ESIP: Secure Incentive Protocol with Limited Use of Public-Key Cryptography for Multi-hop Wireless Networks

Mohamed Elsalih Mahmoud, and Xuemin (Sherman) Shen, IEEE Fellow

Abstract—In multi-hop wireless networks, selfish nodes do not relay other nodes' packets and make use of the cooperative nodes to relay their packets, which has negative impact on the network fairness and performance. Incentive protocols use credits to stimulate the selfish nodes' cooperation, but the existing protocols usually rely on the heavy-weight public-key operations to secure the payment. In this paper, we propose secure cooperation incentive protocol that uses the public-key operations only for the first packet in a series and uses the light-weight hashing operations in the next packets, so that the overhead of the packet series converges to that of the hashing operations. Hash chains and keyed hash values are used to achieve payment non-repudiation and thwart free riding attacks. Security analysis and performance evaluation demonstrate that the proposed protocol is secure and the overhead is incomparable to the public-key based incentive protocols because the efficient hashing operations dominate the nodes' operations. Moreover, the average packet overhead is less than that of the public-key based protocols with very high probability due to truncating the keyed hash values.

Index Terms— Network-level security and protection, Mobile communication systems, Routing protocols, Payment schemes.

1 INTRODUCTION

The interest in multi-hop wireless networks such as mobile ad-hoc networks (MANETs), vehicular Ad-hoc networks (VANETs), multi-hop cellular networks (MCNs), and wireless mesh network (WMN) has been increasing significantly [1], [2], [3], [4]. In these networks, a node's traffic is usually relayed through other nodes to the destination. Multi-hop relaying can enable new applications and enhance the network performance and deployment. It can extend the communication range using limited transmit power, improve area spectral efficiency, and enhance the network throughput and capacity [5], [6]. Moreover, in developing or rural areas, multi-hop wireless networks can be deployed more readily and at low cost. However, due to the involvement of autonomous and self-interested devices in the packet relay, the routing process suffers from new security challenges that endanger the

Digital Object Indentifier 10.1109/TMC.2010.211

practical implementation of these networks.

Most existing routing protocols such as DSR [7] assume that the network nodes are willing to relay other nodes' packets. This assumption is reasonable in disaster recovery or military applications because the nodes pursue a common goal and belong to one authority, but it may not hold for civilian applications where the nodes aim to maximize their benefits, since their cooperation consumes their valuable resources such as bandwidth, energy, and computing power without any benefits. In civilian applications, selfish nodes will not be voluntarily interested in cooperation without sufficient incentive, and make use of the cooperative nodes to relay their packets, which has negative effect on the network fairness and performance. Fairness issue arises when a selfish node takes advantage from the cooperative nodes without contributing to them, and the cooperative nodes are unfairly overloaded. The selfish behavior degrades the network performance significantly resulting in failure of the multi-hop communication, e.g., if 10% to 40% of the nodes behave selfishly, the average throughput degrades by 16% to 32%, and the delay increases linearly with the percentage of the selfish nodes [6], [8].

Reputation and incentive based protocols have been proposed to mitigate the problems caused by the selfish nodes [9], [10]. For reputation-based protocols [11], [12], [13], [14], each network node monitors the transmissions of its neighbors to make sure that the neighbors forward other nodes' traffic and thus the uncooperative nodes can be identified and punished. For incentive-based protocols [9], forwarding other nodes' packets is a service not an obligation, so credits (virtual currency) are used to motivate the nodes to collaborate by proving that cooperation is more beneficial than behaving selfishly. The communicating nodes pay credits to the intermediate ones to relay their packets. Reputation-based protocols suffer from unreliable detection to the selfish nodes because it is difficult to differentiate between a node's unwillingness and incapability, e.g., due to low resources, to cooperate. In addition, these protocols cannot achieve fairness because the nodes with higher contributions (which relay more packets) are not compensated. For example, although the nodes situated at the center relays more packets than those at the periphery, they are not compensated.

Incentive-based protocols are more proper for multi-hop wireless networks because in addition to cooperation stimulation, these protocols can achieve fairness by rewarding credits to the cooperative nodes, and discourage packet-flooding attack where the attackers exchange bogus packets to consume the intermediate nodes' resources because the nodes pay for relaying their packets. Moreover, these protocols can also be used for billing the network services without contacting a distant home network register [15]. However, secure incentive protocols usually use signatures to achieve payment non-repudiation which is important to prevent payment manipulation, and thwart free riding attacks because the message's integrity is checked in each hop. These cryptosystems incur too heavy overhead to be used efficiently in limited-resource nodes. In this paper, we propose an **E**fficient and **S**ecure cooperation **I**ncentive **P**rotocol (**ESIP**) that uses public-key operations only for the first packet in a series, and then the efficient hashing operations are used in the next packets. Security analysis and performance evaluation demonstrate that the proposed protocol is secure and the overhead is incomparable to the signature-based incentive protocols because the hashing operations dominate the nodes' operations.

The remainder of this paper is organized as follows. Section 2 reviews the related work, and Section 3 presents the network and threat models. Overview to ESIP and the major contributions are discussed in Section 4. Section 5 proposes ESIP. Security analysis and performance evaluation are provided in Sections 6 and 7, followed by conclusion and future work in Section 8.

2 Related Work

Cooperation incentive protocols can be classified as tamper-proof-device (TPD), electronic-coin, and central-bank based protocols. For TPD-based protocols [16]-[20], a tamper-proof device (which cannot be tampered) is installed in each device to store its credits and secure its operation. For electronic-coin based protocols [21], a network node buys electronic coins in advance from a centralized accounting center (AC) to pay for relaying its packets. In central-bank based protocols [22]-[27], the intermediate nodes usually compose non-deniable receipts and submit them to the AC to update their accounts.

