliness on part of some of the circuit and using it as an integrity checking mechanism is the crux of the idea). There-

fore, we are able to introduce a new wallet architecture (however, the basic ideas may be of independent interest in any other application context which uses tamper-resistance hardware). The architecture consists of just the observer (i.e., wallet=observer=computer). This relatively simple idea is obviously cheaper than the original wallet [CP92, CP93, B93] where two sets of separated microprocessors and the corresponding memories have to be integrated into a single chip card (hereafter wallet=observer+computer). By combining the ideas borrowed from [XYZZ99b] and the robustness mechanism introduced in this paper, namely a *watermark for wallet*, we construct an e-cash system (actually, two concrete implementations based on the different wallet architectures). The main property of the schemes is that the double-spender will be captured on the spot if the adversary is not powerful enough, whereas the versatile attacker (e.g., funded organization or Class III) will also be revealed after the fact as in [B93]. This may consequently suggest that the retroactive detection mechanism can be viewed as an added favorable property of e-cash over physical money, since in the latter, a once-accepted forged bill will yield little traces of information to enable investigation of the violation.

It is notable that our robustness mechanism is implemented at the price of computing only 1 pre-computable Schnorr [S91] signature in a withdrawal and payment respectively and this signature was developed with the idea of making the operation simple to enable (future) smart cards to perform it (this can be further eased since discrete log based signatures can be calculated without on-line exponentiations). Also, the "money loss-tolerance" issue in [BBC+94] can be obviously dealt with (as in [XYZZ99b]) in our e-cash system, since the bank has a backup of all the e-coins which can be reclaimed as long as the customer does not lose his or her secret key.

Remarks: 1. This paper is motivated by making the "wallet with friendly observer" immune to at least to Class II adversaries. Well established models in the area are missing and are hard to formalize (see [SS99] regarding the difficulties). Therefore, what we claim to have realized in this paper, is claimed informally. The informal part is that an attack "changes in some way" the internal state of the machine but does not change the internals of the observer. This is a reasonable assumption even if the device may not have a "perfect self-zeroing" tamper resistance strategy. Class III attacks may be prevented if the potential benefits are not enough to balance the potential risks since the attackers can be traced after-the-fact due to the conditional anonymity of the

e-cash system (i.e., the identity of a double-spender is always revokable, which is shown formally).

2. Though we borrow many ideas from [XYZZ99b] for our e-cash constructions, no robustness mechanism like our *watermark for wallets* is proposed there. On the other hand, all of the preferred properties implemented in [XYZZ99b] (e.g., active detection against laundering and brute force) can be naturally employed in our e-cash system (see section 4).

3. The hard-coin implementation of [J97] is different from our solution since in [J97] the observer is still not *friendly*, as in [CP92, CP93, B93]

3. THE MODEL

The participants include the user/customer **U** or his computer **C** embedded into his wallet, (i.e., we do not separate them), the bank **B** (i.e., a representative of the transparently maintained proactive RSA [FGMY97] minting infrastructure as in [XYZZ99b]), the shop **S**, the certificate authority **CA** (i.e., a representative of the transparently maintained proactive [HJKY95, HJJKY97] magic-ink signature [JY97] based signing infrastructure as in [XYZZ99b]) responsible for issuing conditionally anonymous certificates to the customers, and the observer **O** responsible for *off-line double-spending prevention* (i.e., on behalf of the bank).

The bank **B** and the shop **S** have their own certificates (i.e., RSA [RSA78] public encryption keys with exponents being the value 3: $(3, N_B)$ and $(3, N_S)$ are certified by a normal certificate authority independent of the current paper) which are publicly known (including the observers) and verifiable (therefore, two multiplications are enough to perform a possibly hybrid encryption). Each observer **O** is equipped with a publicly verifiable native certificate, $Cert_O$, issued by the bank for the observer's signature verification key (as in [CP92, CP93, B93]). That is to say, the signature verification key certificate of the bank **B** is implicitly used and is assumed to be known to at least the shop **S** with whom the validity of a claimed observer certificate can be verified.

