Anonymous Multi-Vendor Micropayment Scheme based on Bilinear Maps

Andrea Huszti

Faculty of Informatics
University of Debrecen
26. Kassai St.
H4028 Debrecen
Hungary
huszti.andrea@inf.unideb.hu

Keywords: micropayment, multi-vendor, anonymous, bilinear maps

Abstract. A hash-based micropayment scheme is introduced that takes advantage of good properties of bilinear maps, provides anonymity of the customers and makes it possible to shop at multiple vendors. The proposed scheme minimizes computational and financial costs. We proved that it possesses secure payment authorization under the chosen-target Computational Diffie-Hellman assumption in the random oracle model and customer's anonymity and coin unreusability against any passive adversary with unlimited computational power.

1 Introduction

Micropayment schemes were invented as a consequence of Internet applications. In particular, there are content and service providers that charge very small amount (e.g. less than a dollar). The usual online purchasing method - payment by credit cards - requires minimal transaction fee and other extra costs, hence it is not applicable for charging small amounts. Special payment systems, so-called micropayment schemes, are required. One can read an overview of micropayment schemes in [17].

Designing micropayment schemes requires special care. First of all, electronic payment systems deal with personal, confidential financial data, that should be protected against malicious attackers. On the other hand financial and also computational costs should be minimized.

In order to minimize financial costs the e-payment scheme should be off-line, that means the broker is off-line during the shopping process (*i.e.* broker is not required to verify whether users can cover their payments or not). We can decrease customers' expenses, if they do not have to possess signature certificates, either.

Computational costs especially for customers during the shopping process should be reduced. Usually, slow asymmetric cryptographic building blocks should be avoided.

One of the most well known hash-based micropayment scheme is the PayWord scheme [13], introduced in 1997. PayWord is not anonymous and designed to be one-vendor, thus if we apply it for multiple vendors, it does not protect against double spending.

Our scheme is multi-vendor and provides anonymity for the customers. Comparing to other protocols in the literature, in order to accomplish multi-vendor property we do not use blacklists ([15],[16]), that means brokers maintain a list of users who did not pay for the products or services. In 2003 Payeras-Capella, Ferrer-Gomila and Huguet-Rotger designed an anonymous hash-based scheme (see [12]), that is semi-offline, *i.e.* every time when a customer is willing to switch to another vendor he should contact with the broker to get the authorized certificate. Therefore in [12] if a customer shops at k vendors, then k signature verifications are needed by the customer. We designed our solution to be off-line. In 2010 Hosseinkhani, Tarameshloo and Shajari in [8] introduced an anonymous, multi-vendor hash-based scheme. They achieved anonymity via an Insert Ticket, that is different for different customers and makes it possible to insert values in the broker's database. This database contains information about the customer's last spent coin, and vendors check it for every purchase. In this scheme double spending protection is achieved via an all time available on-line database, that increases costs.

Bilinear pairings, namely Weil pairing and Tate pairing of algebraic curves were used in cryptography for MOV attack [11] using Weil pairing and FR attack [6] using Tate pairing. These attacks reduce the discrete logarithm problem on some elliptic or hyperelliptic curves to the discrete logarithm problem in a finite field. Bilinear pairings have recently been used to design cryptographic protocols, since as a consequence new constructions of primitives appeared. These primitives either cannot be built using other techniques (e.g. three-party one-round Diffie-Hellman key agreement in [10], Identity-based encryption in [2]), or they can be created via traditional methods, but pairings provide improved functionality (e.g. threshold signature, multisignature, blind signature [4]). Here, we show how these new primitives can be applied in micropayments to accomplish necessary security requirements reducing both financial and computational costs.

We also give a detailed computational security evaluation of our scheme. Most of the proposed micropayment schemes do not prove security properties. There are few solutions ([1],[9]) that give formal security evaluation based on applied pi calculus [14] with the help of automatic protocol verifier called Proverif [3]. We prove that our scheme provides *secure payment authorization* under the chosen-target Computational Diffie-Hellman assumption in the random oracle model and *customer's anonymity and coin unreusability* against any passive adversary with unlimited computational power.