In Nuglets [16], [17], the self-generated and forwarding packets are passed to the tamper proof device to decrease and increase the credit account, respectively. Two models, called the packet purse model (PPM) and the packet trade model (PTM) have been proposed. In the PPM, the source node pays by loading some credits in the packet, and each intermediate node acquires its payment from the packet. In the PTM, each intermediate node buys the packets from the down-stream node and sells them to the up-stream nodes and thus the destination node pays the total cost. In CASHnet [18], [19], for each data packet, the source node's credit account is charged and its signature is attached. The destination node sends back a digitally signed ACK packet to increase the intermediate nodes' credit accounts. The extensive use of digital-signature operations for both the data and the ACK packets is not efficient for limited-resource nodes. For SIP [20], after receiving a packet, the destination node sends back a receipt to the source node that issues a REWARD packet which increments the intermediate nodes' credit accounts, i.e., each packet requires three trips between the source and the destination nodes. However, the TPD-based protocols suffer from the following problems: (1) the assumption that the TPD cannot be tampered is not secure for network with autonomous nodes, and the attackers can communicate freely in undetectable way if they could compromise the devices; and (2) a small number of trusted manufactures can make the network nodes, which is too restrictive for civilian networks.

In [21], each node on a session buys the packets from the down-stream node and sells it to the up-stream node. A packet's buyer contacts the AC to get deposited coins, and the seller claims the coins by submitting them to the AC. The nodes contact the AC interactively in each session to buy and claim the coins, which causes high latency. For Sprite [22], the source node appends its signature to each packet and each intermediate node uses the signature to composes a receipt per packet. Significant communication and computation overhead is implied due to using public-key operations in each packet and generating a receipt per packet. In [23], all the communication has to pass through the base station, which may causes sub-optimal routes when the source and the destination nodes reside in the same cell.

The proposed protocol in [24] reduces the receipts' number by rewarding the nodes probabilistically. The source node appends a payment token to each packet, and the intermediate nodes check whether the token corresponds to winning tickets that are submitted to the AC to reward the winning nodes. However, a fairness issue arises when the nodes are not rewarded adequately, and colluders can exchange tokens to be checked locally to steal. In [25], the source node appends a signature to the full path identities and each intermediate node verifies the signature. The destination node generates a receipt and sends it to the last intermediate node to submit to the AC, but the communicating nodes can communicate freely if the last intermediate node colludes so as not to submit the receipt. Instead of submitting the receipts by all the intermediate nodes, a receipt submission mechanism has been proposed in [26] to reduce the number of submitted receipts and protect against collusion attacks. In addition, a hash chain is used to replace the destination node's signature with hashing operation, but signature operations are used for the data packets. Moreover, instead of generating a receipt per packet, a receipt can contain payment data for multiple packets. Instead of submitting receipts, the proposed protocol in [27] requires submitting a smaller-size report that contains the node's alleged payments in different sessions, and a reputation system is used to identify the cheating nodes. Due to the nature of the reputation systems, some honest nodes may be identified as cheaters falsely, and credit clearance may be delayed identifying the cheating nodes.

3 Network Architecture

3.1 Network Models

The considered multi-hop wireless network includes an AC, a set of base stations (may not available in some networks), and mobile nodes. The AC generates the required cryptographic credentials for a node to participate in the network, and stores and manages the nodes' credit accounts. Once the AC receives a receipt (proof of packet relay), it updates the accounts of the relevant nodes. The source node's packets may be relayed in several hops through the intermediate nodes (and the base station(s) whenever it is necessary) to the destination node. The network nodes can contact the AC at least once during time interval that can be in the range of few days to submit the payment receipts, renew their certificates, and buy credits by real money. This connection can occur via the base stations, Wi-Fi hotspots, or wired networks (e.g., Internet).

For the payment model, a fair charging policy is to support cost sharing between the source and the destination nodes because both of them benefit from their communication. The payment ratio can be adjustable and negotiated during the session establishment phase. For the rewarding policy, some incentive protocols such as [28] rewards the nodes proportionally to the incurred energy in packet relay. This rewarding policy is difficult to implement in practice without involving complicated route discovery process and calculation of en-route individual payments. Therefore, similar to [21], [22], [25], [26], [27], we use fixed rewarding rate, e.g., λ credits per unit-sized packet. The AC charges the two communicating nodes for every transmitted packet even if the packet does not reach the destination node, but the AC rewards the intermediate nodes only for delivered packets. For fair rewarding policy, the value of λ is determined to compensate the nodes for the consumed resources in relaying route establishment packets, packet retransmission, and undelivered packets. In Section 6, we will argue that our payment model can discourage rational attacks and encourage node cooperation.

TABLE 1: USEFUL NOTATIONS.

Symbol	Description
A, B	A is concatenated to B.
$Cert_S$, $Cert_D$	The certificates of the source and the destination nodes, respectively.
H(X)	The hash value resulted from hashing X.
$H_{KSi}(X)$	The hash value resulted from keyed hashing X using the key K_{Si}.
$HS(M_X)$	Hash series of message M_X. A concatenation of the keyed hash values generated by the shared keys between the source (S) and the session nodes (A, B,…, D), e.g. $HS(M_X)=H_{KSA}(M_X)$, $H_{KSB}(M_X)$,…., $H_{KSD}(M_X)$.
ID_i	The identity of intermediate node i, or node with identity ID_i.
ID_S and ID_D	The identities of the source and the destination nodes, respectively.
K_{Si}	The symmetric key shared between the source node and the intermediate node i.
M_i	Message sent in the *i*th data packet.
R	The concatenation of the route nodes' identities, e.g. ID_S, ID_1,…., ID_D
R_L	The route length, i.e., the number of nodes on a route including the source and the destination nodes.
$Sig_S(X)$ and $Sig_D(X)$	The signatures of the source and the destination nodes on X, respectively.
TS	The session establishment time stamp.
V_S^X, V_D^X	The hash value number X in the hash chains generated by the source and the destination nodes, respectively.

