TOMIN: Trustworthy Mobile Cash with Expiration-date Attached

Rafael Martínez-Peláez and Francisco Rico-Novella
Technical University of Catalonia/Department of Telematics Engineering, Barcelona, Spain
Email: {rafaelm, f.rico}@entel.upc.edu

Cristina Satizábal
Universidad Autónoma de Occidente/Departamento de Automática y Electrónica, Cali, Colombia
Email: icsatizabal@uao.edu.co

Abstract—**Online mobile payment systems based on mobile cash provide privacy to customers and are feasible for real point of sale, virtual point of sale, and person-to-person mobile commerce scenarios. The merchant does not perform complex operations and the bank verifies the validity of the mobile cash before the merchant delivers the product. The bank must store the mobile cash spent in a database to prevent a double spending attack. In this paper, we propose an efficient mobile cash scheme in which the customer attaches the expiration date and deposit date. This property reduces the size of the bank's database and the customer must spend the mobile cash before expiry. Moreover, the customer attaches the merchant's identity into the mobile cash in the deposit phase. The scheme requires low computational cost and is suitable for mobile devices**

Index Terms—**electronic cash, micro-payment, mobile commerce, one-way hash function, secure communication**

I. Introduction

Due to the fast progress of mobile commerce, several mobile payment systems [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] have been proposed. Some of theses systems are based on mobile cash[1] (m-cash). The use of m-cash provides privacy to customers and it is suitable for micro-payments.

In classic m-cash and e-cash systems [6], [8], [11], [12] proposed in the literature, the cash is built using hash chains [13]. A hash chain is used to create a chain h_0 = H(*seed*), h_1 = H(h_0), h_2 = H(h_1), …, h_i = H(h_{i-1}), where H(·) is a one-way hash function [14], [15], *seed* is a secret value, h_i denotes the composition of h with itself for i times, and h_1 = H(*seed*) = *seed*. The length of this chain l_i is defined as the number of function compositions required to obtain the highest value of the m-cash.

A typical online mobile payment system contains three types of participants (a customer, a merchant, and a bank) and four phases (initializing, withdrawing, unblinding, and depositing). In the initialization phase, the bank generates the key pair and publishes the public key (e, p) and a one-way hash function H(·). In the withdrawal phase, the customer withdraws her m-cash in a blinded version from the bank. Then, she unblinds her blinded m-cash to obtain valid cash. In the deposit phase, the customer sends the m-cash to the merchant and the merchant forwards the m-cash to the bank. The bank verifies whether or not the m-cash is fresh. Finally, the bank deposits the funds in the merchant's bank account.

In the last few years, new e-cash schemes with date-attached have been introduced in [16], [17], [18]. The use of expiration-date contributes to preventing the bank's database from growing uncontrollably [16]. Moreover, the use of the date in the m-cash gives the option of calculating the interest on the m-cash [17], [18]. Although the protocol of [17] is more efficient than [18] in storage load and computational cost, the customer and the merchant must perform many asymmetric operations (modular exponentiation computations) to process the electronic cash and to verify the electronic cash, respectively. The use of asymmetric operations (e.g. RSA algorithm) requires high energy- and time-consuming, and therefore these schemes are not efficient or suitable for mobile devices.

In this paper, we propose a new m-cash scheme called TOMIN (TruswOrthy Mobile cash with expIration date-attacheNt). In TOMIN, the customer first attaches the expiration date and then withdraws blind m-cash from the bank. The customer uses a value v pre-defined by the bank, so that the bank can verify partial information stored in the blinded message and sign it. The scheme provides privacy to customers in the withdrawal step. After the customer obtains her m-cash, she has a short time within which to use it. Moreover, the customer attaches the deposit date and merchant's identity into the mobile cash using a concatenation operation and a one-way hash function. This characteristic prevents any eavesdropper from trying to deposit the m-cash into her bank account.

The rest of this paper is organized as follows. In Section 2, we review related works. The proposed scheme is described in Section 3. In Section 4, we analyse the performance, security and storage of the proposal, and make comparison with other schemes.

