A SECURE ELECTRONIC PAYMENT PROTOCOL FOR WIRELESS MESH NETWORKS

Helen Cheung and Cungang Yang

Department of Electrical and Computer Engineering
Ryerson University

ABSTRACT

Electronic payment systems for wireless mesh networks need to take into account the limited computational and communicational ability of mesh clients. Micropayment scheme is well suited for this scenario since it is specifically designed for efficient operations in payment transactions. In this article, we propose a one way hash chain structure based on which efficient and secure payment protocols that support both prepaid and credit-based paying schemes are introduced.

KEYWORDS

multiparty micropayment, payment certificate, hash chain, wireless mesh networks, electronic payment

1. INTRODUCTION

There are a few payment models proposed in the literature [5] [6], which can be classified into two categories: the traditional payment model and the micropayment model. The example traditional payment model include the credit card platforms [7] [8] and electric cash platform [9]. The traditional payment models allow only one payment in a transaction, which has been widely adopted for the electric payment applications. These protocols will be too expensive and time-consuming when applied to inexpensive transactions because of the transaction charges of card companies and the computational cost of public-key signature verification. They also place a heavy burden on the computational and storage capabilities of currently available wireless devices. Micropayment models are designed to allow frequent transfer of very small amounts, perhaps less than a cent, in a single transaction, which is considered more efficient than the traditional payment model. The micropayment models are often adopted for mobile and wireless network applications [3][4]. In this paper, we focus on micropayment schemes because this category not only directly addresses the limited resources of mobile communications but also is the most reasonable option for applying to the light-weight payment scheme by mesh clients in wireless mesh networks. The following requirements should be addressed when designing a suitable payment mechanism for mesh networks. First, customers expect a robust, secure, and fair payment mechanism which can be applied in different wireless networks. Second, the payment mechanism should be light-weight (i.e. with low computational complexity and low communication overhead) so that it can be easier run on mobile devices. Third, user anonymity should be achieved. Finally, a payment mechanism should be of low implementation cost.

In this paper, we integrate a new one-way hash chain and the roaming technology to develop novel payment schemes for mesh networks. The main goal is to minimize the number of public-key operations required per payment, using hash operations instead whenever possible. As a

DOI : 10.5121/ijnsa.2014.6501

rough guide, hash functions are about 100 times faster than RSA signature verification, and about 10,000 times faster than RSA signature generation. The contributions of this work are summarized as follows: First, the ticket-base approach supporting authentication and secure billing functionalities makes these two fundamental security operations in mesh networks more efficient. Second, it proposes a novice user-user payment scheme which provides credits so as to encourage MCs (mesh clients) to relay packets for other MCs. Third, the payment schemes support intra-domain roaming. The commutation and communication cost of the billing on intra-domain roaming is more efficient than the cost when a MC logs in.

The remainder of this paper is organized as follow: In section 2, we introduce the effective one-way hash chain. In section 3, we study ticket-based electronic payment protocol in more details. Security analysis of the proposed payment protocol is explained in section 4. Section 5 demonstrates the performance analysis. Section 6 discuss the related works. The paper is concluded in section 7.

2. EFFICIENT ONE-WAY HASH CHAIN

Hash values from a user-generated hash chain can be used as authenticated payment tokens. A one-way chain (V_0 … V_N) is a collection of values such that each value V_i (except the last value V_N) is a one-way function of the next value V_{i+1}. In particular, we have that $V_i = H(V_{i+1})$, for $0 \leq i < N$. Here, H is a one-way function, and is often selected as a cryptographic hash function. A drawback of traditional one-way chains is that the verifier has to perform j-i operations to validate V_j given V_i, which can be expensive if j-i is large. This weakness is solved by the hierarchical one-way chain that is more efficient.

A hierarchical one-way chain consists of two or more levels of chains, where values of a first-level ("primary") chain act as roots of a set of second-level ("secondary") chains[1]. We refer to the secondary chain rooted in the *i*th value of the primary chain as the *i*th secondary chain. Here, all the values of the *i*th secondary chain are released before any of the values of the i + 1st chain is released; the primary chain value V_i is released in between. In a hierarchical one-way chain, all end-values need to be authenticated – both that of the primary chain and those of all secondary chains. The drawback of the hierarchical one-way chain is that the loss of the end value of a secondary hash chain prevents the verifier to authenticate secondary chain values until the next value of the primary chain is disclosed.

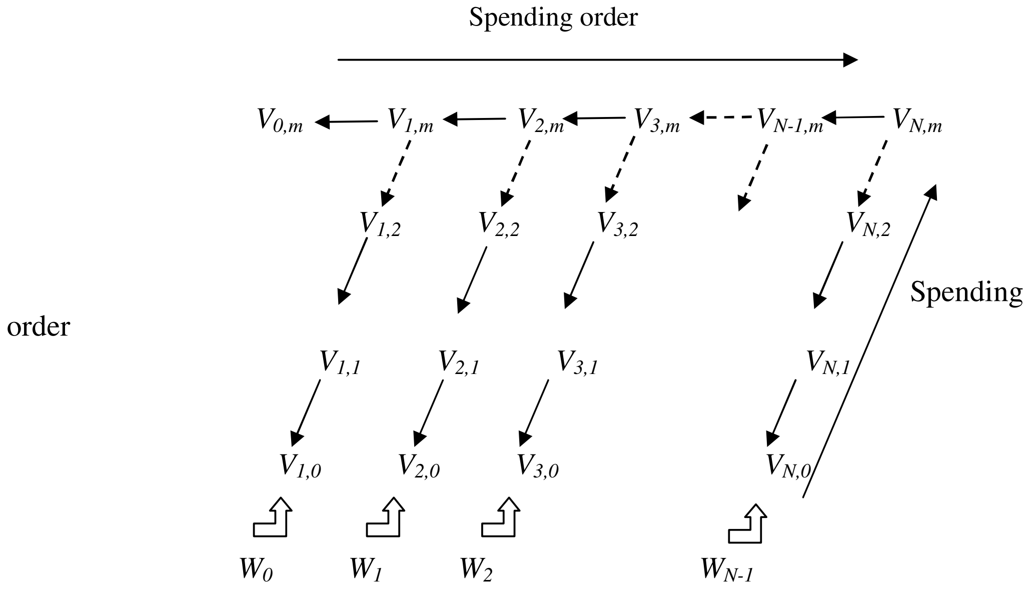


Figure 1. Efficient One-Way Hash Chain

We propose a secure and efficient hierarchy one way chain (see Figure 1) where $V_{N,m}$ is private information and m, N, $V_{0,\ m}$ and W_0 W_1 W_2 ...W_{N-1} are public information. The hash chain is generated by broker and is assigned to MCs or MRs (mesh routers). Before the start of the transactions, MC/MR sends the public information m, N, $V_{0,\ m}$ and W_0 W_1 W_2 ...W_{N-1} to MR/MC. Payment token are $m+1$ elements of the secondary chains. MR/MC authenticates $V_{1,0}$ using $W_0 = h(V_{1,\ 0} \| V_{0,\ m})$. This authentication ensures the correctness of the token values from the same hierarchy hash chain. For the following end values of secondary chain, MR/MC can authenticate them as follows:

$$W_i = h(V_{i+1,0} \| V_{i,m}) \qquad (i \geq 0)$$

where W_i and $V_{0,m}$ are public information and MC/MR may retransmits them periodically. $V_{n,m}$ is private information only hold secretly by MC/MR.

