# Secure Micropayment Using Modified Trapdoor Function

Dr. Mohammed Abdul Waheed[1], Ms. Yamini Bhande[2]

*[1]Associate Professor, Department of Computer Science & Engineering, VTU Centre for PG Studies, Kalaburagi*

*[2]PG Student, Department of Computer Science & Engineering, VTU Centre for PG Studies, Kalaburagi*

**Abstract:** Micropayment have gradually become an important issue nowadays because of the popularity and importance's in e-commerce. Security and convenience related topics are the most important issues that concern customers. However, many research on micropayment have been done which are computational intensive, in-order to reduce the computational cost and increase security, we have described a micro-payment system based on modified trapdoor hash functions. As hash function computations are cheap and signature verifications are only moderately expensive. In order to compute trapdoor hash function, our system require only the use of integer multiplication and addition thus minimize the computation cost further**.**

**Keywords:** Chameleon functions, Trapdoor hash function, Hash function, Electronic payment.

## 1. Introduction

Micropayment schemes have gained rapid attention recently, mostly due to the fact that these schemes exhibit the potential of being embedded in internet based portable computing device. As a special type of electronic payments, micropayment schemes allow a customer to pay a vendor a sequence of small payments over the computer network in exchange for services or electronic products from the vendor. Possible practical applications of micropayment model include digital newspaper, on-line journal subscription, on-line database query, multimedia entertainment over the Internet and Internet advertisement. In addition, accounting and pricing for Internet services and mobile telecommunication may represent yet another set of promising applications of micropayments. Evidently, in such a scenario of payment, all the following costs should be minimized:

1. **Computational cost**: This cost should be comparable with the value to be paid. Therefore, the invocation of public key computation should be prevented or at least be kept as limited to large amount of payment as possible.
2. **Storage cost:** Since there will be a large amount of payment to be handled and each of which is of a tiny value, it is not feasible to keep a record of each payment. This will probably make the cost of processing the payment exceeding the value of the transaction.
3. **Administrative cost**: This includes the minimization of interactions with the trusted third party, usually the bank, the frequency of doing withdrawal and deposit.

In addition to low cost, security requirements are also essential for protecting the integrity of transaction messages. Therefore, we are focused on designing secure modified trapdoor based micropayment for portable computing device which have limited computing resources, e.g., a small amount of memory space, a relatively slow CPU, and a short life span of batteries.

## 2. Literature Review

Public key digital signature schemes have been widely used in electronic payment schemes. However, such schemes are computationally expensive. Therefore, it may not be practical to request customer to sign each and every payment with a public key signature scheme. Recently many schemes based on hash functions has been proposed which generate a one-way hash chain. However, such one-way hash chain has been recognized widely by researchers ever since Lamport first proposed its use in one-time passwords.

Santosh Chandrasekhar, Mukesh Singhal, proposed Multi-trapdoor hash functions and their applications in network security, that allows multiple entities to compute a collision with a given hash value. It describes preliminary designs of aggregate signatures and sanitizable signatures with support for multiple sanitizers. It also describes discrete log-based instantiation of the proposed concept using a novel key-exposure free trapdoor hash function.

Wanget al, proposed a novel payment system with smart mobile devices, wherein customers are not limited to purchase e-cash with the fixed face-value. The amount of every transaction is reduced directly from the customer's account, eliminating the inconvenience of fixed face-value of the e-cash and thus reducing bank online computation cost. Using a technique of trapdoor hash function to mitigate the computational cost, our system can be used with the mobile devices effectively.

Fuw-Yi Yang, Chaoyang , proposed Efficient Trapdoor Hash Functions for Digital Signatures, that presents new trapdoor hash functions with promotion in offline computation, online computation and the size of trapdoor key. By appending some bits to the signature, a trapdoor hash function converts any signature into secure signature scheme with very efficient online computation

Chitra Kiran N, India Dr. G. Narendra Kumar, proposed Implication of Secure Micropayment System Using Process Oriented Structural Design by Hash chaining in Mobile Network, in which author designs the security process using hash chain and Simple Public Key Infrastructure to be implemented on newly designed digital agreement of broker along with paving new secure routing for secure m-transaction as an efficient alternative for digital coin. The system consists of high end encryption using hash function and is independent of any Trusted Third Party when the network topology frequency changes, thereby it is flexible, lightweight, and reliable for secure micropayment systems.