[1] Mobile cash is an extension of electronic cash (e-cash) for mobile devices. In this paper we refer to m-cash instead of e-cash.

doi:10.4304/jsw.5.6.579-584

Finally, concluding remarks of this paper are given in Section 5.

II. PRELIMINARY

A. Untraceable electronic cash based on blind signature scheme

Chaum proposed a blind signature scheme and then applies this scheme to design untraceable electronic cash [19] which provides privacy to customers. There are three participants: a customer, a merchant, and a bank. The scheme consists of the following phases:

Initialization phase. The bank generates its RSA keys. The bank randomly selects two distinct large primes p and q, and then computes $n = pq$ and $\varphi = (p - 1)(q - 1)$. The bank chooses a large integer e at random, where $1 < e < \varphi$ and $GCD(e, \varphi) = 1$, and computes an integer d with $1 < d < \varphi$, such that $ed \equiv 1 \pmod{\varphi}$. Finally, it keeps (d, p, q) secrets and (e, n) public. The bank publishes a one-way hash function H(·) and (e, n).

Withdrawal phase. The customer randomly selects a number r and computes $\alpha = H(m)r^e \pmod{n}$, where m is the amount to withdraw. Then, the customer sends (α) to the bank. The bank computes $\beta = \alpha^d \pmod{n}$ and deducts w funds (e.g. dinar, euro, etc.) from the customer's account. Finally, the bank sends (β) to the customer.

Unblind phase. The costumer computes $s = \beta r^{-1} \pmod{n}$ to obtain (m, s), where (m, s) is the electronic cash for w dinar.

Deposit phase. The customer sends (m, s) to the merchant for a payment of x euros. The merchant verifies the electronic cash computing $s^e \equiv H(m) \pmod{n}$. If the verification process is correct, the merchant forwards (m, s) to the bank. The bank checks whether or not the electronic cash is fresh. Then, the bank will deposit x funds into the merchant's account and keep (m, s) in its database.

B. Untraceable electronic cash based on partial blind signature scheme

Abe and Fujisaki introduced the concept of a partial blind signature scheme [16], based on Chaum's blind signature scheme, in which the bank can verify some term of validity in the signing message. Then, they applied this scheme to date attached electronic cash to prevent the bank's database from growing uncontrolled. There are three participants: a customer, a merchant and a bank. The main difference with Chaum's scheme is that the customer and the bank negotiate and agree the format of constant v which it is used to control the expiration date of the electronic cash. The constant v contains the amount to withdraw w funds (e.g. dinar, euro, etc.) and the expiration date δ. The scheme consists of the following phases:

Initialization phase. The bank generates its RSA keys. The bank randomly selects two distinct large primes p and q, and computes $n = pq$ and $\varphi = (p - 1)(q - 1)$. The bank chooses a large integer e at random, where $1 < e < \varphi$ and $GCD(e, \varphi) = 1$, and computes an integer d with $1 < d < \varphi$, such that $ed \equiv 1 \pmod{\varphi}$. Finally, it keeps (d, p, q) secrets and (e, n) public. The bank publishes a one-way hash function H(·) and (e, n).

Withdrawal phase. The customer defines $v = w \| \delta$, where $\|$ indicates concatenation. Then, the customer randomly selects two integers m and r, and computes $\alpha = H(m)r^{ev} \pmod{n}$. Then, the customer sends (α, v) to the bank. The bank verifies that v is in correct format. If v is correct, the bank computes $\beta = \alpha^{dv} \pmod{n}$, where $d_v = (ev)^{-1} \pmod{\varphi}$. The bank deducts w funds (e.g. dinar, euro, etc.) from the customer's account. Finally, the bank sends (β) to the customer.

Unblind phase. After the customer receives β, she computes $s = \beta r^{-1} \pmod{n}$ to obtain the triple (v, m, s), where (v, m, s) is the electronic cash for w euros.

Deposit phase. The customer sends (v, m, s) to the merchant for a payment of x euros. The merchant verifies the electronic cash computing $s^{ev} \equiv H(m) \pmod{n}$. If the verification process is correct, the merchant forwards (v, m, s) to the bank. The bank verifies if v is in correct format and checks whether or not the electronic cash is fresh. Then, the bank deposits x funds into the merchant's account and keeps (v, m, s) in its database.