The proposed hash chain efficiently authenticates the end-values of the secondary chain at any moment, without assuming any additional authentication protocols. In the meantime, it does not have the problem as the approach proposed by Liu and Ning [1].

If a hash chain is not used up, MCs could still use its remaining tokens when roams to a different MR of the same mesh domain. MCs need to notify new MR the most recent hash value used and its index in the hash chain.

The advantages of the proposed one-way hash chain are as follows: (1) it voids the long chain of the one-way hash chain. The hierarchy one way chain allows MC/MR to reduce storage. (2) it does not need a protocol to authenticate the end values of the hash chain. (3) it is efficient: a MC/MR who received the authentic values such as $V_{0,\ m}$ and W_0 W_1 W_2 ...W_{N-1} can efficiently authenticate secondary chain in hash chain generated by $V_{i,0}$ This approach substantially reduces the verification overhead for a new MR/MC that needs to catch up to current value of the chain.

3. TICKET-BASED ELECTRONIC PAYMENT PROTOCOL

A ticket purchased by the MCs from a broker includes the information of the hierarchy hash chain. The tokens must be spent through its associated MR, who prevents cheating by the MC. Cheating by the MR itself will be detected after the fact. The payment tokens MR collected can be efficiently redeemed from the brokers. Unspent tokens can be spent on a different MR to access a different destination, but utilizing the MRs in the same mesh domain to prevent double spending. If further accesses are not made, tokens may later be refunded by the issuing broker.

The goal of this research is to design ticket-based protocols that support both secure mutual authentication and billing. After MC and MR mutually authenticate with each other[2], MCs access the mesh network and pay for the services through the tokens of the hierarchy hash chain approach.

The payment protocols support two types of paying schemes: directly buy tokens of hash chain with the pre-paid scheme and pay later with credit-based scheme. A MC/MR can apply either or both types of paying schemes from the broker. Normally, a ticket only can be used for one mesh domain. The billing server of the domain is in charge of controlling the credit limit or the balance of tickets to avoid MC using a hash token chain more than one time for pre-paid scheme or spent beyond the credit limit of the credit-based scheme.

- Pre-paid: MC/MR purchase hash chains from brokers before accessing the mesh network. The chain related information will be added to the MC/MR's ticket. Broker may issue multiple pre-paid tickets to a MC.
- Credit-based: the broker assigns a credit limit for MCs. The broker determines the credit limit for each MC ticket. Broker may issue multiple credit-based tickets to a MC. However, the total remaining credit limits of all a MC's tickets must not greater than his/her credit limit. The broker also determines a credit limit for MRs in a mesh domain. Credit-related information will be added to MR/MC's ticket.

For the credit-based scheme, MC/MR pay bills to his/her broker, the broker continues updating the credit balance of the MC/MR. If the MC/MR has available credit, the broker can generate new ticket for him/her.

MC can use the same ticket to roam different MRs in the same domain. A MR will inform the billing server the most recent balance of a MC's ticket when the MC roams out of its covering area. The billing server of the mesh domain will update the balance of MC's ticket afterwards. Broker will also issue multiple credit-base tickets to MR if the balance of the tickets is within the limits of the domain's credit.

Prepaid and credit-based payment approaches need the support of mutual authentication. The MR authenticates MCs to ensure that the MC is trusted. MC authenticates MR to prevent bogus MRs asking higher unit token fee to MCs. This mutual authentication is based on the tickets signed by brokers [2].

A ticket includes the information of credit limit or the balance of MC/MR. The billing server always checks the balance or credit limit, if the balance or credit is beyond the limit, the billing server will notify the MR to cut the MC's service.

In the design of the protocols, the beacon message of a MR should include the information of its service fee and relaying fee. MCs need to know this public information before roaming to a new MR or agreeing to provide the relaying service for other MCs.

We will discuss the ticket-based payment approach for the following two different cases:

(1) Multiple hops between MC and its associated MR
The multiple hops case describes the situation that a MC is at an arbitrary number of hops away from the MR. The MR that MC associated is defined as the associate MR which is in charge of authenticating MC, implementing billing protocol with MC and sending the billing update information to billing server.

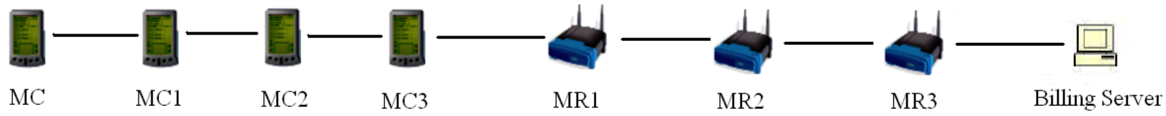


Figure 2. Payment Scheme for Multiple Hops between a MC and its Associated MR

In Figure 2, suppose MC is associated with MR1 and both MC and MR1 have bought tickets separately from their brokers, the procedure of the billing scheme for multiple hops is as follows:

- MC pays MR1 with the tokens which value is the addition of service fee for MR1 and relaying fee for MC1, MC2 and MC3
- MR1 pays tokens for relaying fees of MC1, MC2 and MC3 on behalf of MC
- MR1 collects the tokens from MC and submit them to the billing server
- Billing server get the fund from MC's broker
- MC1, MC2 and MC3 collects the tokens from MR1 and get the fund from MR1's broker

(2) One hop between a MC and its associated MR

It is a special case of multiple hops. If a MC can reach the MR directly, the general procedure of the billing process is as follows:

- MC pays tokens to MR1. MR1 collects the tokens and submits to the billing server
- Billing server get the fund from MC's broker

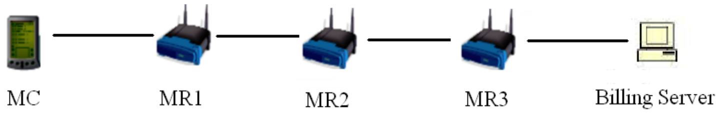


Figure 3. Payment Scheme for One Hop between a MC and MR1

3.1 Payment Hash Chain

Tokens are fundamental component of the payment hash chain. In the prepaid scheme, MC/MR needs to pay first to purchase payment hash chain from brokers. Brokers also need to assign payment hash chains for MC/MR in the credit-based approach and send bill to redeem from MR/MC later.

To avoid double spend of a hash chain, each chain only can be used for one mesh domain. MC needs to prepare different chains for different domains. In the meantime, a MC/MR could also hold multiple hash chains for a single domain.

Two types of payment hash chains are designed: one for MCs and the other is for MRs. MC's hash chain pays tokens to MR it associates. MR's hash chain pays relaying MCs on the route.

The format of purchase hash chain for MC:

P_{MC}: ID_{pmc}, $W_0\ W_1\ W_2 \ldots W_{N-1}$, $V_{0,m}$, m, N, MC, Date, Domain, Value, Ticket Credit
ID_{pmc}: id of the hash chain
***$W_0\ W_1\ W_2 \ldots W_{N-1}$, $V_{0,m}$*, m, N:** public information of hierarchical hash chain
Domain: the only domain that the hash chain could be used.
MC: id of the MC
Date: the expiry date that the ticket is to be expired
Value/Credit: the purchased value or credit limit of the chain

Billing server keeps the record of the P_{MC} and continues to update the balance/credit of the MC whenever he/she logs in or roams in the same mesh domain. If there is no balance or credit, the billing server requests MC to provide a new ticket or ask MR to directly cut off the service. Each payment hash chain only can be used in a specific domain. The payment chain of MC is added to its ticket which will be signed by the MC's broker.