The rest of the paper is organized as follows: Basic definitions, notation and building blocks are given in section 2. In section 3 the proposed micropayment scheme is detailed. Section 4 contains the security evaluation of the scheme

starting with the definitions of the requirements that followed by the proofs. At the end, time and space complexity are given and the conclusion.

2 Preliminaries

2.1 Definitions, notation

In order to give a security evaluation we review some basic definitions, notation. A negligible function is a function $\epsilon(\lambda)$ such that for all polynomials $poly(\lambda)$, $\epsilon(\lambda) < 1/poly(\lambda)$ holds for all sufficient large λ. We call an algorithm efficient, if it is a probabilistic Turing machine running in expected polynomial time. An adversary $\mathcal{A}$ is a PPT (probabilistic polynomial-time) interactive Turing machine. For a finite set X, let $x \leftarrow X$ denote the algorithm that samples an element uniformly random from X.

Our micropayment scheme is based on bilinear pairings. Let us review the definition of the admissible bilinear map [2].

Definition 1. *Let G_1 and G_2 be two groups of order q for some large prime q. A map $e : G_1 \times G_1 \rightarrow G_2$ is an admissible bilinear map if satisfies the following properties:*

1. *Bilinear: We say that a map $e : G_1 \times G_1 \rightarrow G_2$ is bilinear if $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and all $a, b \in Z$.*
2. *Non-degenerate: The map does not send all pairs in $G_1 \times G_1$ to the identity in G_2. Since G_1, G_2 are groups of prime order, if P is a generator of G_1 then $e(P, P)$ is a generator of G_2.*
3. *Computable: There is an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in G_1$.*

We should mention that bilinearity can be restated to for all $P, Q, R \in G_1$ $e(P + Q, R) = e(P, R)e(Q, R)$ and $e(P, Q + R) = e(P, Q)e(P, R)$. We can find G_1 and G_2 where these properties hold. The Weil and Tate pairings prove the existence of such constructions. Typically, G_1 is an elliptic-curve group and G_2 is a finite field. Let us review the relevant security problems.

Definition 2. *Let P, aP, bP are given, for some $a, b \in Z_q^*$. The problem of computing abP is called Computational Diffie-Hellman problem (CDHP) in G_1.*

We assume that CDHP is intractable, which means there is no polynomial time algorithm to solve CDHP with nonnegligible probability.

Definition 3. *Let P, aP, bP, cP are given, for some $a, b, c \in Z_q^*$. The problem of deciding whether $c \equiv ab \mod q$ is called Decisional Diffie-Hellman problem (DDHP) in G_1.*

DDH problem in G_1 can be solved in polynomial time by verifying $e(aP, bP) = e(P, cP)$, hence DDH problem in G_1 is easy.

Definition 4. *When DDHP is easy but the CDHP is hard on the group G_1, we call G_1 a* Gap Diffie-Hellman (GDH) group*.*

Such groups can be found on supersingular elliptic curves or hyperelliptic curves over a finite field.

2.2 Building blocks

In the proposed scheme we apply a blind signature scheme given in [4], that works as follows. Let G_1 be a GDH group, $P \in G_1$ a generator of G_1 and $H : \{0,1\}^n \to G_1^*$ is a one way hash function. We assume a bilinear map $e : G_1 \times G_1 \mapsto G_2$ is given. The signer randomly chooses $s \in Z_q^*$, and calculates sP. Users hold the public key $PK = (G_1, P, H, sP)$. In order to blindly sign a message $M \in \{0,1\}^*$, the user picks a random number $r \in Z_q^*$, computes $\overline{M} = rH(M)$ and sends it to the signer. The signer knows secret key $SK = s$, computes $\overline{\sigma} = s\overline{M}$ and sends it to the user. The users computes $\sigma = r^{-1}\overline{\sigma} = sH(M)$ and outputs (M, σ). Signature σ on M is a valid, since $e(H(M), sP) = e(sH(M), P)$.