Jian-Sen Wang, Fuw-Yi Yang, and Incheon Paik, proposed A Novel E-cash Payment Protocol Using Trapdoor Hash Function on Smart Mobile Devices, proposes a novel payment system with smart mobile devices, where by customers is not limited to purchase e-cash with fixed face-value. The amount of every transaction is deducted directly from the customer's account, eliminating the inconvenience of fixed face-value of the e-cash, and reducing online computation cost of a bank. Using a technique of trapdoor hash function to mitigate the computational cost, the system is considered to be used with the mobile devices effectively

Ronald L. Rivest and Adi Shamir, proposed PayWord and MicroMint: Two Simple Micropayment Schemes, in which, "PayWord," is a credit-based scheme, based on chains of "paywords". The user authenticates to the vendor with a single public-key signature and thus successively reveals each payword in the chain to the vendor to make micropayments. The incremental cost of a payment is thus one hash function computation per party. PayWord is optimized for sequences of micropayments. It is also secure and flexible enough to support larger variable-size payments as well. "MicroMint" was designed to eliminate public-key operations altogether. It has reduced security but higher speed. It introduces a new type of coins by k-way hash-function collision. Just as for a real mint, a broker's "economy of scale" allows customer to produce large quantities of such coins at very low cost, while such small-scale forgery attempts can only produce coins at a cost exceeding their value.

## 3. Trapdoor Hash Function

Hash functions are commonly applied to digital signature techniques, and digital signature algorithms can be broken down into three phases: key generation, signature generation and signature verification. Collision-resistance is one of the main features of traditional hash functions. For Chameleon functions or trapdoor hash functions, the feature of collision-resistance is optional; the owner of a trapdoor key can easily find other collided pre-images and produce the same hash value. For instance, assuming TH() represents trapdoor hash function and the hash value $TH(h_1)$, after knowing $h_1$, the owner of a trapdoor key can then calculate $h_2$; hence, $TH(h_2) = TH(h_1)$. Computing the value of Chameleon functions online requires a multiplication and modulo operation.

Let p, q, P and Q be large prime numbers, the number $n = P.Q$;$P = 2.p +1$; $Q = 2. q+1$ with |P|, |Q| and |p| represent the encoded bit length of P, Q and p. Their lengths can be chosen as follows: $|P| = |Q| = 512$; $|p| = 160$. The order of $g$ □ $Z^*_n$ is p. Randomly selecting x □$_R$ $(0,1)^l$ that is, element **x** is randomly selected from the set $(0,1)^l$.; Calculating $y = g^x \bmod n$. The trapdoor key is $TK = 2^{-1} \bmod \lambda(n)$ and the public key is $HK = (g, n, y)$. Also defines a collision resistant hash function $H_1$: $\{0, 1\}* \rightarrow \{0, 1\}^l$ where l is the security parameter. If a message $m_1$ □ $_R$ $(0,1)^l$ and $r_1$ □ $_R$ $\{0,1\}^{2\ l+k}$, the hash operation to compute hash value of the message $m_1$ is defined as: $TH_{HK}(m_1, r_1) = g^{\ m1+r1} \bmod n$. Assume that a signer has determined to sign a target message $m_2$ □ $(0,1)^l$. Then the signer chooses to store triple $(m_1, r_1, s)$ and performs the trapdoor operation which is defined as $TH_{TK}(m_1, r_1, m_2) = r_2 = 2^k(m_1 - m_2) + r_1 \bmod \lambda\ (n)$; where $k = |\ \lambda\ (n)|$ is the bit length of the trapdoor key $TK = \lambda\ (n)$.

## 4. Modified Trapdoor Function

If we change position of concatenation order of *r* and *m*, then the new definition of hash operation is $TH_{HK}(m_1, r_1) = g^{r1+m1} \bmod n$ and the new trapdoor operation would be $TH_{TK}(m_1, r_1, m_2) = r_2 = 2^{-1}((m_1 - m_2) + 2^l r_1) = 2^{-1}(m_1 - m_2) + r_1 = x(m_1 - m_2) + r_1$; where $x = 2^{-1} \bmod_{\lambda}(n)$.