3.2 Threat and Trust Models

Since the network nodes are autonomous, we assume that an attacker can change the node's operation and infer the cryptographic data. Attackers can work individually or collude with each other under the control of one attacker to launch sophisticated attacks. The AC is fully secure because it is impossible to realize secure payment between two entities without trusted third party [29]. However, the nodes and the base stations may be rational attackers in the sense that they may misbehave when they can achieve more benefits than behaving honestly. The base stations may be owned by different providers that are motivated to misbehave to increase their welfare. Specifically, the attackers misbehave to communicate freely, pay less, and steal credits. For the trust models, the network nodes fully trust the AC to perform billing and auditing correctly, but the AC does not trust any node or base station in the network. Table 1 gives the used notations in this paper.

4 OVERVIEW AND CONTRIBUTIONS

A practical incentive protocol should achieve two essential requirements: lightweight overhead and security. Heavy-overhead protocol degrades the network performance and exhausts the nodes' resources, which stimulates the nodes to behave selfishly. Due to involving virtual currency in the network, insecure protocol lures the nodes to misbehave to steal credits. Secure incentive protocol usually uses public-key cryptography to thwart various attacks such as payment repudiation and free riding. In Fig. 1, if the message's integrity is checked only end-to-end, nodes A and C can launch free riding attack by adding their data to the session packets to communicate freely. Signature-based protocols can thwart this attack because the message's integrity is checked in each hop, i.e., node B can detect the message manipulation and drop the packet. Signature is also necessary to achieve payment non-repudiation, i.e., to ensure the nodes' approvals to pay to secure the payment.

However, *the public-key operations require much more complicated computations than the hashing operations*, e.g., in Section 7, it will be shown that the verifying and signing operations require computation times and energy that are equivalent to (1061 and 927) and (1119 and 1038) hashing operations using DSA and MD5, respectively. In addition, secure public-key cryptosystems usually have long signature tags which increase the packet overhead. *Therefore, if we can replace the public-key operations with hashing operations and reduce the packet overhead, the network performance can be improved significantly*.

In ESIP, the source and the destination nodes generate hash chains by iteratively hashing random values to obtain final hash values called the hash chains' roots. The two communicating nodes authenticate their hash chains by digitally signing the roots and sending the signatures to the intermediate nodes in the route reply and the first data packets. From the second data packet, only the efficient hashing operations are required. Payment non-repudiation can be achieved by releasing the pre-image of the last sent hash value because the hash function is one-way, i.e., only the user can generate the hash chain. In order to thwart free riding attack, the hop-by-hop message integrity can be checked by attaching a truncated keyed hash value for each node on the route. In Section 5.1, it will be shown that each node on the session can compute a shared key with the source node (to compute the keyed hash value) by one inexpensive bilinear pairing operation using identity-based key exchange protocol. Each intermediate node verifies the hash chain' element to ensure that it will be rewarded for relaying the packet, verifies the keyed hash value to ensure the message integrity, and relays the packet after dropping its keyed hash value.

Comparing with signature-based protocols, ESIP invests more overhead in the first data packet, but from the second packet, only the lightweight hashing operations are used, so *for a group of packets, the heavyweight overhead of the first packet vanishes, and the overall overhead converges to the lightweight overhead of the hashing operations*. In Section 7.2.2, it will be shown that the cryptographic delay in ESIP is 1.4 and 1.75 times that in DSA and RSA based incentive protocol for the first packet, and for a series of two packets, the delay ratios drop to 0.68 and 0.88. Therefore, *from the second packet, we gain the revenue of the investment from the first packet*. Moreover, *for a group of 13 packets, ESIP requires only 10% and 12% of the cryptographic delay in DSA and RSA based protocols, respectively*.

For the packet overhead, it is obvious that if the number of intermediate node is large the packet overhead will be long, so for the efficient implementation of ESIP, the keyed hash values are truncated significantly, and each intermediate node drops its hash value. In Section 6, we will argue that the severe hash truncation is secure in our protocol. In Section 7.2.1, it will be shown that the average packet overhead in ESIP is less than that of the signature based protocols with very high probability, e.g., *for a series of 10 packets, the data packet overhead in ESIP is 70% and 37% of that in the DSA and RSA based protocol, respectively*.

5 The Proposed ESIP

Our protocol includes three phases. In *Setup Phase*, a network node receives the necessary cryptographic data to participate in the network. In *Communication Phase*, the nodes are involved in communication sessions and the intermediate node saves the resultant payment receipts. In *Receipt Redemption Phase*, the nodes submit the receipts to the AC to redeem them.

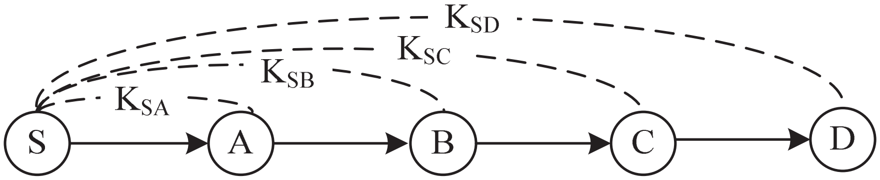


Fig. 1: The source node shares a key with each node on the route.

5.1 Setup Phase

Each node stores a unique identity and public/private key pair with a certificate, the public key of the AC, and the required cryptographic data for the key exchange protocol. As shown in Fig. 1, each node on a session has to share a symmetric key with the source node to compute the messages' keyed hash values. For efficient implementation, an identity-based key exchange protocol based on bilinear pairing can be used because the nodes do not need to exchange messages to compute the shared keys. The AC generates a prime p, a cyclic additive group (G), and a cyclic multiplicative group (G_T) of the same order p such that an efficiently computable bilinear pairing ê: $G \times G \rightarrow G_T$ is known. The bilinear mapping has the following properties:

- **Bilinear**: $\hat{e}(a \cdot P, b \cdot Q) = \hat{e}(b \cdot P, a \cdot Q) = \hat{e}(P, Q)^{a \cdot b}$, for all $P, Q \in G$ and $a, b \in Z_P^*$.
- **Non-degeneracy**: $\hat{e}(P, Q) \neq 1_{GT}$.
- **Symmetric**: $\hat{e}(P, Q) = \hat{e}(Q, P)$, for all $P, Q \in$ G.
- **Admissible**: there is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in G$.