C. Date attachable electronic cash

Fan et al. [17] introduced the concepts of withdrawal date and effective date in the electronic cash. The scheme contains three participants (a customer, a merchant, and a bank) and consists of the following phases:

Initialization phase. The bank randomly selects two distinct primes p and q. Then, it computes $n = pq$ and $\varphi = (p - 1)(q - 1)$. The bank chooses a large integer e at random, where $1 < d < \varphi$ such that $ed \equiv 1 \pmod{\varphi}$. The bank keeps secret (d, p, q), and publishes (e, n) and a one-way hash function H(·).

Withdrawal phase. The customer randomly chooses an integer r and six numbers x_1, x_2, x_3, x_4, x_5, and x_6. Then, she computes $\alpha = H(m)r^e \pmod{n}$, where $m = H^{100}(x_1) \| H^{100}(x_2) \| H^{12}(x_3) \| H^{12}(x_4) \| H^{31}(x_5) \| H^{31}(x_6)$. The customer sends (α) to the bank. The bank computes $\beta = \alpha^d \pmod{n}$ and sends it to the customer. The bank deducts w funds (e.g. dinar, euro, etc.) from the customer's bank account.

Unblind phase. The customer computes $s = \beta r^{-1} \pmod{n}$ and obtains her electronic cash (m, s).

Deposit phase. When the customer wants to pay for a product, she attaches the current date to the electronic cash computing $\alpha_1 = H^a(x_1)$, $\alpha_2 = H^{100-a}(x_2)$, $\alpha_3 = H^b(x_3)$, $\alpha_4 = H^{12-b}(x_4)$, $\alpha_5 = H^c(x_5)$, $\alpha_6 = H^{31-c}(x_6)$, where $a = 1 +$ the two least significant digits of the current year, $b =$ current month, $c =$ current day. Then, she sends $(s, a, b, c, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6)$ to the merchant. The merchant verifies the electronic cash computing $s^e \equiv H(H^{100-a}(\alpha_1) \| H^a(\alpha_2) \| H^{12-b}(\alpha_3) \| H^b(\alpha_4) \| H^{31-c}(\alpha_5) \| H^c(\alpha_6)) \pmod{n}$. If the verification process is correct, the merchant forwards the electronic cash to the bank. The bank verifies the electronic cash and checks whether or not it is fresh. The bank stores $(s, a, b, c, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6)$ in a database to prevent a double-spending attack. Finally, the bank deposits x funds into the merchant's bank account.

D. Flexible date-attachment electronic cash

In this scheme [18], there are three participants (a customer, a merchant and a bank). The customer withdraws the m-cash from the bank and pays it to merchant. This scheme consists of five steps: 1) initializing; 2) withdrawing; 3) unblinding; 4) date-attaching; 5) depositing. The customer attaches the effective date in a phase called date-attaching.

Initialization phase. The bank generates two key pairs. The bank keeps secret (d, p, q) and ($d1$, p_1, q_1), and publishes (e, n), ($e1$, n_1), and a one-way hash function H(·).

Withdrawal phase. The customer randomly chooses an integer r and computes $\alpha = H(m)r^e$ (mod n), where m is a random number in each withdraw. Then, the customer sends (α) to the bank. The bank computes $\beta = \alpha^d$ (mod n) and sends it to the customer. The bank deducts w funds (e.g. dinar, euro, etc.) from the customer's bank account.

Unblind phase. The customer computes $s = \beta r^{-1}$ (mod n) and obtains her electronic cash (m, s).

Date-attachment phase. The customer randomly chooses an integer r_1 and computes $\alpha_1 = H(s)r_1^{e1}$ (mod n_1). The customer sends (α_1) to the bank. The bank computes $\beta_1 = \alpha_1^{d1}$ (mod n_1) and sends it to the customer. The customer computes $\delta = \beta_1 r_1^{-1}$ (mod n_1) and obtains the date slip (s, δ) of her electronic cash. Then, the customer sends (s, δ, yy, mm, dd) to the bank, where yy, mm, dd corresponds to the year, month, and the date for the effective date. The bank verifies if the date slip is valid computing $\delta^e \equiv H(s)$ (mod n_1). If the verification process is correct, the bank computes $\beta_2 = H(s \parallel yy \parallel mm \parallel dd)^{d1}$ (mod n_1) and sends it to the customer. The customer computes $s_1 = \beta_2 r_1^{-1}$ (mod n_1) and obtains the effective date of her electronic cash (m, s, s_1, yy, mm, dd).