The format of purchase hash chain for MR:

P_{MR}: ID_{pmr}, $W_0\ W_1\ W_2 \ldots W_{N-1}$, $V_{0,m}$, m, N, Domain, MR, Credit/Value, Date
ID_{pmr}: id of the hash chain
***$W_0\ W_1\ W_2 \ldots W_{N-1}$, $V_{0,m}$*, m, N:** public information of hierarchical hash chain
Domain: the only domain that the hash chain could be used.
MR: id of the MR
Date: the date that the ticket is to be expired
Value/Credit: the purchased value or credit limit of the chain

Purchase hash chain is a part of MR's ticket [2] which will be signed by the broker.

3.2 Billing Protocol

3.2.1 Login Billing Protocol for Multiple Hops

If a MC cannot directly reach MR, their communication could be relayed by other MCs. To encourage MCs to support packet relaying for others, a relaying fee is provided as a bonus. MR prepares a contract in which each relaying MC need to add its MAC. These MAC values prove that these MCs are on the route of the traffic and have provided the relaying service and therefore should receive the bonus. Moreover, MC prepares a contract for its associated MR. MC's contract proves to broker that MR provides service and therefore redeem the tokens of the MC.

The contract issued by MC:

MC, MR, $P_{start,}$ Unit Fee, $P_{MC,}$ $Broker_{id}$

MC: ID of MC
MR: ID of MR
Unit fee: unit fee for each token
P_{start}: start index of the MR's hash chain
$P_{MC:}$ payment chain

Broker$_{id}$: ID of the broker

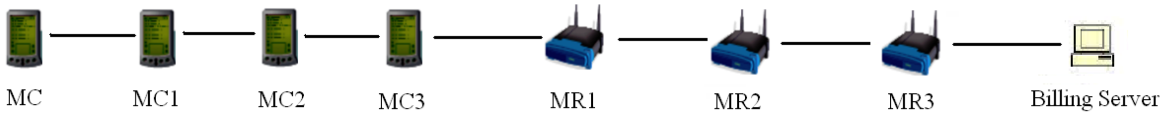
The contract issued by MR:

MR, P$_{start,}$ Fee of Relay, P$_{MR,}$ Broker$_{id}$, MACs
MR: ID of the MR who issues the contract and distributes it to relaying MCs.
P$_{start}$: start index of the MR's hash chain
MACs: MACs generated by relaying MCs and adds to the contract

The detail of the login billing protocol shown in figure 4 is as follows:

- From the beacon message, MC knows the service and relaying fee of its associated MR. MC and MR mutual authenticate each other by exchanging their tickets. MR checks the domain of the purchase hash chain from the MC's ticket and then transfers it to the billing server.
- The billing server checks if the MC has ever visited the domain and the balance or remaining credit of his/her ticket. The billing server will notify MR the index where the hash chain should be started if the payment chain has balance or credit. The billing server records index of MC's hash chain when the last time he/she visits the domain. If the MC is new to the domain, the index value should be the beginning of the MC's hash chain.
- The MR will then inform the MC the unit fee of a token. If MC agrees with this, it generates a contract which includes the unit fee and index information. MC signs the contract and send it back to MR. MR verify the contract and keep it for his future redeem from broker.
- MR generates a contract and distributes it to all relaying MCs on the route. All relaying MCs will add its MAC to the contract. Each MAC key is shared by MC and the broker. MR signs the contract and sends it to all the relaying MCs.
- MC starts and keeps releasing hash tokens from the start index. Each token will be verified by MR. MR also releases tokens from its hash chain that will be verified by all relaying MCs on the route. If the relaying MC cannot receive the token of the MR as the bonus of the relaying service, it will stop to provide the relaying service.

The protocol is briefly shown as follows:



Exchange Ticket with Payment Chain   Check balance/Credit

Unit fee and current hash value and inde x index and current hash value of the hash chain

Temporary ticket and the encrypted MAC
key with MC's public key
unit fee, index and current hash value with MAC value

MC generates contract, sign it. MR1 verify it

MR1 generates the contract and send securely to all relaying MCs
The contract is added with the MAC generated with shared key between neighbours

Each relaying MC verify it and add a MAC (generated with the key shared with broker)

With the same hop-by hop approach, the contract is sent back to MR1

MR1 sign the final contract and distribute to all relaying MCs

The first token is submitted from MC and verified by MR1

MR1 sends first token to relaying MCs and relaying MCs verify the tokens

The second token is submitted from MC and verified by MR1

MR1 sends second token to relaying MCs and relaying MCs verify the tokens

......

The m token is submitted from MC and verified by MR1

MR1 sends m token to relaying MCs and relaying MCs verify the tokens

MC contract, current index and the last hash value of the hash chain used

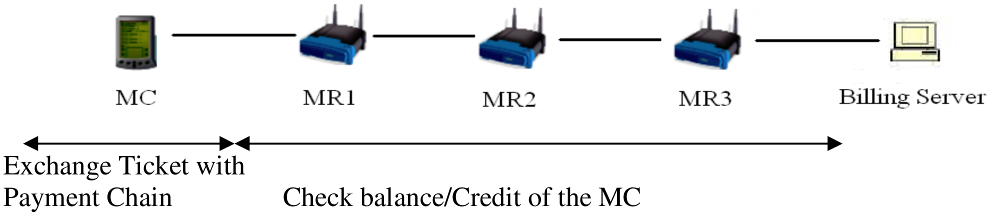
Figure 4. login billing protocol for multiple hops

In the case that the topology changes, for example relaying MCs moves out of the route, MR will ask new joined MCs to sign a new contract with the most recent index of MR's hash chain.

3.2.2 Login Billing Protocol for One-hop

The login protocol for one hop shown in figure 5 is a special case of the protocol of multiple hops. The general procedural of the protocol for one hop is as follows:

- MC and MR mutual authenticate each other by exchanging their tickets. MR verifies the domain of the payment chain in the ticket and transfers the ticket to the billing server.
- The billing server verifies if MC has visited the domain before and checks his/her balance or remaining credit. The billing server will notify MR the index where the hash chain should be started if the payment chain has balance or credit. The billing server records the hash chain information whenever MC roams leaves the domain. If the MC is new to the domain, the index should be the beginning of the hash chain.
- The MR informs the MC his/her balance, unit token fee, and start index of the token. If MC agree with the information from MR, it generates a contract and send to MR1
- MC starts and keeps releasing hash tokens from the start index. Each token will be verified by MR.


MC
MR1
MR2
MR3
Billing Server
Exchange Ticket with Payment Chain
Check balance/Credit of the MC

← ←

Unit fee and current hash value and index
Temporary ticket and the encrypted MAC key with MC's public key
Encrypt unit fee and current hash value and index with the MAC key

index and current hash value of the hash chain

→

MC generates contract, sign it. MR1 verify it

→

The first token is submitted from MC and verified by MR1

→

The second token is submitted from MC and verified by MR1

→

......