The bilinear map ê can be implemented efficiently using the Weil and Tate pairings on elliptic curves [30]. The AC selects a random element $\mu \in Z_P^*$ known as the master key, and computes the secret keys for the nodes based on their identities. The secret key for node ID_i is $Sk_i = \mu \cdot H(ID_i) \in G$, where $H:\{0,1\}^* \rightarrow G$. Two nodes with identity/secret key pairs (ID_S,

SK_S) and (ID_A, SK_A) can independently compute the shared key as follows:

$$\begin{aligned}
\mathbf{K_{SA}} &= \hat{e}(H(ID_A), Sk_S) \\
&= \hat{e}(H(ID_A), \mu \cdot H(ID_S)) \\
&= \hat{e}(\mu \cdot H(ID_A), H(ID_S)) && (\textit{Bilinear property}) \\
&= \hat{e}(Sk_A, H(ID_S)) \\
&= \hat{e}(H(ID_S), Sk_A) && (\textit{Symmetric property}) \\
&= \mathbf{K_{AS}}
\end{aligned}$$

--------------------------------------- **RREQ** ---------------------------------------

S → * : <**RREQ**, ID_S, ID_D, TS, TTL>

A → * : <**RREQ**, ID_S, ID_D, TS, $\underline{\mathbf{ID_A}}$, $\underline{\mathbf{TTL\text{-}1}}$>

B → * : <**RREQ**, ID_S, ID_D, TS, ID_A, $\underline{\mathbf{ID_B}}$, $\underline{\mathbf{TTL\text{-}2}}$>

C → * : <**RREQ**, ID_S, ID_D, TS, ID_A, ID_B, ID_C, $\underline{\mathbf{TTL\text{-}3}}$>

D : <**RREQ**, ID_S, ID_D, TS, ID_A, ID_B, ID_C, TTL-3>

--------------------------------------- **RREP** ---------------------------------------

D → C → B → A → S:

<**RREP**, R, TS, V_D^{N+1}, $Cert_D$, $Sig_D(R, TS, V_D^{N+1})$>

Fig. 2: Route establishment packets.

5.2 Communication Phase

5.2.1 Route Establishment

As illustrated in Fig. 2, the source node initiates route establishment by broadcasting *Route Request Packet* (*RREQ*) that contains its identity (ID_S), time stamp (TS), and the identity of the destination node (ID_D) and the time to live (TTL). If the time stamp is within a proper range and the TTL is not zero, a network node decrements the TTL, appends its identity, and broadcasts the packet. As shown in Fig. 3, the source and the destination nodes generate hash chains by iteratively hashing random values V_S^1 and V_D^1 to obtain final hash values V_S^N and V_D^{N+1}, where $V_S^i = H(V_S^{i-1})$ and $V_D^i = H(V_D^{i-1})$, respectively. The hash values are released in the direction from V_S^N to V_S^1 and V_D^{N+1} to V_D^1. Payment non-repudiation is achievable because it is difficult to compute V_S^{i-1} from V_S^i or V_D^{i-1} from V_D^i for $2 \leq i \leq N$. After receiving the RREQ packet, the destination node sends back the *Route Reply Packet*

(RREP) containing the identities of the nodes on the route (R = ID_S, ID_A, ID_B, ID_C, ID_D), the time stamp (TS), V_D^{N+1}, and its certificate and signature (Sig_D(R, TS, V_D^{N+1})). The signature authenticates the hash chain, links the hash chain to the session, and proves the node's approval to pay for the session. Each intermediate node verifies the signature, and relays the *RREP* packet. It also saves the signature and V_D^{N+1} to be used in the receipt composition. As illustrated in Fig. 1, the source node and the other nodes on the session compute the shared keys as explained in Section 5.1.

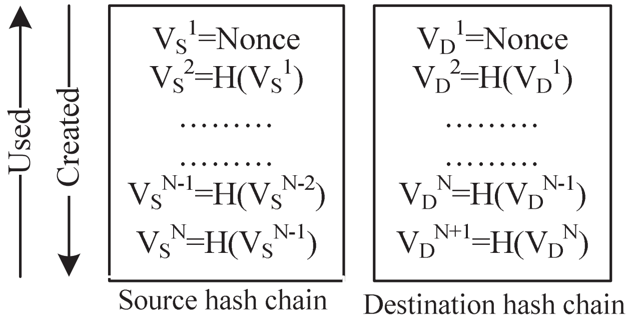


Fig. 3: The source and destination nodes' hash chains.

5.2.2 Data Generation and Relay

In Fig. 4, the source node initiates a packet series with maximum size of N by attaching its signature to the identities of the session nodes, TS, V_S^N. This signature proves the source node's approval to pay for the session, and authenticates its hash chain and links it to the session, i.e., the sender cannot deny generating the hash chain or initiating the session. In order to ensure the hop-by-hop message's authenticity and integrity, the message's hash value ($H(M_1)$) can be included in the signature, but that increases the receipt size because $H(M_1)$ has to be attached to the receipt. Therefore, the source node attaches the hash series $HS(M_1)$ which contains a truncated keyed hash value for each node, i.e., $HS(M_1)$ = $H_{KSA}(M_1)$, $H_{KSB}(M_1)$, $H_{KSC}(M_1)$, $H_{KSD}(M_1)$. Each intermediate node verifies the source node's signature to ensure that it will be rewarded for relaying the packets. Then it verifies its message's truncated hash value to ensure the message's authenticity and integrity, and relays the packet after dropping its hash value. Each intermediate node saves the source node's signature and V_S^N to be used in receipt composition.