We also assume that the customers trust that there will never be a quorum of servers in the **CA** infrastructure that illegally conspire to reveal their anonymity, and that the bank will neither incriminate them nor steal their money (while this can be realized with technical measures, we believe that the "market laws" will do it better; on the other hand, this trust has already existed in the real world). Technically we assume that neither discrete log (i.e., DSS [NIST91] and Schnorr [S91]) nor RSA

[RSA78] signatures are existentially forgeable under the adaptive chosen message attack [GMR88].

What is most different in our system from those in [CP92, CP93, B93] is that the *friendly* observer **O** is allowed to communicate with the outside world (and is trusted not to abuse the protocol). Indeed, this is essential in our solution and the very reason why the ideas cannot be easily adapted to the previous solutions based on the *non-friendly* observer since attacks like the *Mafia* attack which the observer can mount are rather difficult to deal with [BC93].

3.1. THE ADVERSARY MODEL

We distinguish a strong attacker from a weak one. One who can tamper with a wallet without leaving traces (i.e., the tampered state of the wallet is always consistent with the watermark). The watermark itself may also be tampered with (i.e., the powerful adversary can reveal the secret key of the observer). Such an attacker (defined below) is called a *strong* adversary. Hopefully, such a strong adversary can at most be created by well funded organizations (i.e., Class III in the introduction). Another adversary who is able to tamper with the state of a wallet but unable us to keep the state of the wallet consistent with the watermark (therefore leaving traces) is said to be a *weak* one. As we will see, a *weak* adversary which conducts double-spending is always captured on the spot (i.e., prevented), whereas the identity of a *strong* adversary can only be traced after the fact of some double-spendings. This depends on the strategy of the system, namely, on how often the shop deposits the received e-coins (thereby double-spending is discovered and further measures can be taken).

3.2. THE WALLET ARCHITECTURES

We denote wallet=computer+observer for the original [CP92, CP93, B93] architecture of a "wallet with observer", i.e., a wallet consists of the customer's computer **C** and the corresponding observer **O**. We define a new wallet architecture in which a wallet logically still consists of the customer's computer **C** and the corresponding observer **O**, but physically consists of the *friendly* observer only, that is, wallet=computer=observer. Anyhow, the observers in both architectures in the current paper are *friendly*.

The only difference between our two e-cash schemes based on the different wallet architectures is whether the issuer (i.e., the bank) of the observer is able to spend the customer's money or not. Indeed, the trust model in the new architecture e-cash scheme is somewhat stronger be-

cause the bank knows the secret key of the observer as well as the e-coin structure corresponding to the customer. However, the new architecture may still be much more attractive in practice due to the sharply decreased wallet cost and the fact that the trust concern can be eased if the secret key corresponding to the observer's certificate $Cert_O$ is destroyed by the bank at initiation and is thus known only to the observer.

Remark: As we focus on the improved *off-line double-spending prevention* property resulting from the enhanced tamper-resistance of the wallet, other objectives of our e-cash schemes are omitted in the current version of this paper (refer to [XYZZ99b] for the details).

4. THE BASIC SCHEMES

In the basic schemes a customer has only one conditionally anonymous certificate, $Cert_C$, issued by the **CA** using the proactivized [HJKY95, HJJKY97] version of the magic-ink signature scheme in [JY97] to generate a DSS [NIST91] signature on his Schnorr [S91] signature verification key, and only one anonymous account at the bank. In order to focus on the newly introduced techniques, we assume that only one e-coin is associated with an anonymous account (i.e., a certificate), therefore, only one e-coin in the withdrawal and the payment protocols is involved (various extensions to realizing improved scalability and control are depicted in section 4). As in [XYZZ99b], we adopt the technique in [JY97] to realize conditionally anonymous certificates just for the sake of a case study, whereas many other building blocks can be used instead.