→

The m token is submitted from MC and verified by MR1

→

→

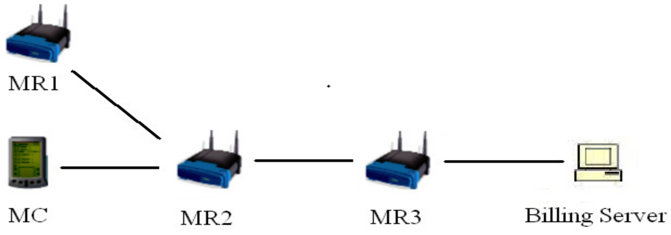
MC contract, current index and the last hash value of the hash chain used

Figure 5. Login Billing Protocol for One-hop

3.3 Billing Issues During Intra-domain Roaming

When a MC roams, for example, from MR1 to MR2 of the same domain in the mesh network, MR1 needs to securely forward the MC's MAC key and the balance/credit of MC to MR2. MC2 monitor the balance of MC and will stop the service once the balance is used up. In the mean time, MR1 submits MC's contract, balance of the contract, the collected tokens from MC and the record of delivered tokens to the billing server. Billing server will verify this message and update the MC's balance. Billing server collects these information for fund redeem from the broker. Since the new balance of MC is securely forwarded from MR1, MR2 does not need to recheck the balance of the MC from the billing server. Also, the index of the current token chain of MC's is also forwarded from MR1 to MR2. For the case of one-hop, only MC1 generates contract. For the case of multiple hops, MR2 will create the contracts according to new start index and the balance forwarded from MR1. If no balance value and contract are forwarded, MR2 will implement the login process as indicated in section 3.2.1.

The Intra-domain Billing Protocol for One-hop shown in figure 6 is as follows:



→

MC show temporary ticket

←

unit fee and index with the MAC key

→

MC generates contract, sign it. MR2 verify it

→

The first token is submitted from MC and verified by MR2

→

The second token is submitted from MC and verified by MR2

→

......

→

The m token is submitted from MC and verified by MR2

→

→

MC contract, current index and the last hash value of the hash chain used

Figure 6. Intra-domain Billing Protocol for One-hop

The Intra-domain Billing Protocol for Multiple Hop shown in figure 7 is as follows:

Temporary Ticket

←

Unit fee, index with MAC

→

MC generates contract, sign it. MR2 verify it

←

MR2 generates the contract and send securely to all relaying MCs
The contract is added with the MAC generated with shared key between neighbours
Each relaying MC verify it and add a MAC (generated with the key shared with broker)

→

With the same hop-by-hop approach, the contract is sent back to MR2

←

MR2 sign the final contract and distribute to all relaying MCs

→

The first token is submitted from MC and verified by MR2

←

MR2 sends first token to relaying MCs and relaying MCs verify the tokens

→

The second token is submitted from MC and verified by MR2

←

MR2 sends second token to relaying MCs and relaying MCs verify the tokens

......

The m token is submitted from MC and verified by MR2

←

MR2 sends m token to relaying MCs and relaying MCs verify the tokens

→

MC contract, current index and the last hash value of the hash chain used

Figure 7. Intra-domain Billing Protocol for Multiple Hop

Comparing with login billing protocols, the intra-domain billing schemes have one more advantage: when perform intra-domain roam, MR2 doesn't need to perform signature verification for authentication and billing procedure. Symmetric key encryption is employed for securely transfer the balance and start index of MC1 between MR1 and MR2.

Since the authentication and the billing information are in the same ticket, very limited expensive signature verifications are required for both authentication and billing procedure between MC and MR. Once the MC passed the authentication and its hash chain balance/credit is available, it will be more efficient for MC to perform intra-domain roam in mesh networks.

3.4 Fee Clearance

MRs always needs to submit MC's contracts and his/her current index to the billing server. Periodically, after collecting this information from all MRs in the same domain, the billing server transfers it to the broker for fund redeeming. The broker verifies MC's signature and ensure that MC should pay the MRs. The amount is calculated according to the number of hashed tokens received and the unit token fee identified in the MC's contract.

On the other hand, contract generated by MR proves which relaying MCs are involved in the relaying process and should receive the relaying fee from the MR. Relaying MC submits the contract the MR signed and the most current index of MR's hash chain. Broker verifies the contract and calculates the fund based on the current index and unit relaying fee described in the MR's contract.

4. SECURITY ANALYSIS

There are a few useful features of the proposed payment scheme including the avoidance of overspending and double spending, the fairness, the user anonymity, and privacy. The proposed scheme meets the security requirements of mesh networks and is secure against various attacks.

(1) Outsider attack: An attacker cannot obtain value during a payment chain purchase from a broker. The ticket signed by broker can be obtained by an eavesdropper. However, the attacker cannot generate token because the secret information $V_{n,m}$ of the hash chains is not known by the outside attackers. Also an attacker cannot redeem value even if all payment messages are observed. MC or MR will not release tokens until a contract has been received. The contracts indicate the MCs or MR who can redeem the tokens and the unit fee of each token. Redeeming relaying MCs or MR must authenticate themselves using a signature to broker. The outsider attacker cannot redeem without the correct signature for authentication. An attacker cannot impersonate a valid MC or MR. A valid MR or MC holds a public key certificate. With the support of bop by hop authentication between neighbours, even if the outsider attacker can get the certificate, he will be identified in the contract and will be detected by its neighbours. The attacker-signed pricing contract is also a proof of fraud.

(2) MC attacks: for pre-paid scheme, the MC cannot spend more than the total value of a hash chain in the ticket. The billing server will track the balance of the MC's ticket and prevent the hash value being exceeded. For credit-based scheme, the MC cannot spend more than the limit credit from the broker. For each credit-based ticket, each ticket is specific design for a domain. The billing server of the domain will monitor the balance of the ticket and ensure the credit will not be exceeded. The broker will issue new tickets for MCs only if the credit balance of the MC is still within the limit credit of the MC.

Moreover, a MC cannot double spend a hash chain of a ticket. A ticket with a payment hash chain must be spent through MRs of the specific domain that defined for the hash chain. The billing server will track the balance of the hash chain and will not allow hash chain to be double spent. Furthermore, anonymity is provided: MC use tickets which including the hash chains for the billing purpose. The tickets issued by broker only include the ID of the user. The real identity only known by broker. The mesh network cannot know the real identity of customers.

(3) Relaying MC Fraud: the relaying MC cannot obtain more than paid by MR. The value of the payment to a relaying MC is specified in the MR's contract. The broker will use the contract and number of tokens the relaying obtained by the relaying MC to calculate how much is owned by MR. To increase the value per token requires contract to be modified, which is not possible without forging signature. A relaying MC cannot obtain values belongs to another relaying MC. All relaying MCs redeem the same tokens. To obtain other relaying MC's value also requires the forged digital signature. Moreover, contract cannot be replayed without detection. A relaying MC may try to replay an old contract issued by MR and submit to broker for redeem. The TID in the contract is a random number that ensure an old contract cannot be replayed.

(4) MR Fraud: payment chain overspending by the MR can be detected. The broker records the total amount redeemed against a payment chain. When more than the total value spent, it will be detected by the broker.

5. PERFORMANCE ANALYSIS

In this section, we compare the computation and communication costs of MC and MR when login and intra-domain payment protocol are carried out. We consider two scenarios: one hop and multiple hops. Only billing-related computation and communication costs are considered. The authentication messages of the protocols are not included and the costs of acknowledgement messages are ignored.