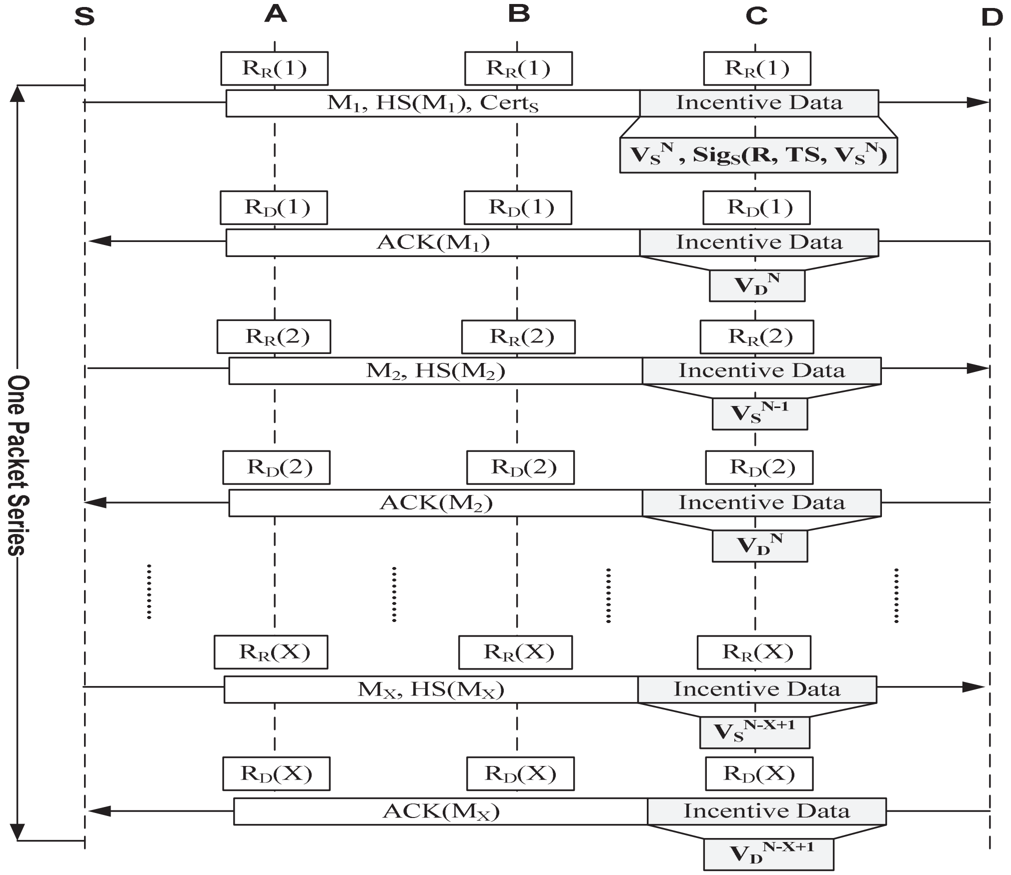
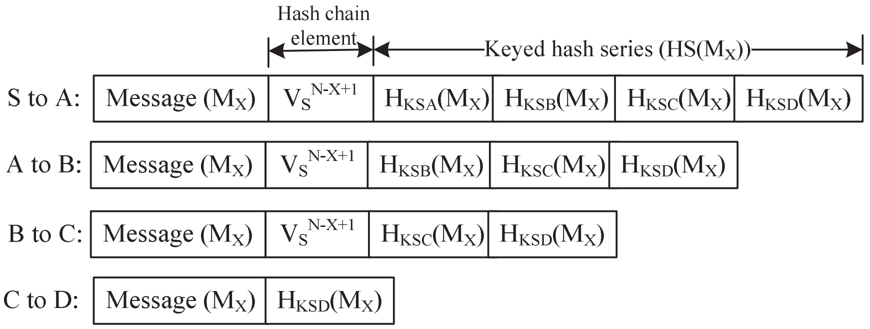


Fig. 4: The exchanged packets in a packet series.



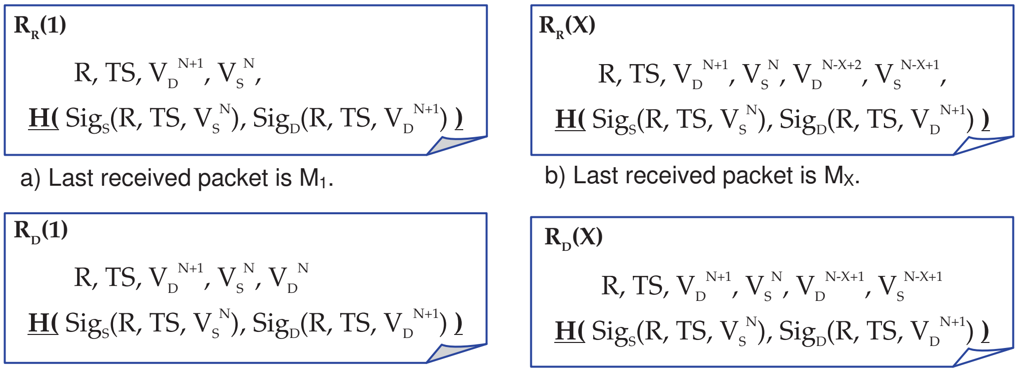
S to A:	Message (M_X)	V_S^{N-X+1}	$H_{KSA}(M_X)$	$H_{KSB}(M_X)$	$H_{KSC}(M_X)$	$H_{KSD}(M_X)$
A to B:	Message (M_X)	V_S^{N-X+1}	$H_{KSB}(M_X)$	$H_{KSC}(M_X)$	$H_{KSD}(M_X)$	
B to C:	Message (M_X)	V_S^{N-X+1}	$H_{KSC}(M_X)$	$H_{KSD}(M_X)$		
C to D:	Message (M_X)	$H_{KSD}(M_X)$				

Fig. 5: The hop-by-hop security packet-overhead in the *X*th data packet, (X > 1).