Deposit phase. When a customer wants to pay for a product, she sends (m, s, s_1, yy, mm, dd) to the merchant. The merchant verifies the electronic cash computing $s^e \equiv H(m)$ (mod n) and $s_1^{e1} \equiv H(s \parallel yy \parallel mm \parallel dd)$ (mod n_1). Then, the merchant forwards the electronic cash to the bank. The bank verifies the electronic cash and check whether or not it is fresh. Finally, the bank deposits x funds into the merchant's bank account.

III. TOMIN SCHEME

In this section, we propose an efficient m-cash scheme with expiration-date attached to control the bank's database and memory of the customer's mobile device. The date is attached using a string concatenation and a one-way hash function. The proposal does not require several hundred hashes or complex mathematical operations. The m-cash maintains the privacy of the customers. Moreover, the scheme contemplates the deposit date for financial purposes. The proposed scheme includes the following participants.

- A customer is a person who uses a mobile device and m-cash to pay for products.
- A merchant is a person or vending machine that accepts m-cash.
- A bank is the entity that issues and manages m-cash.

The scheme consists of four phases: initializing, withdrawing, unblinding, and depositing. Figure 1 shows the proposed scheme.

- In the initialization step, the bank generates its private/public keys, and publishes the public key and one-way hash function.
- In the withdrawal step, the customer attaches the expiration date to the m-cash and withdraws her m-cash from the bank.
- In the unblinding step, the customer recovers her m-cash signed by the bank.
- In the depositing step, the customer pays w m-cash to a merchant and the bank deposits the funds in the merchant's account.

The notation used in the description of the proposed protocol is given in Table 1.

TABLE I.
DEFINITION OF NOTATIONS USED IN THE SCHEME

Notation	Meaning
B	Bank
C_i	ith customer
M_j	jth merchant
ID_M	Merchant's identity
δ	Expiration date
δ_1	Deposit date
w	The amount to withdraw
x	The amount to pay and to deposit
r, $seed$	Random numbers
$h_i = H^n(seed)$	Perform n times one-way hash function
l_i	The length of the hash chain
$\parallel$	The concatenation operation

The details of the scheme are as follows:

A. Initialization phase

The bank randomly selects two distinct primes p and q. Then, it computes $n = pq$ and $\varphi = (p-1)(q-1)$. The bank chooses a large integer e at random, where $1 < d < \varphi$ such that $ed \equiv 1$ (mod φ). The bank keeps secret (d, p, q), and publishes (e, n) and a one-way hash function H(·).

B. Withdrawal phase

We assume that the customer C_i wants to withdraw w units of m-cash. The customer C_i should perform the following steps:

- She randomly chooses two integers r and $seed$, and defines $v = w \parallel \delta$, where δ represents a short time pre-defined by the bank (e.g. 30 days).
- She computes $h_i = H^w(seed)$ and $\alpha = H(m)r^e$ (mod n), where $m = h_i \parallel l_i \parallel v$.
- She sends (α, v) to the bank B through a secure channel.

Upon receiving the message from the customer C_i, the bank B performs the following steps:

- It verifies the correct format of v.
- It computes $\beta = \alpha^d \pmod n$.
- It sends (β) to the customer C_i through a secure channel.

Finally, the bank deducts w funds (e.g. dinar, euro, etc.) from the customer's account. The transmitted messages of this phase are showed in Figure 1.

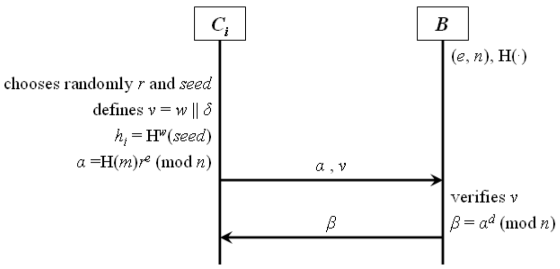

Figure 1. Withdrawing m-cash

C. Unblind phase

The customer C_i computes $s = \beta r^{-1} \pmod n$ and obtains her electronic cash (m, s).