Signatures we receive in this way are called short signatures [5] and have many useful features. They are approximately 170 bits long instead of 320 or 1024, having the same security level. In protocols dealing with many signatures originating from the same user, bilinear short signatures can be verified in an aggregate way, calculating only two bilinear maps. Hence we can significantly increase efficiency. We use bilinear blind short signatures to authorize the coin commitment values and also provide customer's anonymity. Security of this blind signature scheme is based on the chosen-target CDH problem in G_1.

Definition 5. *(chosen-target CDH assumption)*
Let $G =< P >$ be a GDH group of prime order q, and $y \leftarrow Z_q^*$ be a secret, and $Y = yP$ a public key generated according to security parameter λ. The adversary is given (P, G, q, Y) and has access to a target oracle $\mathcal{O}_T$, that returns random points Z_i from G, and also a helper oracle $\mathcal{O}_H$, that calculates yQ for an input point Q. Let n_T and n_H denote the number of queries the adversary can make to the target and helper oracle, respectively. Adversary $\mathcal{A}$ outputs: $((X_1, j_1), \ldots, (X_l, j_l))$. The advantage of the adversary $Adv_{G,\mathcal{A}}^{ct-CDH}(\lambda)$ defined by

$$Pr[\forall 1 \leq i \leq l \ \exists 1 \leq j_i \leq n_T \ X_i = yZ_{j_i} \ \wedge \ X_i \text{ are distinct } \wedge \ n_H < n_T].$$

The chosen-target Computational Diffie-Hellman assumption states, that for all PPT adversary $Adv_{G,\mathcal{A}}^{ct-CDH}(\lambda)$ is negligible.

Observe, that if the adversary $\mathcal{A}$ makes exactly one query to the $\mathcal{O}_T$, then the chosen-target CDH assumption is equivalent to the standard CDH assumption. We assume that chosen-target CDH problem is intractable for all groups, where the standard CDH problem is hard.

3 The proposed scheme

Let us introduce our micropayment scheme. We consider the following participants. There is a *Broker*, that authorizes the coins and checks whether there is sufficient fund on user's account. There are k *Vendors* that sell their products and there are several *Customers*, who intend to shop at more then one Vendor.

In order to run the protocol, system parameters, secret and public keys are generated. First of all bilinear map $e : G_1 \times G_1 \mapsto G_2$ is set determining groups $G_1 =< P >, G_2$ with prime order q. A security parameter λ and two hash functions are chosen, $H : G_1 \mapsto G_1$ is necessary for signature generation and $H_q : Z_q^* \mapsto Z_q^*$ is used for generating the coins.

Secret and public keys are generated for the Broker and all the Vendors. The length of the keys depends on the security parameter λ. The Broker's key pair is $(SK, PK) = (b, bP)$,where $b \leftarrow Z_q^*$ and SK denotes the secret and PK the public key. Broker's secret key is used for signature generation, the public key for signature verification. Similarly, all Vendors receive $(SK, PK) = (v^i, v^i P)$ key pairs, where $v^i \leftarrow Z_q^*$. In the proposed scheme Customers are not required to possess a key pair.

The scheme consists of three stages. In the first stage Customers apply for the Broker's authorization on their coins. During the second stage Customers proceed their shopping with the Vendors and at the end Vendors redeem the coins at the Broker.

Customer-Broker relationship

A Customer first decides about the Vendors where he would like to shop. Let us denote the number of chosen Vendors by k. For each Vendor he generates a hash chain. The elements of the chain are the coins. Random values $w_n^i \leftarrow Z_q^*$, where $i = 1, \ldots, k$ are generated, and the hash chain elements $w_j^i = H_q(w_{j+1}^i)$, where $j = n-1, \ldots, 0$ are calculated. We call values w_0^i commitment values. These commitment values are made vendor specific by a multiplication: $w_0^i v^i P$, where $v^i P$ is the Vendor's public key.