As shown in Fig. 5, for the successive packets (X > 1), the source node appends the pre-image of the last sent hash value (V_S^{N-X+1}) as an approval to pay for one more packet, and the truncated hash series (HS(M_X)). Each intermediate node verifies its message's truncated keyed hash value, verifies that V_S^{N-X+1} is generated from hashing V_S^{N-X}, and relays the packet after dropping its hash value. Each intermediate node saves the last received hash value (V_S^{N-X+1}) to be used in receipt composition.

5.2.3 ACK Generation and Relay

From Fig. 4, after receiving a data packet, the destination node sends back ACK packet containing a fresh hash value from its hash chain as an approval to pay for the message. Each intermediate node verifies that V_D^{N-X+1} is generated from hashing V_D^{N-X}, and saves the last hash value (V_D^{N-X}) to be used in receipt composition.



$R_R(1)$

R, TS, V_D^{N+1}, V_S^{N},

$\underline{\mathbf{H(}}\, Sig_S(R, TS, V_S^{N}), Sig_D(R, TS, V_D^{N+1})\, \underline{\mathbf{)}}$

a) Last received packet is M_1.

$R_R(X)$

R, TS, V_D^{N+1}, V_S^{N}, V_D^{N-X+2}, V_S^{N-X+1},

$\underline{\mathbf{H(}}\, Sig_S(R, TS, V_S^{N}), Sig_D(R, TS, V_D^{N+1})\, \underline{\mathbf{)}}$

b) Last received packet is M_X.

$R_D(1)$

R, TS, V_D^{N+1}, V_S^{N}, V_D^{N}

$\underline{\mathbf{H(}}\, Sig_S(R, TS, V_S^{N}), Sig_D(R, TS, V_D^{N+1})\, \underline{\mathbf{)}}$

c) Last received packet is ACK of M_1.

$R_D(X)$

R, TS, V_D^{N+1}, V_S^{N}, V_D^{N-X+1}, V_S^{N-X+1}

$\underline{\mathbf{H(}}\, Sig_S(R, TS, V_S^{N}), Sig_D(R, TS, V_D^{N+1})\, \underline{\mathbf{)}}$

d) Last received packet is ACK of M_X

Fig. 6: The receipt formats.

5.2.4 Receipt Composition

If the session is broken after receiving the first data packet, the intermediate nodes compose a receipt for receiving one packet $R_R(1)$, as shown in Fig. 4. In Fig. 6(a), $R_R(1)$ contains the payment data and a security token. The payment data includes the identities of the payers and payees (R), the time of the transaction (TS), and the roots of the payers' hash values. The security token is the hash value of the source and destination nodes' signatures or H(Sig_S(R, TS, V_S^N), Sig_D(R, TS, V_D^{N+1})). Attaching the hash of the signatures instead of the signatures can reduce the receipt size significantly. The security token can guarantee that the receipt is non-deniable and non-forgeable. The $R_R(1)$ is a proof of receiving the message

M_1, but if node ID_i submits $R_R(1)$, it is clear that all the nodes before ID_i on the session route indeed relayed the packet. If the session is broken after receiving M_X, the intermediate nodes compose receipt $R_R(X)$ for delivering X-1 packets and receiving one. From Fig. 6(b), the number of delivered messages (X-1) is the same as the number of hash values in mapping V_D^{N-X+2} to V_D^N, and the number of transmitted packets (X) is the same as the number of hash values in mapping V_S^{N-X+1} to V_S^N. Since the session is broken before receiving the ACK of M_X, the last released hash value from the destination node is V_D^{N-X+2} not V_D^{N-X+1}. If the last received packet is the ACK of M_X, the receipt $R_D(X)$ is composed which is a proof for successfully delivering X messages. In Fig. 6(c), $R_D(1)$ contains two hashes from the destination but only one hash value from the source node. In Fig. 6(d), the number of delivered messages is the same as the number of hash values for mapping V_S^{N-X+1} to V_S^N, or for mapping V_D^{N-X+1} to V_D^{N+1}. The evolution of the session receipt is shown in Fig. 4.

5.3 Payment Redemption Phase

The network nodes periodically submit the receipts to the AC to redeem them. Once the AC receives a receipt, it first checks that the receipt has not been deposited before using the receipt's unique identifier, i.e., the identities of the nodes on the route and the establishment time (R, TS). Then, the AC verifies the receipt credibility by generating the source and the destination nodes' signatures, and matching the signatures' hash value with the receipt's security token. Finally, the AC counts the packets' number from the hash chain's elements, and clears the receipt accoding to the rewarding and charging policy discussed in Section 3.1.

6 Security Analysis

To simplify our presentation, we considered that a keyed hash value covers only the message, but for better security, it should cover the whole packet. For example, in Fig. 5, the keyed hash value of node B should be $H_{KSB}(M_X, V_S^{N-X+1}, H_{KSC}(M_X), H_{KSD}(M_X))$, so if node

A manipulates the hash value of D, e.g., to consume the nodes' resources because the packet will be dropped at D, node B can stop propagating the incorrect packet. Since the source node attaches a keyed hash value for each node on the route, it is obvious that the packet overhead will be large for long routes. To reduce the packet overhead, the message's keyed hash value can be truncated significantly, e.g., the size of the truncated hash value (γ) can be 4 or 5 bytes instead of 16 bytes in HMAC-MD5. This severe hash truncation is secure in our protocol for the following reasons: (1) *The packet security lifetime is extremely short*, i.e., if an intermediate node does not relay a packet in a short time, the route is considered broken and re-established, so a malicious node does not have long time to run complicated algorithms to figure out the truncated keyed hash values for the manipulated message; (2) *Without knowing the secret key, computing the keyed hash value is difficult*; and (3) *An attacker has to figure out a keyed hash value for each victim between itself and the other colluder.* Therefore, *an attacker has to compute multiple truncated keyed hash values without knowing the keys in a limited time, which is so difficult*. What an attacker can do is to replace the truncated hash with a random value, but the probability to hit the correct value is extremely low, e.g., for $\gamma = 4$ bytes, the probabilities to hit one and two correct hash values are 0.23×10^{-9} and 0.05×10^{-18}, respectively.