D. Deposit phase

We assume that the customer C_i buys a product and must pay x units of m-cash. The customer C_i performs the following steps:

- She computes $h_{i-x} = \mathrm{H}^{i-x}(m)$ such that $\mathrm{H}^x(h_{i-x}) = h_i$.
- She computes $F = \mathrm{H}(x \parallel h_{i-x} \parallel ID_M \parallel \delta_I)$ and $G = \mathrm{H}(m \parallel s \parallel F)$.
- She sends (m, s, x, h_{i-x}, ID_M, δ_I, G) to the merchant M_j through a secure channel.

After the merchant M_j receives the message, she performs the following steps:

- She verifies whether or not δ_I is correct.
- She computes $F^* = \mathrm{H}(x \parallel h_{i-x} \parallel ID_M \parallel \delta_I)$ and $G^* = \mathrm{H}(m \parallel s \parallel F)$.
- She compares F^* with F and G^* with G.
- She sends (m, s, x, h_{i-x}, ID_M, δ_I, G) to the bank B through a secure channel.

Upon receiving the message from the M_j, the bank B performs the following steps:

- It verifies whether or not δ_I is correct.
- If computes $F = \mathrm{H}(x \parallel h_{i-x} \parallel ID_M \parallel \delta_I)$ and $G = \mathrm{H}(m \parallel s \parallel F)$.
- It compares F^* with F and G^* with G
- It verifies whether or not v and δ are correct.
- It checks whether or not h_i is fresh.
- It computes $s^e \equiv \mathrm{H}(m) \pmod n$ and $h_i = \mathrm{H}^x(h_{i-x})$.
- It deposits w funds into the merchant's account.
- It computes $s' = \mathrm{H}(m')^d \pmod n$, where $m = h_{i-x} \parallel l_{i-x} \parallel v$.
- It sends (m', s') to the merchant M_j through a secure channel.

Finally, the bank stores (m, s) in a database to prevent a double-spending attack and to delete the m-cash expire. Then, the merchant M_j forwards the m-cash to the customer C_i.

Figure 2 shows the transmitted messages among the customer C_i, merchant M_j and bank B.

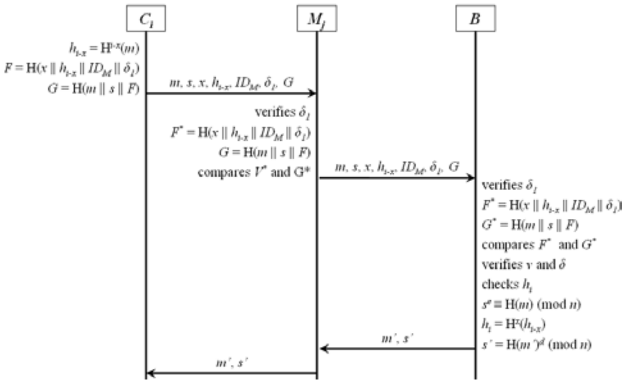

Figure 2. Depositing m-cash

IV. Analysis and Comparisons

In this section, we examine the security, performance and storage of the proposed scheme.

A. Security Analysis

In this sub-section, we analyse the security of the proposed scheme.

In the i-th withdrawal step, the bank cannot link s to β because it does not have the blinding factor r. Moreover, the merchant cannot obtain information about the customer from the payment information (m, s, x, h_{i-x}, ID_M, δ_I, G). The customer's privacy from the bank and the merchant is guaranteed in the withdrawing and depositing steps.

In the i-th withdrawal step, the bank verifies the correct format of v.

In the i-th unblind step, the customer cannot forge another m-cash (m, s) without the bank's secret key d. Factoring n is intractable, so that it is computationally infeasible for an attacker forging a valid m-cash (m, s), such that $s^*\mathrm{H}(m^*)m^* \equiv 1 \pmod n$.