A Customer requires the Broker's authorization via a blind signature scheme, he generates $r^i \leftarrow Z_q^*$, sends $\rho^i = r^i H(w_0^i v^i P)$ and his identity number ID to the Broker on an open channel. He also calculates

$$\Gamma = \sum_{i=1}^{k} H(w_0^i v^i P).$$

The Broker authenticates the Customer in a secure way. The entity authentication and authorization might happen even off-line. The Broker debits the proper amount on the Customer's account and sends $\epsilon^i = b r^i H(w_0^i v^i P)$, where $i = 1, \ldots, k$ back.

The Customer after receiving the messages, with the knowledge of r^i, calculates $\sigma^i = bH(w_0^i v^i P)$ and

$$\Delta = \sum_{i=1}^{k} bH(w_0^i v^i P) = b \sum_{i=1}^{k} H(w_0^i v^i P).$$

The Customer verifies whether

$$e(\Gamma, bP) = e(\Delta, P)$$

Observe that the Customer in order to verify all the signatures calculates only two bilinear maps.

Customer-Vendor relationship

In order to start the shopping process with Vendor i, the Customer sends the certificate $\sigma^i = bH(w_0^i v^i P)$ and commitment value w_0^i on an open channel. The vendor calculates $H(w_0^i v^i P)$ from w_0^i, and verifies whether $e(\sigma^i, P) = e(H(w_0^i v^i P), bP)$ and stores w_0^i. The customer starts spending the coins with w_1^i, w_2^i, etc. After receiving w_j^i the vendor verifies whether $H(w_j^i)$ equals to the value stored. If the verification holds, then the vendor stores w_j^i and sends the product.

Observe, that for the verification processes the broker does not need to be available.

Vendor-Broker relationship

After receiving a few coins, that can happen at the end of the day or after few days, the vendor starts to redeem them. He sends $(\sigma^i, w_0^i, w_l^i, l)$ to the broker. The broker verifies the certificate by calculating $H(w_0^i v^i P)$ and $bH(w_0^i v^i P)$ and correctness of the last coin by $H^l(w_l^i) = w_0^i$. If all verifications hold, then transfers the proper amount to the vendor.

4 Security evaluation

4.1 Security requirements

We consider three security requirements for micropayment schemes: anonymity, secure payment authorization and unreusability. We define anonymity and secure payment authorization properties via experiments involving an adversary $\mathcal{A}$ and the challenger.

Payment authorization guarantees a proof for the vendor, that there is sufficient fund on the user's account. This proof or certificate is created by the broker. *Secure* payment authorization is achieved, if the certificate is undeniable.

Definition 6. *(Secure Payment Authorization) The experiment is parameterized by security parameter λ and l.*

1. *The challenger generates the public system parameters, which include groups $G_1 =< P >$, G_2 and the bilinear map e, runs key generation algorithm for the input 1^λ. Public keys are given to the adversarial user $\mathcal{A}$.*
2. *Adversary $\mathcal{A}$ makes polynomial number l certificate queries from the Broker. The Broker provides valid certificates to $\mathcal{A}$.*
3. *Adversary $\mathcal{A}$ outputs a list of message-certificate pairs: $(m_1, \sigma_1), \ldots, (m_t, \sigma_t)$.*

We define the advantage of $\mathcal{A}$ by

$$Adv_{MS,\mathcal{A}}^{SPA}(\lambda) = Pr[\forall 1 \leq i \leq t \; Ver_{PK_B}(m_i, \sigma_i) = 1 \wedge l < t],$$

where $Ver_{PK_B}(m_i, \sigma_i) = 1$ denotes, that certificate σ_i is valid for message m_i. A micropayment scheme provides secure payment authorization if for all PPT

adversary $Adv^{SPA}_{MS,\mathcal{A}}(\lambda)$ *is negligible, where probability is taken over the coin-flips of* $\mathcal{A}$*, as well as the random coins used in the experiment for key generation.*

$\mathcal{A}$ chooses a random value w that is different from the previous ones, and must then produce a valid signature on $H(wv_iP)$. If he can produce any such document/signature pair which is accepted by the verication algorithm, then the attack is successful.