However, hash truncation increases the random collision probability, i.e., the corrupted and the original messages have the same truncated keyed hash value. Using birthday paradox, the random collision probabilities for γ of 4 and 5 bytes are 1.2×10^{-5} and 7.63×10^{-7}, respectively. In addition, since message integrity is checked in each hop, the probability that the destination node falsely accept a corrupted message as correct is ($n_1 \times 1.2 \times 10^{-5}$) for γ of 4, which is equivalent to the probability that hash collision occurs in n_1 successive nodes, where n_1 is number of nodes from the node at which the message is corrupted to the destination node. This probability can be reduced with the increase of γ but the packet overhead increases, so γ can be dynamic to balance between the probability of falsely accepting corrupted message and the packet overhead, i.e., γ can be longer for short routes. Moreover, some nodes on the route can have longer γ than others, e.g., γ can be longer for the destination node to prevent falsely accepting corrupted messages. MD5 is faster and has shorter digest length than SHA-2, but SHA-2 is more collision resistant, so SHA-2 can be used in signing operations that require high collision resistance, and MD5 is used to compute the keyed hashes and the hash chain.

For *Free Calling* (or *Riding*) *Attacks*, the attackers attempt to communicate freely. Two colluding intermediate nodes on a legitimate session may manipulate the packets by adding their exchanged data. Attackers may record valid packets and replay them in different place and/or time to establish sessions under the name of others. To thwart these attacks, a message integrity and authenticity is checked in each hop by verifying the message's keyed hash value. Time stamp is used to thwart packet-replay attack. For *Double Spending Attack*, attackers attempt to generate identical receipts for different sessions to pay once. In ESIP, even if an attacker establishes different sessions at the same time, the receipts' identifiers are different because at least one intermediate node is different. In *Double Clearance Attack*, the attacker tries to clear a receipt several times to be rewarded multiple times for the same session. The AC can thwart the attack and identify the attackers using the receipts' unique identifiers. For *Credit Stealing Attack*, the attacker attempts to forge receipts or manipulate valid receipts to increase their rewards. This is impossible in ESIP due to the difficulties of forging or modifying the payers' signatures and computing X from H(X). In *Message Repudiation Attacks*, attackers deny transmitting a message. In ESIP, each node can ensure that the intended user has sent a message, but unlike signature-based protocols, it cannot prove that to a third party. However, message non-repudiation is important for other applications such as electronic commerce where a user sends messages to authorize the recipients to perform actions on its behalf. In payment repudiation attacks, attackers attempt to deny initiating a session or the amount of payment so as not to pay. In ESIP, the payers cannot deny the payment data because the signatures and the hash chains can guarantee the payment non-repudiation.

Our payment model can discourage rational cheating actions, and encourage the nodes' cooperation. Particularly, a rational node can exhibit one of the following actions: (1) If the nodes are rewarded for every relayed packet even if it does not reach the destination, the colluding nodes can increase their rewards with consuming low resources by relaying only the smaller-size security tags (hash chains' elements) instead of the message. Our payment model encourages the nodes to rely the messages because they are rewarded only when the destination node acknowledges receiving them; and (2) If the communicating nodes are charged only for delivered messages, the destination node may receive a message, but does not send ACK. To thwart this cheating action, the communicating nodes are charged for un-delivered messages.

7 Performance Evaluation

In this section, simulation results are given to evaluate the overhead cost and the expected network performance using ESIP.

7.1 Simulation Setup

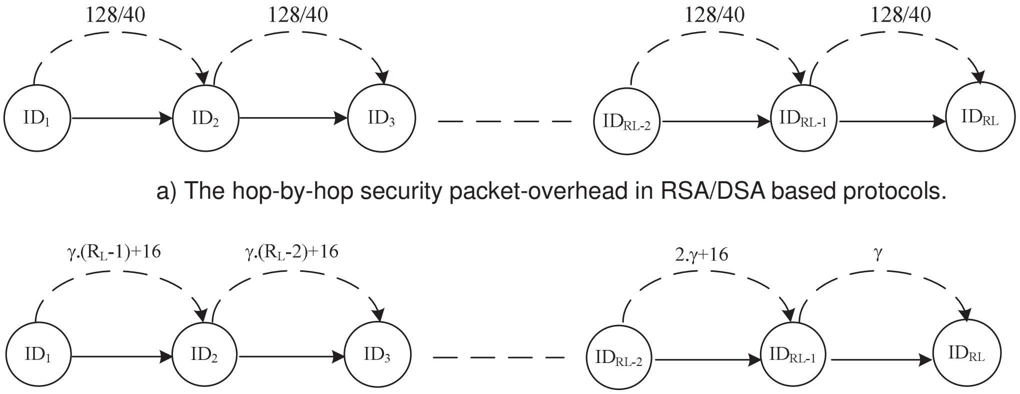
We use 1024-bit RSA and 1024-bit DSA with signature tags of 128 and 40 bytes, respectively, because the secure private keys should have at least 1024 bits according to NIST guidelines [31]. For the hash functions, we use MD5 and HMAC-MD5 [32] with digest length of 16 bytes. For the pairing operation, we consider the Tata pairing implementation on MNT curves where G is represented by 171 bits, and the order P is represented by 170 bits. The discrete logarithm in G is as hard as the discrete logarithm in Z_p^* where $P = 1024$ bits. Network simulator NS2 is used to implement ESIP and signature-based protocol that uses public-key operations in each packet. We simulate multi-hop wireless network by randomly deploying 35 mobile nodes in a square cell of 800 m × 800 m. The Distributed Coordination Function (DCF) of IEEE 802.11 is implemented as the medium access control (MAC) layer protocol. The radio transmission range of a node is 125 m, and the transmission data rate is 2 Mbits/s. A node movement is simulated using the random waypoint model [33] with speed and pause times uniformly distributed in the ranges [0, 10] m/s and [0, 50] s, respectively. Specifically, a node travels towards a random destination uniformly selected within the network field; upon reaching the destination, it pauses for some time; and the process repeats itself afterwards. The constant bit rate (CBR) traffic source is implemented in each node, and the source and destination pairs are randomly chosen. All the data packets are 512 bytes and sent at rate of 2 packets/s. The time stamp and an identity are five and four bytes, respectively. Each simulation is performed 50 runs, and each run is executed for 15 simulated minutes. The averaged results are presented with 95% confidence interval.