Note that the modular reduction in the original trapdoor operation disappears. The first papers in which online signature computations save the cost of modular reduction are Girault at Eurocrypt'91 and Poupard-Stern at Eurocrypt'98. The quantity $r_2$ is computed using integer arithmetic. The bit length of $r_2$ may vary in a wide range because that $r_2$ is a result of integer arithmetic. Therefore, the new definition is forced to switch the position of *r* and *m*.

- OFFLINE HASH OPERATION: Signer chooses at random a pair ($m_1$, $r_1$) $\in_R$ {0, 1} $^l$ × Zλ(n) and performs hash operation on this pair, i.e., $TH_{HK}(m_1, r_1) = g^{r1+m1}$ mod n where $m_1 \in_R$ {0, 1} $^l$ , $r_1$ ∈R {0, 1} $^{k+1}$ , k = |λ(n)|. Then using signer's secret key ***d***, a RSA signature ***s*** is generated, s = $H(TH_{HK}(m_1, r_1))$ d mod N. The signer stores the triple (m1, $r_1$, s) in storage. Later signer can begin a trapdoor operation to obtain $m_2$ and $r_2$ such that $V=TH_{HK}(m_1, r_1) =TH_{HK}(m_2, r_2)$. Therefore, all the heavy computation are calculated in offline.
- ONLINE TRAPDOOR OPERATION: Upon receiving the message M, the signer performs the trapdoor operation on ($m_1$, $r_1$) and $m_2 = H_1(M)$; namely $TH_{TK}(m_1, r_1, m2) = r2 = x(m_1 - m_2) + r_1$, where ($m_1$, $m_2$, $r_1$) $\in_R$ $\{0, 1\}^l \times \{0, 1\}^l \times \{0, 1\}^{k+1}$ . Then the signed message on M is the tuple (M, $r_2$, s). Note that in addition to the message M and signature s, an appended string $r_2$ is added to the signed message.
- VERIFICATION**:** The signed message is verified by checking that $s^e = H(TH_{HK}(m_2, r_2))$ mod N, where $m_2 = H_1(M)$ .i.e $TH_{HK}(m_2, r_2) = TH_{HK}(m_1, r_1)$.

**Definition:** A secure trapdoor hash function has three properties:
1) EFFICIENCY**:** Given hash key HK and (m, r) ∈ $\{0, 1\}^l$ × Z λ(n), the hash value $TH_{HK}(m, r)$ is computable in polynomial time.
2) COLLISION RESISTANT: Given hash key HK, there exists no probabilistic polynomial time algorithm outputs two pairs ($m_1$, $r_1$) and ($m_2$, $r_2$) producing the same hash value with non-negligible probability.
3) TRAPDOOR COLLISIONS: Given trapdoor key and a triple ($m_1$, $m_2$, $r_1$) ∈ $\{0, 1\}^l \times \{0, 1\}^l \times Z\lambda(n)$, there exists a probabilistic polynomial time algorithm outputs value $r_2$ ∈ Zλ(n) and satisfies $TH_{HK}(m_1, r_1) = TH_{HK}(m_2, r_2)$. Further, if $r_1$ is uniformly distributed over its domain then $r_1$ and $r_2$ have statistically indistinguishable distribution in the same domain.

## 5. Entity In Micropayment

1. **Customer**: Requests products or services from vendor.
2. **Vendor:** Provides product or service to customer according to their request.
3. **Bank:** Issues the certificate and acts as a CA (Certificate Authority)

## 6. Micropayment Procedure

1. Customer register with bank by sending virtual identity message $ID_i$ along with the trapdoor hash value A. Receive endorsement data consisting of random value, banker's name, customer virtual id, expiration date and other information secured by bank private keys.
2. When the customer wishes to purchase products, customer sends signed encrypted purchase request to the vendor.
3. Vendor accepts the purchase order and verify request with his public key.
4. Vendor then send reply by signed encrypted transaction.
5. Customer after receiving order, send request for the bank to complete transaction. Bank in turn, decrypt the request and verify the customer details and may proceed for further action.