There are situations when customers do not want to reveal their real identity during their shopping process. We also consider providing anonymity for the users.

Definition 7. *(Anonymity) The experiment is parameterized by security parameter* λ *and a bit* b*.*

1. *The challenger generates the public system parameters, which include groups* $G_1 =< P >$*,* G_2 *and the bilinear map* e*. Broker's and Vendors' secret and public keys are generated for input* 1^λ *and they are given to the adversarial user* $\mathcal{A}$*.*
2. *Adversary* $\mathcal{A}$ *outputs a pair of identity numbers* ID^0, ID^1*, that represent two different customers.*
3. *The protocol is run with the customer possessing* ID^b *and adversary* $\mathcal{A}$*, where* b *is the randomly chosen input bit.*
4. *Adversary* $\mathcal{A}$ *outputs a bit* b' *according to the knowledge* $\mathcal{A}$ *gained colluding with the Broker and the Vendors.*

We define the advantage of $\mathcal{A}$ *by*

$$Adv^{Anon}_{MS,\mathcal{A}}(\lambda) = |2Pr[b = b'] - 1|.$$

A micropayment scheme provides customer anonymity if for all PPT adversary $Adv^{Anon}_{MS,\mathcal{A}}(\lambda)$ *is negligible, where probability is taken over the coin-flips of* $\mathcal{A}$*, as well as the random coins used in the experiment for key and identity number generation.*

The proposed micropayment scheme is off-line, hence the Broker is not able to prevent double spending. Being a multi-vendor scheme we show, that coins already spent are not reusable, neither at the same vendor, nor at different ones.

Definition 8. *(Coin unreusability) A micropayment scheme provides coin unreusability, if an adversary resends a coin that was already spent before, then the vendor detects it with overwhelming probability.*

4.2 Results

In this section we show that our proposed scheme is secure, *i.e.* it provides payment authorization, user's anonymity and coin unreusability.

Theorem 1. *(Payment authorization) The proposed scheme provides secure payment authorization under the chosen-target Computational Diffie-Hellman assumption in the random oracle model.*

Proof. We prove the security of our scheme by contradiction in the random oracle model. We suppose our scheme is not secure, there exists an adversary $\mathcal{A}$ that is able to break secure payment authorization of the scheme. If $\mathcal{A}$ exists, then we are able to build an efficient simulator algorithm $\mathcal{S}$, that with the help of $\mathcal{A}$ succeeds in breaking the chosen-target CDH assumption, that leads to a contradiction.

Let $G_1 =< P >$ and G_2 be groups of prime order q, where G_1 is a GDH group, and given $e : G_1 \times G_1 G_1 \rightarrow G_2$. After key generation simulator $\mathcal{S}$ receives public key $Y = yP$, where $y \leftarrow Z_q^*$ is the secret key. With the knowledge of (P, q, G_1, G_2, e, Y) algorithm $\mathcal{S}$ simulates the challenger for $\mathcal{A}$ in the following:

1. $\mathcal{S}$ provides $\mathcal{A}$ public parameters (P, q, G_1, G_2, e) and also gives $Y + rP$ as a public key, where $r \leftarrow Z_q^*$.
2. In order to respond valid certificates, $\mathcal{S}$ has access to a target oracle $\mathcal{O}_T$ and a helper oracle $\mathcal{O}_H$. $\mathcal{O}_T$ works as follows. $\mathcal{S}$ maintains a list of tuples (m_i, a_i, b_i, c_i), that is empty at the beginning. For a hash request of a message $m_i \in G_1$:
   - If m_i is on the list, then $\mathcal{S}$ outputs a_i as a hash value.
   - Otherwise, $\mathcal{S}$ generates a random bit b_i with $Pr[b = 0] = \frac{1}{n_T}$ probability, chooses $c_i \leftarrow Z_q^*$ and calculates $a_i = (1 - b_i)m_i + c_i P$. Finally $\mathcal{S}$ inserts (m_i, a_i, b_i, c_i) to the list.