When considering the communication cost of a payment scheme, the size, length, and number of messages sent between parties must be calculated. When paying for the volume of traffic transported, or if making frequent payments, the signalling overhead, due to payment process should be kept small relative to the payload sent. If payment is made to or from a mobile device over an air interface, with limited or scarce bandwidth, the volume of payment messages should also be minimized.

We use the numbers in table 1 as the size of the basic micropayment constructions. Since the chain length will not exceed 2^{16}, thus m is 1 byte and n is 1 byte.

Table 1

Object	**Size (bytes)**
Hierarchical hash chain	18+16*N (N: depth of the chain)
MC Ticket	232
MC contract	15
MR contract (initial)	27
MR contract (final)	36 + 16*S (S: # of relaying MCs)
MC payment hash chain	27 + 16*N (N: depth of the chain)
MR payment hash chain	27 + 16*N (N: depth of the chain)

We use the numbers in table 2 as the size and speed of the cryptographic algorithms.

Table 2

Cryptographic algorithm	Size (bytes)	Speed (ms)
ECC signature generation		46.4
MD5 hash function	16	0.009
ECC signature verification		92.4
MAC	16	0.015

The computation cost comparisons of login and intra-domain handover in case of one hop are shown in figure 8 and 9.

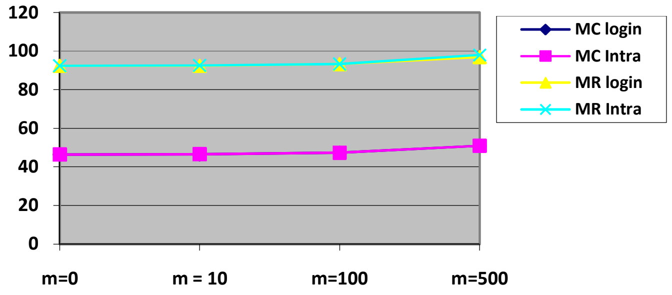


Figure 8 Computation Cost

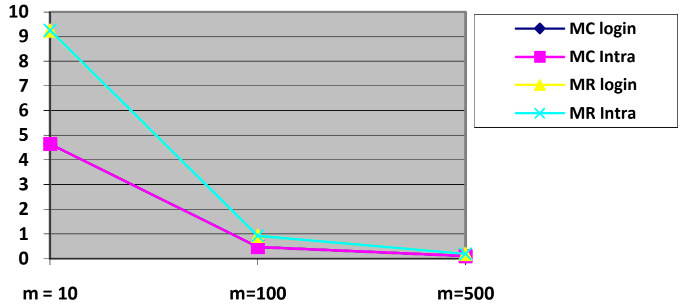


Figure 9. Average Computing Cost Per Payment

We find that the computation cost of MC is about 50% of that of MR. Login and Intra-domain is almost same for MCs or MRs. In addition, the communication cost for token contribution is low and the average computation and communication cost prepayment is reduced dramatically with the increase number of payments. This scheme is optimized for repeated payments to the same vendor.

The comparisons of communication bandwidth used for login and intra-domain handover in case of one hop are shown in figure 10 and 11.

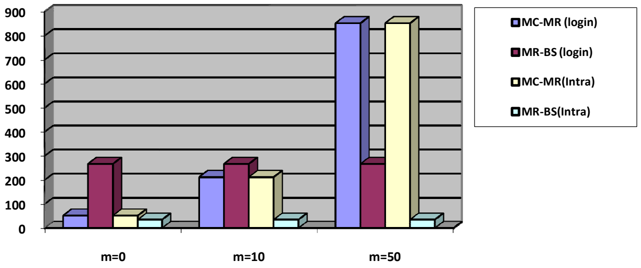


Figure 10. Communication Bandwidth Used

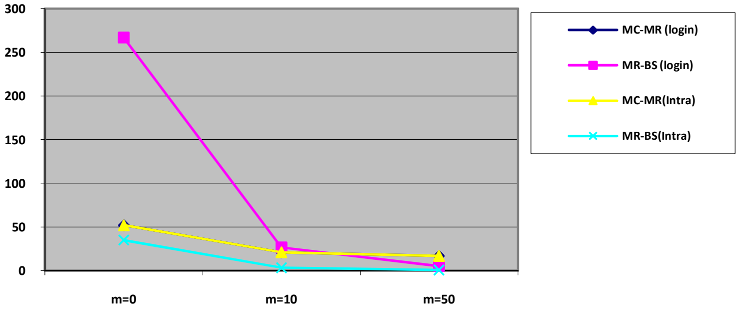


Figure 11. Average Communication Bandwidth Per Payment

If we don't consider hash component (m=0), the bandwidth used by intra-domain of one hop is less than login protocol of one hop. However, with the addition of hashes, the bandwidth of login and intra-domain will gradually become close since the weight of hash will comprise large part of the bandwidth of the communication.

The computation and communication cost of login and intra-domain handover in case of multiple hops (the number of hops is 2) are shown in figure 12 and 13.

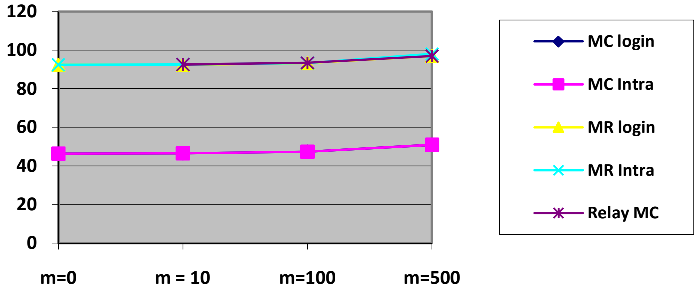


Figure 12: Computation cost (number of hops is 2)

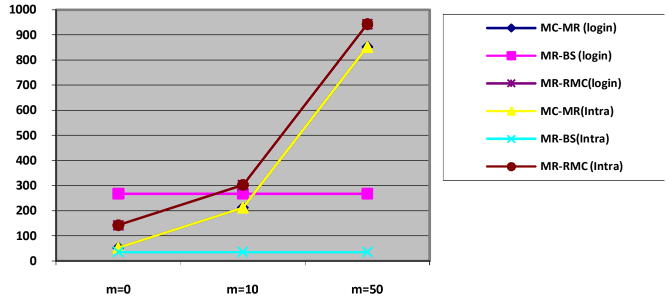


Figure 13. Communication cost (number of hops is 2)

We find that the computation costs of login and Intra-domain of MC-MR are very close. The computation costs of login and Intra-domain of MC-RMC are very close. In addition, the computation cost of a relaying MC is almost the same as the computation cost of MR.

For multiple hops mesh networks, with a specific number of hops, the communication cost for login and intra-domain of MC-MR is very close. The communication cost for login and intra-domain of MR-RMC is also very close. However, the MR-BS for intra-domain is less that that of the login.