In the i-th deposit step, the customer cannot use her m-cash after the expiration date δ. If the customer tries to pay using expired m-cash, the bank will reject the transaction.

In the i-th deposit step, the merchant can deposit the m-cash (m, s) just on the indicated date by the customer. If the merchant wants to deposit the m-cash after the deposit date δ_I, the bank will reject the transaction.

If the merchant tries to deposit the m-cash more than one time, the bank will detect the attack when it checks the validity of h_i.

If an attacker intercepts the payment information, she cannot deposit the m-cash (m, s) in her bank account because the m-cash is linked together with the merchant's identity ID_M via $F = \mathrm{H}(x \parallel h_{i-x} \parallel ID_M \parallel \delta_I)$. The difficulty of deriving $F^* = \mathrm{H}(x \parallel h_{i-x} \parallel ID_M{}^* \parallel \delta_I)$ and $G = \mathrm{H}(m \parallel s \parallel F^*)$ with other merchant's identity $ID_M{}^*$ relies on the strength of the one-way hash function.

In the *i*-th deposit step, the bank verifies whether or not h_i is fresh.

In the *i*-th deposit step, the bank verifies whether or not δ and δ_1 are correct.

In the *i*-th deposit step, the merchant cannot use the m-cash (m', s') to carry out a payment because she do not know the *seed* of the hash chain.

Given the payment information (m, s, x, h_{i-x}, ID_M, δ_1, G), the bank cannot find the instance of the withdrawing step that produced the m-cash.

B. Performance Analysis

Due to the resources constraints of mobile devices, the m-cash scheme must take efficiency evaluation into consideration. In this sub-section, we evaluate our scheme and compare with related schemes in Table 2. The notation T_h and T_{exp} are defined as the execution times for hash functions and exponential operations, respectively. The execution time of exponential operations, under a modulus n, is about $O(|n|)$ time, where $|n|$ denotes the bit length of n. On the other hand, the execution time of a hash function does not take longer than the exponential operations. The time complexity associated with the different operations can be expressed as $T_h << T_{exp}$.

The computational cost is defined as the total time of various operations executed in each step. According to the above definition, the computation cost in the withdrawal phase is time $(w + 1)T_h + 2T_{exp}$. The consumer requires T_{exp} time in the unblinding phase. The times requires by the customer, merchant and bank in the depositing phase are $(w + x + 2)T_h$, $2T_h$, and $(x + 4)Th + 2T_{exp}$, respectively.

TABLE II.
A PERFORMANCE COMPARISON BETWEEN OUR SCHEME AND RELATED SCHEMES

	[19]	[16]	[17]	[18]	Our
P_1	$T_h + 2T_{exp}$	$T_h + 2T_{exp}$	$(N + 1)T_h + 2T_{exp}$	$2T_h + 4T_{exp}$	$(2w + x + 2)T_h + 2T_{exp}$
P_2	$T_h + 2T_{exp}$	$T_h + 2T_{exp}$	$(O + 1)T_h + 2T_{exp}$	$2T_h + 4T_{exp}$	$2T_h$
P_3	$T_h + 3T_{exp}$	$3T_h + 6T_{exp}$	$(O + 1)T_h + 3T_{exp}$	$T_h + 5T_{exp}$	$(x + 4)T_h + 3T_{exp}$
P_4	6	6	6	8	6

P_1: Computational cost of the customer.
P_2: Computational cost of the merchant.
P_3: Computational cost of the bank.
P_4: Number of rounds in the scheme. A round is the number of messages exchanged between two participants.
N: $H^{100}(x_1) \| H^{100}(x_2) \| H^{12}(x_3) \| H^{12}(x_4) \| H^{31}(x_5) \| H^{31}(x_6)$.
O: $H^{100-a}(\alpha_1) \| H^{a}(\alpha_2) \| H^{12-b}(\alpha_3) \| H^{b}(\alpha_4) \| H^{31-c}(\alpha_5) \| H^{c}(\alpha_6)$.