Notations. In this paper, $ENC_X(\cdot)$ denotes the encryption function under the public key of participant X, and $SIG_Y(\cdot)$ denotes the signing function under the private key of participant Y. All of the moduli are omitted when they are clear from context. Therefore, $Cert_C = SIG_{CA}(h^{x_C})$ is the certificate for the customer's conditionally anonymous signature verification key h^{x_C}, $Cert_O = SIG_B(g^{x_O})$ is the certificate signed by the bank for the observer's signature verification key g^{x_O}. Both are well-defined (thus omitted) in the discrete log context (refer to, say, [JY97, S91]). When x and y are two strings, (x, y) denotes their concatenation.

4.1. THE BASIC IDEA

The customer first obtains a conditionally anonymous certificate $Cert_C$ from the **CA**, and then presents it with ownership authentication to the bank to setup an anonymous account A_U (as in [XYZZ99b]) and to

obtain an observer **O**. Therefore, the corresponding observer certificate $Cert_O$ is associated with the anonymous account A_U (therefore $Cert_C$). Having done this, the customer can deposit either physical money or some unconditionally anonymous e-check (e.g., [C82]) at his anonymous account A_U, from which he can withdraw an e-coin with the indirect account ownership authentication via the associated $Cert_O$. Obviously, while the *friendly* observer is able to leak no information about the identity of the wallet holder by nature, the holder can always reclaim his money at the corresponding anonymous account via the ownership authentication of anonymous certificate if his wallet is lost.

For the sake of clarity, we here assume that there has already been an appropriate balance at the customer's anonymous account A_U, thus we need only to focus on the withdrawal and the payment protocols. Also, the deposit protocol is omitted in this paper due to its triviality.

4.2. PRELIMINARY

A valid e-coin is a pair $(x, f^{1/3}(x) \ mod \ N_d)$ where $(3, N_d)$ is the bank **B**'s signature verification key for denomination d, where x is the serial number of the coin, and f is a suitable one-way function (e.g., collision-free) as in [C82] and $f^{1/3}$ denotes the signature on x which is hashed and signed with the RSA decryption exponent. Every e-coin in the wallet thus has a basic data structure, like $(x, f^{1/3}(x), balance)$, where *balance* is the available value. For example, a totally fresh coin of denomination \$10 means $balance = 10$; from now on it will be debited according to the payments. Thus, after paying \$2 and \$4 in sequence, the balance is \$4. Therefore, $balance = 0$ means that the e-coin cannot be used for purchases anymore.

Definition 4.1 *(watermark for wallets) The watermark for a wallet, denoted byw, is a Schnorr [S91] signature by the observer on the current state of the wallet. Specifically, let the watermark be* $w \stackrel{def}{=} (c, d)$*, such that* $c \stackrel{def}{=} H_1(H_2(x, balance), g^k)$*,* $d \stackrel{def}{=} c \cdot x_O + k$*, where* x *is the serial number of the e-coin,* k *is randomly chosen from the exponentiation group* Z_g *of* g *(i.e.,* $k \in_R Z_g$*), both* H_1 *and* H_2 *are ideal hash functions [BR93] whereas the range of* H_1 *is within* Z_g*.*

Remarks: 1. A watermark itself cannot be verified without the intended message because seeing just the pair (c, d), it is still computationally infeasible to find m such that $c = H_1(m, g^d(g^{x_O})^{-c})$. On the other hand, verifying a watermark is easy knowing $H_2(x, balance)$ and nothing about x and the *balance*. It is trivial to see that such an exten-

sion will lose no security from the original Schnorr signature scheme in the Random Oracle model.