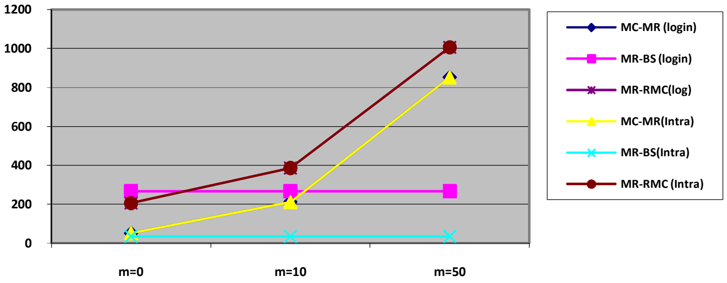


Figure 14. Communication cost (number of hops is 5)

If the number of hops is increase from 2 to 5 (see figure 14), with the increase number of hops, we find that only the MR-RMC communication cost is increased, but others are not affected.

6. RELATED WORK

In this section, we compare our proposal with other micropayment schemes. PayWord[6] is a credit-based scheme. Since the hash chain is generated by the user, there is no control of the credit that the user could have. In our approach, a credit limit has been assigned to each ticket by the broker to control of credit of MCs. S. M. Yen [11] is a prepaid payment scheme. Each customer should buy bank tokens in advance which are used to buy merchant. The MicroMint[6] has double-spending problem. A user may use the same coin to pay two different vendors and the vendors cannot find it until they check with broker. To overcome this problem, the MicroMint scheme needs to trace all the users who purchased the coins. (The vendor need to know the users too so that he can demonstrate who spent those coins). As a result, this schema is not anonymous any more.

Zhang proposed a billing scheme for wireless mesh networks[13]. In the protocol scheme, client and mesh router authenticates each-other and the client has to pay for the services for all the involved entities. The serving mesh router provides network access and backhaul internet service. Brokers are responsible for micropayment aggregation among the entities. The scheme comprises two phases as entity authentication and billing. In the billing scheme, the commitment of payment structure is user-router specific thus there is no chance of double-spending and double-redemption. The good feature of the billing scheme is that new signed commitment of payment structures is not required after route change in ad-hoc network portion as all the relay nodes are paid by the associated router. The usage of payment structures only one payment chain in generated but in total multiple chains are used which will reduced the payment chain storage cost and computation cost of next payment hash generation. However, the scheme is vulnerable to security risk from the payment chain. The unspent portion of the payment hashes may be claimed by corrupt routers by changing the payment hash as authenticated hashes are known to him or user may deny his last payment chain.

Netbill[12] offers a number of advanced features. However, it is relatively expensive: digital signatures are heavily used. This scheme will not be suitable to be applied to mesh networks. Yang[10][11] applied symmetric-key cryptography instead that is more efficient than the public key cryptography and is more suitable for mobile devices. Unfortunately, the symmetric key cryptography requires more frequent key establishments and updates to prevent the shared key from being comprised. In our proposed billing scheme, the authentication and billing approach only use one digital signature and some hash chains. Since the authentication protocol authenticates mesh clients that are not known beforehand by MR, only symmetric key cryptography cannot realize the mutual authentication task. However, we have already minimized the number of the digital signatures in this proposed billing approach. In summary, the related works are not suitable for the security requirements of mesh networks which have been indicated in section 1.

7. CONCLUSION

In this paper, we propose the novice billing scheme for login and intra-domain for wireless mesh networks. The security analysis shows that the protocol is resilient to various kinds of attacks. The performance analysis proves that the intra-domain billing scheme is more efficient when compared with login billing approach.

REFERENCES

[1] Donggang Liu and Peng Ning, Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks. In *Network and Districuted System Security Symposium*, NDSS 2003, pp. 263-276, 2003.

[2] Celia Li and Uyen Trang Nguyen, Trust-based Secure Authentication for Wireless Mesh Networks.

[3] Phone Lin and Hung-yueh Chen, A secure mobile electronic payment architecture platform for wireless mobile networks IEEE Transactions on Wireless Communications, Vol. 7, No. 7, pp. 2705 – 2713, July 2008.

[4] DongGook Park, Colin Boyd and Ed Dawson, Micropayments for Wireless Communications, Lecture Notes in Computer Science, Vol. 2015/2001, pp. 192-205.

[5] N. Asokan and P.A. Janson, "The state of the art in electric payment systems," IEEE Computer, Vol.30, No.9, Oct, 1998.

[6] 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.

[7] M. Bellare, J. Garay and R. Hauser, "Design, implementation and deployment of a secure account-based electronic payment system", IEEE J. Select. Areas Commun., Vol.18, pp. 611-627, Apr. 2000.

[8] Mastercard and Visa, SET Secure Electronic Transactions Protocol, version 1.0 edition.

[9] F. Medvinsky and B. Neuman, “NetCash: a design for practical electronic currency on the Internet”, in Proc. First ACM Conference on Computer and Communication Security, pp. 102-106, 1993.

[10] Z. Yang, W. Lang, and Y. Tan., “A new fair micropayment system based on hash chain.”, in Proc. IEEE International Conference on e-Technology, e-Commerce and e-Service, pp 139-145, 2004.

[11] S. M. Yen, “PayFair: a prepaid Internet micropayment scheme ensuring customer fairness,” in Proc. IEE Computers and Digital Techniques, Vol. 148, No. 6, pp.207-213, Nov. 2001.

[12] The NetBill Electronic Commerce Project, 1995.

[13] Y. Zhang and Y. Fang, “A secure authentication and billing architecture for wireless mesh networks”, Wireless Networks, vo. 13, no.5, pp. 663-678, 2007.

APPENDIX A: PERFORMANCE CALCULATION

When considering the communication cost of a payment scheme, the size, length, and number of messages sent between parties must be calculated. When paying for the volume of traffic transported, or if making frequent payments, the signalling overhead, due to payment process should be kept small relative to the payload sent. If payment is made to or from a mobile device over an air interface, with limited or scarce bandwidth, the volume of payment messages should also be minimized.

(1) MC Ticket:

TicketMC = MC, Brokerid, Date, PubMC, P_{MC}, SigBroker
(2 bytes) MC: the identifier number of the MC
(2 byte) Brokerid: the identifier number of the broker who issue this ticket
(3 bytes) Date: the expire date and time of the ticket
(20 bytes) Pub_{MC}: the public key of the MC
(20 bytes) Sigbroker: the signature signed by the private key of the broker

P_{MC}: ID_{pmc}, W_0 W_1 W_2 ...W_{N-1}, $V_{0,m}$, m, N, Domain, Value, Ticket Credit
(4 bytes) ID_{pmc}: id of the hash chain
(1 byte) Domain: the only domain that the hash chain could be used.
(2 bytes) Value: the purchased value of the chain for the pre-paid approach.
(2 bytes) Ticket Credit: the credit limit for the ticket.
(160 bytes) W_0.... W_{N-1}: 16 bytes * N
(16 bytes) $V_{0,m}$
(1 bytes) m
(1 bytes) N

Public key of ECC: 20 byte (160bits) key is similar as 128 bytes RSA
MD5 hash and MAC size: 16 bytes
Chain length will not exceed 2^{16}, thus m is 1 byte and n is 1 byte.
Size of the MC Ticket: 72 bytes + 16 N bytes = 232 bytes

To control the size of the ticket, we assume N=10. The number of tokens depends on the number of m.