   Observe, that a_i is uniform in G_1 and independent of $\mathcal{A}$'s view, hence $\mathcal{S}$ perfectly simulates the real has query operation.
3. Helper oracle $\mathcal{O}_H$ is constructed in a way, that for an input m_i first calls the target oracle $\mathcal{O}_T$ to calculate $a_i = H(m_i)$, then:
   - If $b_i = 0$, then $\mathcal{S}$ terminates.
   - Otherwise, signature $\sigma_i = c_i yP + ra_i = (y + r)c_i P$ is returned back. Observe, that σ_i is a valid signature for m_i under the public key $Y + rP$.
4. Adversary $\mathcal{A}$ asks the target oracle $\mathcal{O}_T$ to calculate hash of m, and with the help of $H(m)$ provides a list of message-certificate pairs: $(m_1, \sigma_1), \ldots, (m_t, \sigma_t)$, that contains the (m, σ) pair.
   - If σ is not a valid signature for m, then $\mathcal{S}$ terminates.
   - Otherwise, $\mathcal{S}$ looks for m from the list of $\mathcal{O}_T$. Let denote this tuple by (m, a, b, c). If $b = 1$, then $\mathcal{S}$ terminates, if $b = 0$, then $\mathcal{S}$ calculates $\sigma - rm - ra$. We remark, that $a = m + cP$, and assuming that σ is valid:

     $$\sigma - rm - rcP = (y + r)(m + cP) - rm - rcP = y(m + cP).$$

     $\mathcal{S}$ outputs $(m_i, c_i yP)$ for all signatures that are generated with the help of $\mathcal{O}_H$ and $(m, \sigma - rm - rcP)$.

   Observe, that $\mathcal{S}$ outputs a correct signature on m.
   We prove that $\mathcal{S}$ generates the correct output with non-negligible probability. We consider the situation, when $\mathcal{S}$ does not terminate. There are events that should happen:

– ε_1: For certificate queries $\mathcal{S}$ does not terminate, *i.e.* $b_i = 1$.
– ε_2: For message m in the (m, a, b, c) tuple $b = 0$.
– ε_3: $\mathcal{A}$ provides valid (m, σ) pair.

Let us denote the probability that $\mathcal{S}$ succeeds by $Pr[\varepsilon]$. It is easy to see that $Pr[\varepsilon] = Pr[\varepsilon_1 \wedge \varepsilon_2 \wedge \varepsilon_3] = Pr[\varepsilon_1] \cdot Pr[\varepsilon_2|\varepsilon_1] \cdot Pr[\varepsilon_3|\varepsilon_2]$.
Assuming there are $n_T - 1$ target oracle queries $Pr[\varepsilon_1] = (1 - \frac{1}{n_T})^{(n_T-1)} > \frac{1}{e}$, where e denotes the base of the natural logarithm. Similarly, $Pr[\varepsilon_2] = \frac{1}{n_T}$. The probability of event ε_3 is the advantage $Adv^{SPA}_{MS,\mathcal{A}}(\lambda)$. Therefore $Pr[\varepsilon] > \frac{1}{en_T} \cdot Adv^{SPA}_{MS,\mathcal{A}}(\lambda)$. Since $Adv^{SPA}_{MS,\mathcal{A}}$ is non-negligible, $Pr[\varepsilon]$ is also non-negligible.
We show that $\mathcal{S}$ is efficient. $\mathcal{S}$ makes n_T hash and n_H signature queries, for each signature query a hash query also happen. Including calculating the output, efficiency of $\mathcal{S}$ is $(n_T + 3n_H + 2)MULT$ and $(n_H + 2)ADD$, where $MULT$ and ADD denote operations multiplication and addition in the GDH group.