2. According to [CHTY96, XYZZ99a], an e-coin can only be deleted from the wallet after the intended fair-exchange (e.g., exchange e-coin for e-goods) has been acknowledged by the customer in case there are Byzantine faults (i.e., both participants may be dishonest). Therefore, the above basic data structure for e-coins in the wallet needs to be extended to implement money conservation. Such an extension is omitted in the current paper.

4.3. SCHEME 1: WALLET=COMPUTER+OBSERVER

In this architecture, the wallet consists of the computer with the customer's anonymous certificate $Cert_C$, and the observer with a native certificate $Cert_O$. Each has its own microprocessor and memory as in [CP92, CP93, B93]. While they are correlated via the anonymous account A_U, $Cert_C$ is directly used in signing a payment, whereas $Cert_O$ is explicitly used in the withdrawal protocol.

4.3.1 Withdrawal. Protocol 1. The user/customer withdraws an e-coin from his anonymous account A_U.

1 The user **U** (i.e., **C**) claims to the bank **B** that he is the owner of A_U.

2 **B** asks the observer **O** to authorize a minting request.

3 **O** chooses randomly a pair (r, x), and supplies the semantically secure encryption $ENC_B(r, x, d)$ and the standard Schnorr [S91] signature $SIG_O(r, x, d)$ to **B** *directly*, where d is the denomination of the intended coin.

4 After checking the semantics of the decryption and the signature (including the fact that the $Cert_O$ is associated with the claimed A_U), **B** returns **O** *directly* with the threshold-wise (i.e., by its internal minting servers) signed $rf^{1/3}(x)\ mod\ N_d$, debits the A_U by d, and records the e-coin x at this account.

5 The observer **O** extracts $f^{1/3}(x)\ mod\ N_d$, and verifies its validity. The fresh e-coin $(x,\ f^{1/3}(x)\ mod\ N_d,\ balance = d)$ is kept by the observer, who then computes and stores the watermark of the current state $w \stackrel{def}{=} (c, d)$ by: choosing $k \in_R Z_g$, then calculating $c \stackrel{def}{=} H_1(H_2(x, balance), g^k)$, and $d \stackrel{def}{=} c \cdot x_O + k$. **O** will let **C** know $f^{1/3}(x)$ but not x.

Remark: In the withdrawal protocol, the rational for the customer to use a seemingly redundant r is so that the message $rf^{1/3}(x)\ mod\ N_d$ need not to be encrypted.

4.3.2 Payment. Protocol 2. The user **U** (i.e., **C**) and the shop **S** engage in such a protocol through some untraceable communication channel.

1 **U** and **S** negotiate through some interaction, and agree on the *price*, the goods description *desc*, and invalidate/time for this transaction $\tau = \tau_0 + \Delta$ where τ_0 is the time that the interested transaction begins and Δ is the negotiated lifetime for it. Alternatively, τ is the deadline for **S** to deposit the received e-coin (refer to [XYZZ99a]).

2 If $balance \geq price$, **O** sends the watermark (c, d) computed after the last payment (or, the withdrawal at the first time) transaction and the newly calculated hash of the *current* state of the wallet $s_{new} \stackrel{def}{=} H_2(x, balance)$ to **S** *directly*. If the watermark w is inconsistent with the new state, i.e., $c \neq H_1(s_{new}, g^d(g^{x_O})^{-c})$, it warns that the wallet has been tampered with and further measures can be taken. Otherwise, **S** requests **O** to send the intended money.

3 **O** optimistically and *directly* sends the encrypted version of the e-coin, $ENC_S((x, f^{1/3}(x)))$, and **C** presents the Schnorr signature

$$SIG_C(f^{1/3}(x), price, desc, \tau, I_S)$$

where I_S is **S**'s account, to **S**. Then, **O** calculates and stores $balance' \stackrel{def}{=} balance - price$, and the new watermark $w' \stackrel{def}{=} (c', d')$ by: choosing a new $k' \in_R Z_g$, then calculating:
$c' \stackrel{def}{=} H_1(H_2(x, balance'), g^{k'})$, and $d' \stackrel{def}{=} c' \cdot x_O + k'$.