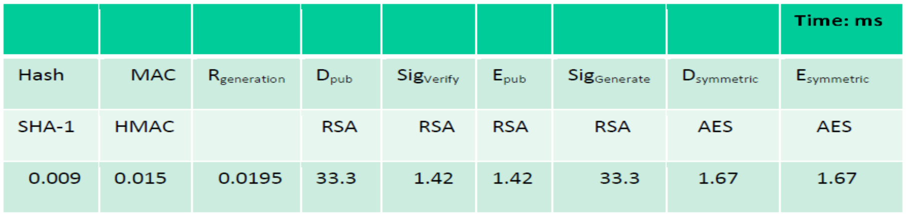
(2) Hash size: 16 bytes

(3) MC contract
ID_{pmc}, MC, MR, $P_{start,}$ Unit Fee, $Broker_{id}$
(2 bytes) MC: ID of MC
(2 bytes) MR: ID of MR

(1 byte) Unit fee: unit fee for each token
(4 bytes) P_{start}: start index of the MR's hash chain
(2 bytes) $Broker_{id}$: ID of the broker
(4 bytes) ID_{pmc}
Size of MC contract: 15 bytes

(4) unit fee, index and current hash value with MAC value
1 byte + 4 bytes + 16 bytes + 16 byte = 37 bytes

(5) Current index and the last hash value of the hash chain used (MR send to billing server)
MC contract + 4 bytes + 16 bytes = 15 bytes +4 bytes + 16 bytes = 35 bytes
(6) MR initial contract
MR, $P_{start,}$ Fee of Relay, $ID_{pmr,}$ $Broker_{id}$, MAC
(2 bytes) MR: ID of MR who issues the contract and distributes it to relaying MCs.
(4 bytes) P_{start}: start index of the MR's hash chain
(16 bytes) MAC
(4 bytes) ID_{pmr}
(1 byte) fee of relay
Size of MR original contract: 27 bytes
(7) P_{MR}: ID_{pmr}, W_0 W_1 W_2 ...W_{N-1}, $V_{0,m}$, m, N, Domain, MR, Credit/Value, Expiry Date
(4 bytes) ID: id of the hash chain
(1 byte) Domain: the only domain that the hash chain could be used.
(2 bytes) Value: the purchased value of the chain for the pre-paid approach.
(2 bytes) Credit of the ticket: the credit limit for the ticket.
(160 bytes) W_0 W_1 W_2 ...W_{N-1}: 16*N bytes
$V_{0,m}$: 16 bytes
m: 1 byte
N: 1 byte
Size of P_{MR}: 187bytes
(8) MR final contract
MR, $P_{start,}$ Fee of Relay, $ID_{pmr,}$ $Broker_{id}$, MACs, Sig
(2 bytes) MR: ID of MR who issues the contract and distributes it to relaying MCs.
(4 bytes) P_{start}: start index of the MR's hash chain
(16 bytes*S) MACs: each relaying MC adds its MAC to the contract
(4 bytes) ID_{pmr}
(1 byte) Fee of Relay
(1 byte) $Broker_{id}$
(20 bytes) Sig
Size of MR final contract: 36 + 16*S



								Time: ms
Hash	MAC	$R_{generation}$	D_{pub}	Sig_{Verify}	E_{pub}	$Sig_{Generate}$	$D_{symmetric}$	$E_{symmetric}$
SHA-1	HMAC		RSA	RSA	RSA	RSA	AES	AES
0.009	0.015	0.0195	33.3	1.42	1.42	33.3	1.67	1.67

One-hop login
MC:
Computation: 1 MAC + Siggeneration + m hash
Computation cost: 0.015 + 46.4 + 0.009*m = 46.415 + 0.009m
ECC signature reference: generation: 46.4ms, verify: 92.4ms

MR:

Computation: 1 MAC + Sigverify + m hash

Computation cost: 0.015 + 92.4 + 0.009*m = 92.415 + 0.009m

Communication:

MC-MR: unit fee, index and current hash value with MAC value (37bytes) + MC contract +

m hash (15 bytes + 16m = 15 + 16m)

Bandwidth: 52 + 16m

MR-BS: MC Ticket 232 bytes + Current index and its hash value of the hash chain used (35 bytes)

Bandwidth: 267 bytes

Multiple-hop login

MC:

Computation: 1 MAC + Siggeneration + m hash

Computation cost: 0.015 + 46.4 + 0.009m = 46.415 + 0.009m

MR:

Computation: 1 MAC + Sigverify + 1 MAC +1 MAC + Siggeneration+m hash +m hash + 1 MAC

Computation cost: 0.015 + 92.4 + 0.03 + 46.4 + 2*0.009m + 0.015 = 138.86 + 0.018m

Relaying MC (with authentication, use this for the combined paper of authentication and billing)

Computation: 2MAC + 2 MAC + m hash + m hash +Sigverify

Computation cost: 4*0.015 + 2*0.009*m + 92.4 = 92.46 + 0.018m

Communication:

MC-MR: MC contract + m hash (15 + 16m) + unit fee, index and current hash value with MAC value (37)

Bandwidth: 52 + 16m

MR-BS: Ticket size (MR sends to billing server) 232 + Current index and its hash value of the hash chain used Communication cost (35)

Bandwidth: 267

MR-RMC: 2 * initial contract + the final contract (2 * 63 + 16*S) + m hash (16m)

Bandwidth: 126 + 16S +16m

One-hop Intra-domain:

MC:

Computation: 1 MAC + Siggeneration + m hash

Computation cost: 0.015 + 46.4 + 0.009 * m = 46.415 + 0.009m

MR:

Computation: 1 MAC + Sigverify + m hash

Computation cost: 0.015 + 92.4 + 0.009*m = 92.415 + 0.009m

Communication:

MC-MR: unit fee and index with the MAC key (37 bytes) + MC contract + m hash (15 + 16 * m)

Bandwidth: 52 + 16m

MR-BS: Current index and its hash value of the hash chain used (35 bytes)

Bandwidth: 35 bytes

Multiple-hop Intra-domain:

MC
Computation: 1 MAC + Siggeneration + m hash
Computation cost: 0.015 + 46.4 + 0.009 * m = 46.415 + 0.009m

MR
Computation: 1 MAC + 1 Sigverify + 1 MAC + Siggeneration + m hash +m hash + 1 MAC
Computation cost: 0.015 + 92.4 + 0.015 + 46.4 + 2 * 0.009 * m + 0.015 = 138.845 + 0.018m

Relaying MC:
Computation: 2 MAC + 1 MAC + Sigverify + m hash
Computation cost: 3 * 0.015 + 92.4 + 0.009 * m = 92.445 + 0.009m

Communication:
MC-MR: MC contract + m hash (15 + 16 * m) + Unit fee, index with MAC (37) +
Bandwidth: 52 + 16m
MR-BS: current index and its hash value of the hash chain used (35)
Bandwidth: 35
MR-RMC: 2 * initial contract + final contract (2 * 63 + 16 * S) + m hash (16m)
Bandwidth: 126 + 16S + 16m

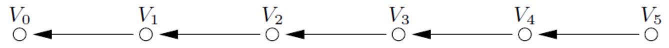
To control the size of the MR final contract, we assume the number of relaying MCs is at most 5. MC software control that when is asked to join the relay service, it should be not moving or move at slow speed.

Appendix B: one-way chain

For setup of the one-way chain, the generator chooses at random the root or seed of the chain, i.e., the value V_N, and derives all previous values V_i by iteratively applying the hash function H.