### 6.1 Parameters and Symbols

The parameters and symbols used in micropayment system can be categorized as bank parameters, customer parameters and vendor parameters. They are described as follows:

| Parameters | Symbols | Meaning |
|---|---|---|
| Bank | $ID_{BK}$ | It signifies the identity of a bank. |
| | $Enc_{BK}()$,$Dec_{BK}()$ | They are the encryption and decryption functions of the bank, respectively. |
| Customer | $ID_i$ | It signifies the identity of a customer. |
| Vendor | $ID_v$ | It signifies the virtual identity of vendor. |
| | $TD_i$ | It stands for the name and price of the purchased Product |
| | TS | It stands for the transaction stamp including transaction time, date, and serial number |
| | $Sig_v()$ | Vendor uses signing key to sign the message and generates signature. |
| | $Ver_v()$ | Customer or bank verifies the signature with the vendor's public verification key. $Ver_v()$, returns either string "true" or " false". |
| | $Enc_v()$, $Dec_v()$ | Encryption/decryption functions of the vendor |

The micropayment system can be divided into three phases: customer registration phase, payment phase, and bank redemption phase. The followings describe the details of each phase.

**6.2 Customer Registration**

In this phase, the customer has to register with bank and apply for certificate σ, by sending trapdoor hash value and virtual identity message $ID_i$. The process is as follows.

1. Initially customer randomly generate message $m_1 \square_R \{0,1\}^{l}$ and number $r_1 \square_R \{0,1\}^{2l+k}$.
2. Calculate the trapdoor hash value $A = TH_{HK}(m_1, r_1) = g^{r1}y_1^{m1}$ mod n.
3. Send virtual identity $ID_i$ and the trapdoor hash value A to the bank.

The customer stores the random pair $(m_1,r_1)$. This pair will enable customer to calculate another pair $(m_2, r_2)$ in the payment phase such that both pairs generate the same trapdoor hash value. After receiving the registration message, the bank uses its private key **d** to sign the message and generate certificate σ. The bank then sends certificate σ to the customer. The process is as follows.

1. Calculate $\sigma = H(ID_i ,A)^d$ mod n.
2. Send the certificate σ to the customer.

After receiving σ, the customer verifies if **s** is a valid signature on the registration message signed by the bank; that is, check $\sigma^{e}$ ?= $H(ID_i,A)$ mod n; where e public verification key, e∈R Zn,de = 1 mod (P−1)(Q−1).

**6.3 Customer Payment Phase**

When a customer wishes to purchase product or service, he first sends requests for product or service details. After learning the name and price details, the customer creates a purchase order with information including: customer virtual identity, certificate, trapdoor hash value, and name and price of product. The customer encrypts the order, that is, □ = $Enc_v(\sigma, ID_i, A, TD_i)$ and sends it to the vendor. After receiving the encrypted order α, the vendor uses his decryption key to obtain the order information and verify commitment. The vendor then sign the order and send it to the customer. The customer then encrypts the payment order and send it to the bank to complete the transaction payment. The process is as follows.

1. Vendor use the private key to decrypt the message $(\sigma, ID_i, A, TD_i) = Dec_v(\alpha)$.
2. Then uses the bank's public key to verify the certificate, to detect $\sigma^e$ ?= $H(ID_i, A)$ mod n.
3. The vendor adds the transaction timestamp TS to the payment order and sign the quotation $s = Sig_v(\sigma$, IDi, A, $TD_i$, TS). He then sends the transaction timestamp TS and signature **s** to the customer. The customer verifies the validity of the quotation signed by the vendor; that is, check $Ver_v(s)$ ?="true".
4. Customer then construct the document $m_2 = (ID_i, ID_v, ID_{BK}, TD_i, TS)$.
5. The customer retrieves the stored information $(m_1, r_1)$ and select a new trapdoor key $x_2 \square_R \{0,1\}^l$ and calculate $y_2 = g^{x2}$ mod n. Then execute the trapdoor operation $TH_{TK}(m_2) = r_2 = r_1+x_1m_1-x_2m_2$. For $r_1+x_1m_1 = r_2+x_2m_2$, it implies that $A = g^{r1}y_2^{\ m1} = g^{r2}y_2^{\ m2}$ mod n.
6. The Customer then encrypts $(\sigma, ID_i, ID_v, ID_{BK}, TD_i$, TS, s, $r_2$, $y_2)$ payment information and sends it to the bank, to complete payment transaction.