C. Storage Analysis

In this sub-section, we compare our scheme with the related schemes in terms of storage capacity. We use the following assumptions to evaluate the storage capacity of our scheme and related works. Assume the ID_M, l_i, r, *seed*, x and w are 40-bit length, and δ and δ_1 are 64-bit length; the large prime in modular operation is 1024-bit length. We also assume that the output size of a one-way hash function is 128-bit length. In our scheme, the customer stores the following parameters in her mobile device: m, s, r, and *seed*, so the memory needed in the customer's mobile device is 1376((4*40) + 64 + 128) bits. The comparison of our scheme and related schemes are shown in Table 3.

TABLE III.
A STORAGE OMPARISON BETWEEN OUR SCHEME AND RELATED SCHEMES

	[19]	[16]	[17]	[18]	Our
P_1	1064 bits	1168 bits	1792 bits	2136 bits	1376 bits
P_2	e, n	e, n	e, n	e, n, e_1, n_1	e, n
P_3	m, s	m, s, v	m, s	m, s, s_1, yy, mm, dd	m, s

P_1: Storage size of the m-cash.
P_2: Public keys.
P_3: M-cash.

Moreover, we summarize the functionality of the proposed scheme and make comparison with related schemes in Table 4. It demonstrates the advantage of our scheme.

TABLE IV.
A GENERAL COMPARISON BETWEEN OUR SCHEME AND RELATED SCHEMES

	[19]	[16]	[17]	[18]	Our
P_1	No	Yes	No	No	No
P_2	No	Yes	No	No	Yes
P_3	No	No	Yes	Yes	Yes
P_4	No	Partially	No	No	Yes
P_5	-	H(·), and (e, n)	Several H(·), and (e, n)	H(·), and many (e, n)	H(·), and (e, n)
P_6	Yes	Yes	Yes	Yes	Yes
P_7	No	Yes	No	No	No
P_8	No	No	No	No	Yes

P_1: Withdrawal date.
P_2: Expiration date.
P_3: Deposit date.
P_4: Controls customer's database.
P_5: Operations used to attach the date.
P_6: Multiple payments.
P_7: Need to withdraw for each payment.
P_8: Attaches the merchant's identity to the m-cash.
H(·) = One-way hash function.
(e, n) = Exponential operation.

V. CONCLUSIONS

In this paper, we have proposed TOMIN, an efficient, practical, and trustworthy m-cash with expiration-date attached. In this scheme, the expiration date is important to control the bank's database and the customer's mobile device memory. The scheme uses a concatenation operation and a one-way hash function to attach the expiration date to m-cash. We demonstrate that our scheme satisfies the basic security requirements and prevents forgery and double-spending attacks. It is well suited to mobile devices.

In addition, the proposed scheme has the following merits: 1) the storage capacity and computational cost are more efficient than those of previous works; 2) our scheme can resist several attacks; 3) the deposit date is attached by the customer; and 4) the customer attaches the merchant's identity into the m-cash.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish public funded projects ARES (CONSOLIDERINGENIO-2010 CSD2007-00004) and ITACA (TSI2006-13409-C02-02), and graduate scholarship from CONACYT (Mexico).

REFERENCES

[1] R. Abbadasari, R. Mukkamala, and V. Kumari, "Mobicoin: digital cash for m-commerce," in *Distributed Computing and Internet Technology*, vol. LNCS 3347, R. K. Ghosh and H. Mohanty, Eds. Springer-Verlag, 2004, pp. 441-451.

[2] Z. Y. Hu, Y. W. Liu, X. Hu, and J. H. Li, "Anonymous micropayments authentication (AMA) in mobile data network," in *Proc. of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM'04*). Hong Kong, 2004, pp. 46-53.

[3] J. Gao, K. Edunuru, J. Cai, and S. Shim, "P2P-paid: a peer-to-peer wireless payment system," in *Proc. of the 2nd IEEE International Workshop on Mobile Commerce and Services* (*WMCS'05*). Munich, Germany, 2005, pp. 102-111.

[4] B. K. Lee, T. C. Lee, and S. H. Yang, "A MEP (mobile electronic payment) and intcA protocol design," in *High Performance Computing and Communications*, vol. LNCS 3726, L. T. Yang, O. F. Rana, B. Di Martino, and J. Dongarra, Eds. Springer-Verlag, 2005, pp. 331-339.