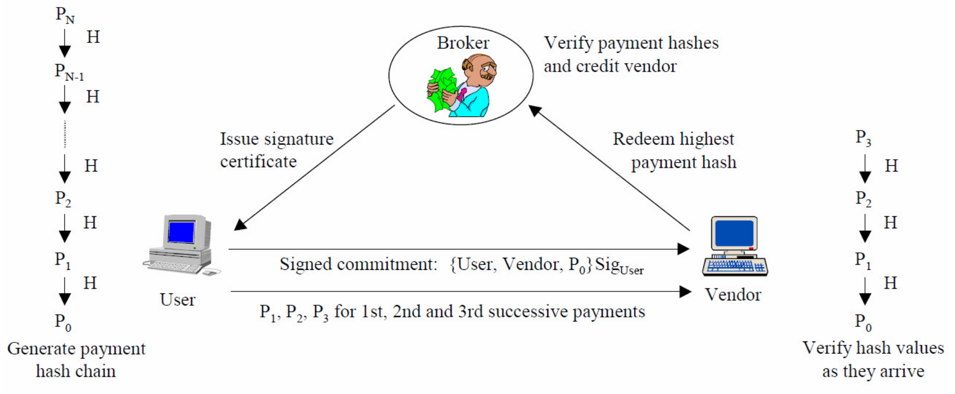
The value V_0, which we refer to as the end-value, is normally made public, and potentially linked to the identity of the user possessing the corresponding root value.

An example of a standard hash chain:



Standard one-level one-way chain.

Hash values from a user-generated hash chain can be used as authenticated payment tokens.


P_N
H
P_{N-1}
H
H
P_2
H
P_1
H
P_0
Generate payment hash chain
Broker
Verify payment hashes and credit vendor
Issue signature certificate
Redeem highest payment hash
User
Vendor
Signed commitment: $\{User, Vendor, P_0\}Sig_{User}$
P_1, P_2, P_3 for 1st, 2nd and 3rd successive payments
P_3
H
P_2
H
P_1
H
P_0
Verify hash values as they arrive

Hash Chain Payment Scheme

On the first payment to a new vendor, the user signs a commitment to that vendor with a new hash chain. By including the vendor identity in the commitment, the vendor is linked to the chain, preventing it being redeemed by other vendors. For each micropayment, the user releases the next payment hash, the pre-image of the current value, to the vendor. Since the hash function is one-way, only the user could have generated this value, and knowledge of it can constitute proof of payment. In essence, the hash chain links the correctness of the current payment to the validity of previous payments. Each hash value is worth the same amount, which can be specified in the commitment. A payment of *m* units is made by releasing the single hash which is the *mth* pre-image of the current hash in the chain. The vendor only needs to apply *m* hashes to verify it.

A broker, or trusted third party, is introduced to aggregate micropayments to many different vendors. Actual monetary value is claimed by redeeming the highest spent hash token, along with the commitment, at a broker with whom the user has an account.

By using a hash chain, the computational cost of a payment is now a single hashing operation for the vendor, after the initial single digital signature verification for a new chain. Where a user spends *n* hashes from a chain to make *z* payments at the vendor the average cost per payment is *(n hashes + 1 signature)/z*. Thus, in the worst case, where a user only ever makes a single purchase from a vendor, the cost is similar to the public key schemes. Therefore, as with the majority of micropayment systems, the scheme is optimized for repeated payments to the same vendor.

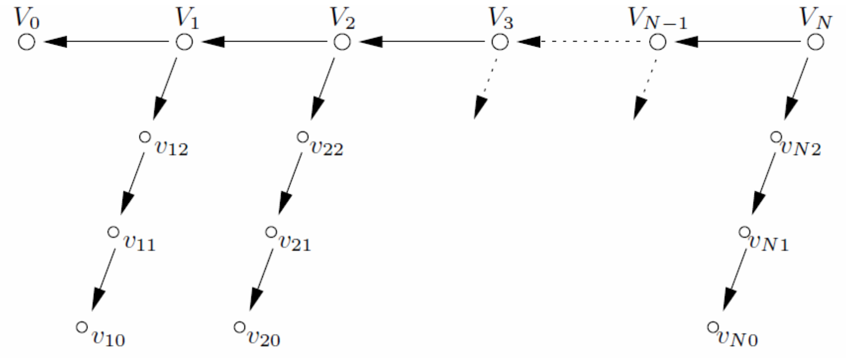
One-way Chain Advantages and Disadvantages

Traditional one-way chains have many advantages. First of all, given only a trusted value V_i of the chain, it is intractable to find a value V_j , where j > i, such that $H^{j-i}(V_j) = V_i$ (assuming that H is a secure one-way function and that the output of H is sufficiently large, we further discuss the security of one-way chains below). However, it is easy to assess the validity of a value V_j, where j > i, by verifying that $H^{j-i}(V_j) = V_i$.

Hierarchical One-Way Chains

A hierarchical one-way chain consists of two or more levels of chains, where values of a first-level (“primary”) chain act as roots of a set of second-level (“secondary”) chains. We refer to the secondary chain rooted in the *i*th value of the primary chain as the *i*th secondary chain. Here, all the values of the *i*th secondary chain are released before any of the values of the i + 1st chain is released; the primary chain value V_i is released in between.

Different one-way functions may be used for primary and secondary chains, with the aim of lowering the communication costs. To set up the hierarchical chain, the generator picks V_N at random and computes the primary chain $V_{N-1}…V_0$. The generator computes the secondary chain on the fly. A clear advantage is the very efficient setup, as only N/K operations are needed to compute V_0, where K is the length of the secondary chain.



Hierarchical one-way chain, where $V_N \ldots V_0$ are values on the one-way primary chain, and the values $v_{i0} \ldots v_{i2}$ are values of a secondary light chain. There is no subchain under V_0 since it serves as the verification value.

To use this one-way chain, the generator traverses all the secondary chains in sequence (e.g., v_{00}, v_{01}, v_{02}, v_{20} …$v_{N0,}$ $v_{N1,}$ v_{N2}) and discloses the values of the primary one-way chain when possible.

A disadvantage of the hierarchical chain is the authentication of end-values of secondary chain. This hierarchical chain was proposed by Liu and Ning [1]. Liu and Ning propose to use the TESLA authentication protocol using the primary chain to authenticate the end-values of the secondary chain. This approach has the shortcoming that the hierarchical chain can only be used in conjunction with the TESLA authentication protocol, as they propose to authenticate the end-values of the secondary chain with the TESLA authentication protocol using the primary chain. The disadvantage of that approach is that the loss of the authentication message prevents the verifier to authenticate secondary chain values until the next value of the primary chain is disclosed. Another shortcoming of their approach is that the authentication is staged, as the generator can only send authentication values at transitions of the primary chain. The tradeoff is clear, on one hand we would like to have infrequent transitions in the primary chain, but on the other hand we prefer a short authentication delay.

Note that the all end-values need to be authenticated - both that of the primary chain and those of all secondary chains. The authentication mechanism by Liu and Ning has several shortcomings. To overcome these shortcomings, we propose the hash chain scheme enabling efficient authentication of the end-values of the secondary chain at any moment, without assuming any additional authentication protocols.

The problem of hierarchical one-way chain is that if the commitment to the end-value of the secondary chain is lost (for example, if V_{20} is lost, the verifier must wait for the disclose of V3 before authenticate the whole chain of V2), the verifier has to wait until the generating value of the secondary chain (i.e., the value of the primary chain) is disclosed.