4 **S** verifies the validity of the e-coin (i.e., issued by the bank), the validity of the signature (i.e., payment certified by a customer of $Cert_C$), and the semantics of the signed information. If it is okay, **S** delivers the goods to **U**.

4.4. SCHEME 2: WALLET=COMPUTER=OBSERVER

In this architecture, the wallet is just the observer with a native certificate $Cert_O$, therefore, there is only one microprocessor as well as the corresponding memory. The $Cert_O$ is explicitly used in both the withdrawal and the payment, whereas the $Cert_C$ is not explicitly used.

4.4.1 Withdrawal. Protocol 3. The user/customer withdraws an e-coin from his anonymous account A_U.

1 The observer **O** (i.e., the customer **C**) claims to the bank **B** that he is the owner of A_U.

2 **B** asks the observer **O** to authorize a minting request.

3 **O** chooses randomly a pair (r, x), and supplies the semantically secure encryption $ENC_B(r, x, d)$ and the standard Schnorr [S91] signature $SIG_O(r, x, d)$ to **B** *directly*, where d is the denomination of the intended coin.

4 After checking the semantics of the decryption and the signature (including the fact that the $Cert_O$ is associated with the claimed A_U), **B** returns **O** *directly* with the threshold-wise (i.e., by its internal minting servers) signed $rf^{1/3}(x) \ mod \ N_d$, debits the A_U by d, and records the e-coin x in this account.

5 The observer **O** extracts $f^{1/3}(x) \ mod \ N_d$, and verifies its validity. The fresh e-coin $(x, \ f^{1/3}(x) \ mod \ N_d, \ balance = d)$ is kept by the observer, who then computes and stores the watermark of the current state $w \stackrel{def}{=} (c, d)$ by: choosing $k \in_R Z_g$, then calculating $c \stackrel{def}{=} H_1(H_2(x, balance), g^k)$, and $d \stackrel{def}{=} c \cdot x_O + k$. **O** will let **C** know $f^{1/3}(x)$ but not x.

Remark: In the withdrawal protocol, the rational for the customer to use a seemingly redundant r is that the message $rf^{1/3}(x) \ mod \ N_d$ needs not to be encrypted.

4.4.2 Payment. Protocol 4. The observer **O** (i.e., the customer **U** or **C**) and the shop **S** engage in such a protocol through some untraceable communication channel.

1 **U** and **S** negotiate through some interaction, and agree on the *price*, the goods description *desc*, and invalidate/time for this transaction $\tau = \tau_0 + \Delta$ where τ_0 is the time that the interested transaction begins and Δ is the negotiated lifetime for it. Alternatively, τ is the deadline for **S** to deposit the received e-coin (refer to [XYZZ99a]).

2 If $balance \geq price$, **O** sends the watermark (c, d) computed after the last payment (or, the withdrawal at the first time) transaction and the newly calculated hash of the *current* state of the wallet,

$s_{new} \stackrel{def}{=} H_2(x, balance)$ to **S** *directly*. If the watermark w is inconsistent with the new state, i.e., $c \neq H_1(s_{new}, g^d(g^{x_O})^{-c})$, it warns that the wallet has been tampered and further measures can be taken. Otherwise, **S** requests **O** to send the intended money.

3 **O** optimistically and *directly* sends the encrypted version of the e-coin, $ENC_S((x, f^{1/3}(x)))$, and the Schnorr signature

$$SIG_O(f^{1/3}(x), price, desc, \tau, I_S)$$

where I_S is **S**'s account, to **S**. Then, **O** calculates and stores $balance' \stackrel{def}{=} balance - price$, and the new watermark $w' \stackrel{def}{=} (c', d')$ by: choosing a new $k' \in_R Z_g$, then calculating:
$c' \stackrel{def}{=} H_1(H_2(x, balance'), g^{k'})$, and $d' \stackrel{def}{=} c' \cdot x_O + k'$.