[5] S. Fong, and E. Lai, "Mobile mini-payment scheme using SMS-credit," in *Computational Science and Its Applications*, vol. LNCS 3481, O. Gervasi, M. L. Gavrilova, V. Kumar, et al., Eds. Springer-Verlag, 2005, pp. 1106-1114.

[6] M. S. Hwang, and P. C. Sung, "A study of micro-payment based on one-way hash chain," *International Journal of Network Security*, vol. 2, pp. 81-90, 2006.

[7] J. Téllez, and J. Sierra, "A secure payment protocol for restricted connectivity scenarios in m-commerce," in *E-Commerce and Web Technologies*, vol. LNCS 4655, G. Psaila and R. Wagner, Eds. Springer-Verlag, 2007, pp. 1-10.

[8] R. J. Hwang, S. H. Shiau, and D. F. Jan, "A new mobile payment scheme for roaming services," *Electronic Commerce Research and Applications*, vol. 6, pp. 184-191, 2007.

[9] R. Martínez-Peláez, F. Rico-Novella, and C. Satizábal, "Mobile payment Protocol for micropayments: withdrawal and payment anonymous," in *Proc. of the 2nd New Technologies, Mobility and Security* (*NTMS'08*). Tangier, Morocco, 2008, pp. 1-5.

[10] M. Hassinen, K. Hyppönen, and E. Trichina, "Utilizing national public-key infrastructure in mobile payment systems," *Electronic Commerce Research and Applications*, vol. 7, pp. 214-231, 2008.

[11] M. S. Manasse, "The millicent protocol for electronic commerce," in *Proc. of the 1st USENIX Workshop on Electronic Commerce*. New York, USA, 1995, pp. 117-123.

[12] R. Rivest, and A. Shamir, "Payword and micromint: two simple micropayment schemes," in *Security Protocols*, vol. LNCS 1189, G. Goos, J. Hartmanis, and J. V. Leeuwen, Eds. Springer-Verlag, 1996, pp. 69-87.

[13] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, pp. 770-772, 1981.

[14] NIST, NIST-FIPS-PUB-186-2, Digital Signature Standard, National Institute of Standards and Technology, U.S. Department of Commerce, 2001.

[15] R. Rivest, The MD5 Message-digest Algorithm, RFC 1321, Internet Acitivites Board, Internet Privacy Task Force, 1992.

[16] M. Abe, and E. Fujisaki, "How to date blind signatures," in *Advances in Cryptology*, vol. LNCS 1163, K. Kwangjo and M. Tsutomu, Eds. Springer-Verlag, 1996, pp. 244-251.

[17] C. I. Fan, W. K. Chen, and Y. S. Yeh, "Date attachable electronic cash," *Computer Communications*, vol. 23, pp. 425-428, 2000.

[18] C. C. Chang, and Y. P. Lai, "A flexible date-attachment scheme on e-cash," *Computers & Security*, vol. 22, pp. 160-166, 2003.

[19] D. Chaum, "Blind Signatures for untraceable payments," in *Advances in Cryptology*, Crypto'82, D. Chaum, R. L. Rivest, and A. T. Sherman, Eds. Springer-Verlag, 1983, pp. 199-203.

Rafael Martínez-Peláez received his degree in Computational System Engineering from the University of the Valley of Mexico (México) in 2004. Currently, he is carrying out a PhD in Telematic Engineering at the Technical University of Catalonia (Spain). His main areas of interest are electronic and mobile payment systems, and applications of smart card and biometrics with security protocols design.

Francisco Rico-Novella received his degree in Telecommunication Engineering and his PhD from the Technical University of Catalonia (Spain) in 1989 and 1995, respectively. Presently, he works in the Department of Telematic Engineering with the Telematics Service group. His current research interests include network security and electronic commerce.

Cristina Satizábal received her degree in Electronic and Telecommunications Engineering from Cauca University (Colombia) in 2000 and her PhD in Telematic Engineering from the Technical University of Catalonia (Spain) in 2007. Currently, she is part of the Departamento de Automática y Electrónica of Universidad Autónoma de Occidente (Colombia) and belongs to LOGOS research group.