Theorem 2. *(Anonymity, coin unreusability) The proposed micropayment scheme provides customer's anonymity and coin unreusability against any passive adversary with unlimited computational power.*

Proof. First we deal with the customer's anonymity. It is sufficient to prove that, for any view (ρ, ID, ϵ) of the adversary $\mathcal{A}$ and any message-certificate pair (w_0, σ), there exists a blind factor that maps the view and the message-certificate pair.

Let $(\rho^i, ID^i, \epsilon^i)$ for $i = 0, 1$ two views of $\mathcal{A}$ and (w_0^j, σ^j) two message-signature pairs from customer $j = 0, 1$. Since $\mathcal{A}$ colludes with the Broker and all the Vendors, we assume that $\mathcal{A}$ observes two views, that correspond to the same Vendor with keys (v, vP). We state that, with the knowledge of vP,

$$s = \rho^i[H(w_0^j vP)^{-1}]$$

is a correct blind factor for any $(\rho^i, ID^i, \epsilon^i), (w_0^j, \sigma^j)$ pairs. Since

$$s\sigma^j = \rho^i[H(w_0^j vP)^{-1}]bH(w_0^j vP) = b\rho^i = \epsilon^i.$$

It is easy to see that, even if $\mathcal{A}$ knows the Broker's and Vendors' secret and public keys (b, bP, v^i, v^iP) for $i = 1, \ldots, k$, adversary $\mathcal{A}$ does not gain more knowledge about the chosen blind factor.

Since $(\rho^0, ID^0, \epsilon^0)$ and $(\rho^1, ID^1, \epsilon^1)$ have the same relation to (w_0^j, σ^j), any adversary with unlimited computational power can guess j correctly with probability exactly $1/2$.

Let us prove coin unreusability of the scheme. We assume, that a passive adversary $\mathcal{A}$ possesses a valid certificate $\sigma^i = bH(w_0^i v^i P)$ and the commitment value w_0^i that are sent to vendor i with public key v^iP. Without the loss of generality, we consider double spending of w_1^i. We assume that coin w_1^i is already sent once, hence vendor i stores w_1^i as the last coin in the database. Resending these values to a vendor leads to two cases:

1. Tuple (σ^i, w_0^i, w_1^i) is sent to vendor i. The vendor notice, that these values are already received before and checks whether $H(w_1^i)$ is the stored value.
2. Tuple (σ^i, w_0^i, w_1^i) is sent to vendor j, where $i \neq j$. The vendor looks for the certificate in his database, since he does not find it verifies the validity of σ^i by testing $e(H(w_0^i v^j P), bP) = e(\sigma^i, P)$.

In both cases the vendor detects double spending.

5 Comments

Due to their good properties, bilinear maps are often used in cryptographic protocols. Some pairing-based constructions provide breakthroughs, that other techniques cannot. We take advantage of aggregate signature verification in efficiency. We achieved that customers verify all the certificates by calculating only two bilinear maps.

Customers' time complexity during the shopping process is very low, only hash calculations are made. In the registration process, that happens only once, customers compute $2k$ multiplications, $2k$ additions and k multiplicative inverses in G_1 and only two bilinear maps. We refer to [7], that one (supersingular) mapping $e : G_1 \times G_1 \mapsto G_2$ is approximately equal to eight multiplications in G_1 for a security level of 256 bits.

Vendors' time complexity is only one multiplication and a hash calculation in G_1 and two bilinear map computations per a customer. Brokers proceed k multiplications in G_1 per a customer during registration and two multiplications in G_1 per a certificate besides hash calculations.

Considering implementation of our scheme, the Broker is off-line and customers do not need to have certificates, hence we minimized financial expenses.