4 **S** verifies the validity of the e-coin (i.e., issued by the bank), the validity of the signature (i.e., payment certified by an observer of $Cert_O$), and the semantics of the signed information. If OK, **S** delivers the goods to **U**.

Remark: As $Cert_C$ is not directly used, there exists a risk that the bank can pay some purchases at the cost of the customer. While this can be prevented in our trust model, the bank might have to destroy the private key corresponding to $Cert_O$ in order to prevent certain legal risks (e.g., a bank robber may be able to obtain the corresponding secret key as well as the coin construction).

4.5. PROPERTIES OF THE SCHEMES

As we intended to realize improved *off-line double-spending prevention* with the enhanced tamper-resistance for the wallet, we omitted the claims for the other properties of the e-cash schemes which are rather similar to [XYZZ99b].

Claim 1 *In the scheme of wallet=observer+computer, the improved off-line double-spending prevention can only be corrupted by a strong adversary, who is still traceable from the double-spender due to the conditional anonymity of the e-cash.*

proof: (sketch) The robustness mechanism of *watermark for wallets* can only be corrupted by a *strong* adversary (e.g., funded organization), which is the same as the case in the real world whenever an attacker can forge paper money which is able to pass a bill-checking-machine.

However, even if there is such a *strong* adversary, the double-spender is still caught in the deposit protocol where further measures can be taken.

Similarly, we have the following theorem by assuming a somewhat stronger trust model as mentioned above.

Claim 2 *In the scheme of wallet=observer=computer, the improved off-line double-spending prevention can only be corrupted by a strong adversary, who is still traceable from the double-spender due to the conditional anonymity of the e-cash.*

5. EXTENSIONS

Let n denote the number of anonymous certificates/accounts one is allowed to hold, m the number of e-coins each anonymous account/certificate is allowed to withdraw (therefore to spend). In the former two basic schemes, we have $n = m = 1$. An extreme extension is to require that $n = 1$ and $m \geq 1$, therefore, all the payment transactions are anonymously associated with the same account because each customer is allowed to hold only one anonymous certificate/account. In this extension, the best control (i.e., a mechanism can be equipped to the law enforcers to *actively* detect attacks/abuses including laundering and brute force) and scalability are traded for so-called "best-effort (i.e., linkable) anonymity" [XYZZ99b]. Another more moderate extension is the setting of $n \geq 1$ and $m \geq 1$, therefore, all of the payment transactions associated with the same account are linkable, whereas the transactions associated with different accounts are unlinkable. This gives the flexibility to balance between control, privacy, and scalability to the law makers (opening more accounts reduces the potential of tracing anonymous acount limits).

In any case, we should adapt a little the protocols in the basic schemes. For example, the customer can easily obtain v coins by presenting

$$ENC_B(r_1, x_1, d_1 \cdots, r_v, x_v, d_v)$$

and

$$SIG_O(r_1, x_1, d_1 \cdots, r_v, x_v, d_v)$$

to the bank in the withdrawal protocol, and the corresponding watermark $w \stackrel{def}{=} (c, d)$ can be calculated by: choosing $k \in_R Z_g$, then calculating $c \stackrel{def}{=} H_1(H_2(x_1, balance_1, \cdots x_v, balance_v), g^k)$ and $d \stackrel{def}{=} c \cdot x_O + k$. The calculation of the watermark in a payment protocol can be similarly adapted to the multiple coins context.

6. CONCLUSION AND DISCUSSIONS

By introducing the concept of a *friendly* observer, we propose a robustness mechanism, namely a *watermark for wallet*, to enhance the tamper-resistance of a wallet. The protection is based on the adversary modifying the state outside of the observer (thus, changing the state which will not match the state watermark). We also presented two concrete e-cash schemes based on the original wallet architecture. We note that our robustness mechanism is implemented at the price of computing only 1 pre-computable modular exponentiation.