**6.4 The Bank Redemption Phase**

After receiving conformation, the bank uses private key to decrypt the message and obtain payment information $(\sigma, ID_i, ID_v, ID_{BK}, TD_i$,TS s, $r_2$,$y_2)$.The bank then verifies the customer's account balance ,customer's certificate and order information. The redemption and verification process are as follows:

1. Produce the document $m_2 = (ID_i, ID_v, ID_{BK}, TD_i,TS)$.
2. Calculate the hash value $A = TH_{HK}(m_2, r_2) = g^{r2}y_2^{m2}$ mod n.
3. Verify the legality certificate; that is $\sigma^e$ ?= $H(ID_i;A)$ mod n.
4. Verify the vendor's quote $Ver_v(s)$ ?="true".

After verifying the signature of vendor, the bank accepts the payment information and wait for customer conformation. Upon receiving conformation, bank saves the transaction details $(\sigma, ID_i, ID_v, ID_{BK}$,$TD_i$,TS, s, $r_2$,$y_2)$ in its database and transfers the amount to the vendor's account.

## 7. Conclusion.

With the rapid development of the Internet and information technology, e-commerce system is flourishing. Unfortunately, current micropayment systems have proven to be computational intensive. Therefore, to reduce computation cost and increase security. We have reviewed hash based micropayment system and proposes a modified trapdoor hash function with very efficient in online computation. The on-line computation is showed to be more efficient than scheme *TH*. The proposed scheme seems efficient for low battery-powered computing devices because no more operation of modular reduction is required in online phase.

In particular, we are investigating the possibility to allow trapdoor hash function support multiple vendor while maintaining the same level of security and usability.

## References

[1] R.L. Rivest, Electronic lottery tickets as micropayments," *Proc. of Financial Cryptography Conference, FC '97*, Lecture Notes in Computer Science, Vol.1318,Springer Verlag, pp.307,314, 1998.

[2] R.L. Rivest and A. Shamir, PayWord and MicroMint: Two simple micropayment schemes," *Proc. of Security Protocols Workshop*, Lecture Notes in Computer Science, Vol.1189, Springer Verlag, pp.69,87, 1997.

[3] H. Krawczyk and T. Rabin, “Chameleon signatures”, Symposium on Network and Distributed Systems Security (NDSS’00), pp.143-154, 2000.

[4] Shamir and Y. Tauman, “Improved online / offline signature schemes”, Advances in CryptologyCRYPTO’01, LNCS 2139, pp. 355-367, 2001.

[5] F. Y. Yang, S. H. Chiu, and C. M. Liao, “Trapdoor Hash Functions with Efficient Online Computations”, The Proceedings of Multimedia and Networking Systems Conference 2006 (MNSC 2006), 2006.

[6] F. Y. Yang, “Efficient trapdoor hash function for digital signatures”, Chaoyang Journal, Vol. 12, pp. 351- 357, 2007.

[7] F. Y. Yang, “Improvement on a trapdoor hash function”, International Journal of Network Security, Vol. 9, No. 1, July, pp. 17-21, 2009.

[8] C. C. Chang and Y. P. Lai, “A flexible Date-attachment Scheme on E-cash”, Computers & Security, Vol. 22, No. 2, pp.160-166, 2003.

[9] Santosh Chandrasekhar; Mukesh Singhal,“ Multi-trapdoor hash functions and their applications in network security “IEEE Conference on Communications and Network Security, 2014.

[10] Chitra Kiran N, India Dr. G. Narendra Kumar ,”Implication of Secure Micropayment System Using Process Oriented Structural Design by Hash chaining in Mobile Network “International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, January 2012 .

[11] Jian-Sen Wang , Fuw-Yi Yang, and Incheon Paik, “A Novel E-cash Payment Protocol Using Trapdoor Hash Function on Smart Mobile Devices “ International Journal of Computer Science and Network Security, VOL.11 No.6, June 2011.