Our scheme minimizes space complexity on the user side. Users store k hash root values for calculating hash chain elements with the indexes that show which coins were spent already and also store k short signatures.

6 Conclusion

Bilinear maps are used for broad wide of cryptographic applications. We introduced a hash-based micropayment scheme that is anonymous and multi-vendor. By using bilinear maps we increased efficiency and decreased space complexity. We also proved that our scheme provides secure payment authorization under the chosen-target Computational Diffie-Hellman assumption in the random oracle model and customer's anonymity and coin unreusability against any passive adversary with unlimited computational power.

Acknowledgment

The publication was supported by the TÁMOP-4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed

by the European Social Fund. The author is also supported by the Hungarian National Foundation for Scientific Research Grant No. K75566 and NK 104208.

References

1. L. Aszalós, A. Huszti, *Payment Approval for PayWord*, D. H. Lee, M. Yung (Eds.): Information Security Applications - 13th International Workshop (WISA) 2012, Lecture Notes in Computer Science 7690, (2012), 161–176, Springer-Verlag.
2. Dan Boneh and Matthew Franklin, *Identity based encryption from the Weil pairing*, Advances in Cryptography - Proceedings of Crypto 2001, Vol 2139 of LNCS, (2001), 213–229.
3. B. Blanchet, B. Smyth, *ProVerif 1.85:Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*, http://www.proverif.ens.fr/manual.pdf, (2011)
4. A. Boldyreva, *Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme*, International Workshop on Theory and Practice in Public Key Cryptography(PKC) 2003 Proceedings, Vol 2567 of LNCS, (2003), 31–46.
5. D. Boneh, B. Lynn and H. Shacham, *Short signatures from the Weil pairing*, Advances in Cryptology ASIACRYPT 2001, Vol 2248 of LNCS, (2001), 514532. Full version: Journal of Cryptology, 17 (2004), 297319.
6. G. Frey, H. Ruck, *A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves*, Mathematics of Computation 62 (1994), 865–874.
7. S. Hohenberger, *Advances in Signatures, Encryption, and E-Cash from Bilinear Groups*, PhD Dissertation, 2006.
8. M. Hosseinkhani, E. Tarameshloo, M. Shajari, *AMVPayword: Secure and Efficient Anonymous Payword-Based Micropayment Scheme* International Conference on Computational Intelligence and Security (CIS), (2010), 551–555.
9. A. Huszti, *Multi-Vendor PayWord with Payment Approval*, Proceedings of the 2013 International Conference on Security and Management, CSREA Press, (2013), 265–271.
10. A. Joux *A One Round Protocol for Tripartic Diffie-Hellman*, Proceedings of 4th International Symposium of Algorithmic Number Theory, Vol 1838 of LNCS, (2000), 385–394.
11. A. Menezes, T. Okamoto, S. Vanstone, *Reducing Elliptic Curve Logarithms to Logarithms in a finite field*, IEEE Transaction of Information Theory, Vol 39 (1993), 1639–1646.
12. M. Payeras-Capella, J. L. Ferrer-Gomila, L. Huguet-Rotger, *An efficient anonymous scheme for secure micropayments*, Web Engineering, Lecture Notes in Computer Science 2722, (2003), 80–83, Springer-Verlag.
13. R. Rivest, A. Shamir, *PayWord and MicroMint: Two simple micropayment schemes*, Security Protocols, (1997), 69–87.
14. M. D. Ryan, B. Smyth, *Applied pi calculus*, Formal Models and Techniques for Analyzing Security Protocols, (2011), chapter 6.
15. C.T. Wang, C.C. Chang, C.H. Lin, *A new micro-payment system using general payword chain*, Electronic Commerce Research, 2, (2002), 159–168.
16. H. Wang, J. Ma, J. Sun, *Micro-payment protocol based on multiple hash chains* Second International Symposium on Electronic Commerce and Security, 1, (2009), 71–74.
17. Weidong Kou, *Payment Technologies for E-Commerce*, Springer, (1998).