References

[AK97] R. Anderson and M. Kuhn, Low Cost Attacks on Tamper Resistant Devices, Security Protocol, 1997

[B93] S. Brands, Untraceable Off-Line Cash in Wallets with Observers, Crypto'93

[BCBS-98] Basle Committee on Banking Supervision, Risk Management for Electronic Banking and Electronic Money Activities, 1998

[BIS-97] Bank for International Settlements, Consumer Protection, Law Enforcement, Supervisory and Cross Border Issues, http://www.bis.org, 1997

[BBC+94] J. Boly, A. Bosselaers, R. Cramer et al., The ESPRIT Project CAFE: Highly Security Digital Payment Systems, ESORICS'94

[BC93] S. Brands and D. Chaum, Distance-Bounding Protocols, Eurocrypt'93

[BGJY98] M. Bellare, J. Garay, C. Jutla, and M. Yung, VarietyCash: A Multi-Purpose Electronic Payment System (Extended Abstract), Usenix Workshop on Electronic Commerce'98

[BR93] M.Bellare and P. Rogaway, Random Oracles are Practical: A Paradigm for Designing Efficient Protocols, ACM CCS'93

[C82] D. Chaum, Blind Signatures for Untraceable Payments, Crypto'82

[CFN88] D. Chaum, A. Fiat, and M. Naor, Untraceable Electronic Cash, Crypto'88

[CHTY96] J. Camp, M. Harkavy, J. D. Tygar, and B. Yee, Anonymous Atomic Transactions, 2nd Usenix on Electronic Commerce, 1996

[CP92] D. Chaum and T. P. Pederson, Wallet Database with Observers, Crypto'92

[CP93] R. Cramer and T. P. Pederson, Improved Privacy in Wallets with Observers, Eurocrypt'93

[EG83] S. Even and O. Goldreich, Electronic Wallet, Crypto'83

[FGMY97] Y. Frankel, P. Gemmel, P. D. MacKenzie, and M. Yung, Optimal-Resilience Proactive Public-Key Cryptosystems, FOCS'97

[FTY96] Y. Frankel, Y. Tsiounis, and M. Yung, Indirect Discourse Proofs: Achieving Fair Off-Line E-Cash, Asiacrypt'96

[HJKY95] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, Proactive Secret Sharing, or: How to Cope with Perpetual Leakage, Crypto'95

[HJJKY97] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung, Proactive Public Key and Signature Systems, ACM CCS'97

[J97] M. Jakobsson, Privacy vs. Authentication, PhD Thesis, 1997

[JY96] M. Jakobsson and M. Yung, Revokable and Versatile E-Money, 3rd ACM Computer and Communication Security, 1996

[JY97] M. Jakobsson and M. Yung, Distributed "Magic Ink" Signature, Eurocrypt'97

[NIST91] National Institute for Standards and Technology, Digital Signature Standards, 1991

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystem, Communication of the ACM, Vol. 21, No. 2, 1978

[S91] C. P. Schnorr, Efficient Signature Generation by Smart Cards, J. Cryptology, 4(3), 1991, 161-174

[SET97] Mastercard, Visa: Secure Electronic Transactions Version 1.0, 1997

[SS99] B. Schneier and A. Shostack, Breaking up Is Hard to Do: Modeling Security Threats for Smart Cards, 1999

[vSN92] B. von Solms and D. Naccache, On Blind Signatures and Perfect Crimes, Computers & Security, 11(6), 1992, 581-583

[XYZZ99a] S. Xu, M. Yung, G. Zhang, and H. Zhu, Money Conservation via Atomicity in Fair Off-Line E-Cash, ISW'99

[XYZZ99b] S. Xu, M. Yung, G. Zhang, and H. Zhu, Anonymous Payment System with Active Attacks/Abuses Detections, manuscript, 1999