In order to estimate the expected processing times of the cryptographic primitives, we have implemented the cryptographic primitives using Crypto++5 library [34] in a laptop with an Intel processor at 1.6 GHZ and 1 GB RAM. The processing times for signing and verifying operations are (15.63 and 0.53) ms and (7.94 and 9.09) ms for RSA and DSA, respectively. The processing times for hashing a 512-byte message and performing a pairing operation are 8.56 μs and 4.34 ms, respectively. A concern in using DSA for multi-hop networks is that the verifying operations performed by the intermediate and destination nodes require more times than the signing operations performed by the source node, and a concern in using RSA is its longer signature tag. The resource of a limited-resource node may be less than a laptop so the processing times are scaled by the factor of five.

7.2 Simulation Results

7.2.1 Average Packet Overhead

We define the *average security packet-overhead* as the average security related data relayed in all the session's hops. In Fig 7(a), the security packet-overhead in signature-based protocols is due to fixed-size and route-length-independent signature, e.g., 40 and 128 bytes for DSA and RSA based protocols, respectively. However, in Fig. 7(b), the security packet-overhead in ESIP is due to the 16-byte hash chain's element (V_S^{N-X+1}) and the message hash series HS(M_X) with γ-byte truncated hash values at X > 1. Fig. 7(b) also shows that V_S^{N-X+1} is not required in the last hop, and the security packet-overhead is reduced by γ bytes in each hop because each node drops its hash value. Unlike signature-based protocols, the security packet-overhead in ESIP is function of the route length (R_L).



a) The hop-by-hop security packet-overhead in RSA/DSA based protocols.

b) The hop-by-hop security packet-overhead in ESIP.

Fig. 7: The hop-by-hop security packet-overhead in ESIP and signature-based protocols.

Fig. 8 gives the relation between the average security packet-overhead and the route length for ESIP. *The figure shows that even at unrealistic and extreme cases, e.g., R_L = 20 nodes, the average security packet-overhead is less than 55 bytes at $\gamma = 4$ bytes.* Fig. 9 gives the equivalent route lengths in signature-based protocol and ESIP for the same average security packet-overhead at γ = 4 bytes. For example, the routes with six nodes in DSA and RSA based protocols are equivalent to routes with 8 and 15 nodes in ESIP for the same average security packet-overhead, respectively. The figure shows that *the average security packet-overhead in ESIP is less than that of the DSA and RSA based protocols when R_L < 13 nodes and R_L < 24 nodes at $\gamma = 4$, respectively*. Moreover, the security packet-overhead of ESIP is less than that of the DSA and RSA based protocols when R_L is less than (17 and 75) nodes and (10 and 45) nodes for γ = 3 bytes and 5 bytes, respectively. Although the DSA has less signature size than RSA, it causes much more end-to-end packet delay due to its longer verification time as shown in Section 7.2.2.

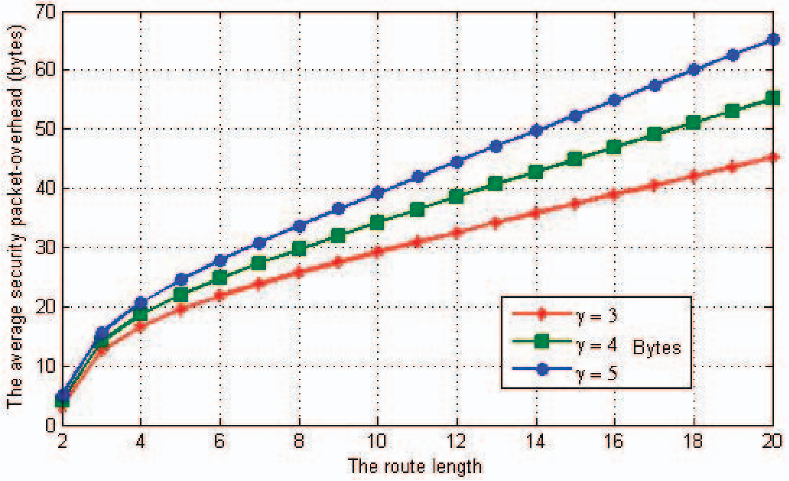


Fig. 8: The average packet security-overhead in ESIP.

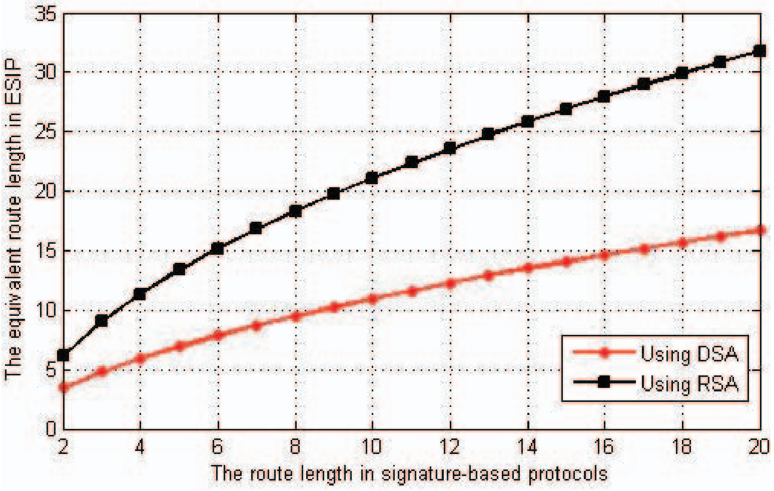
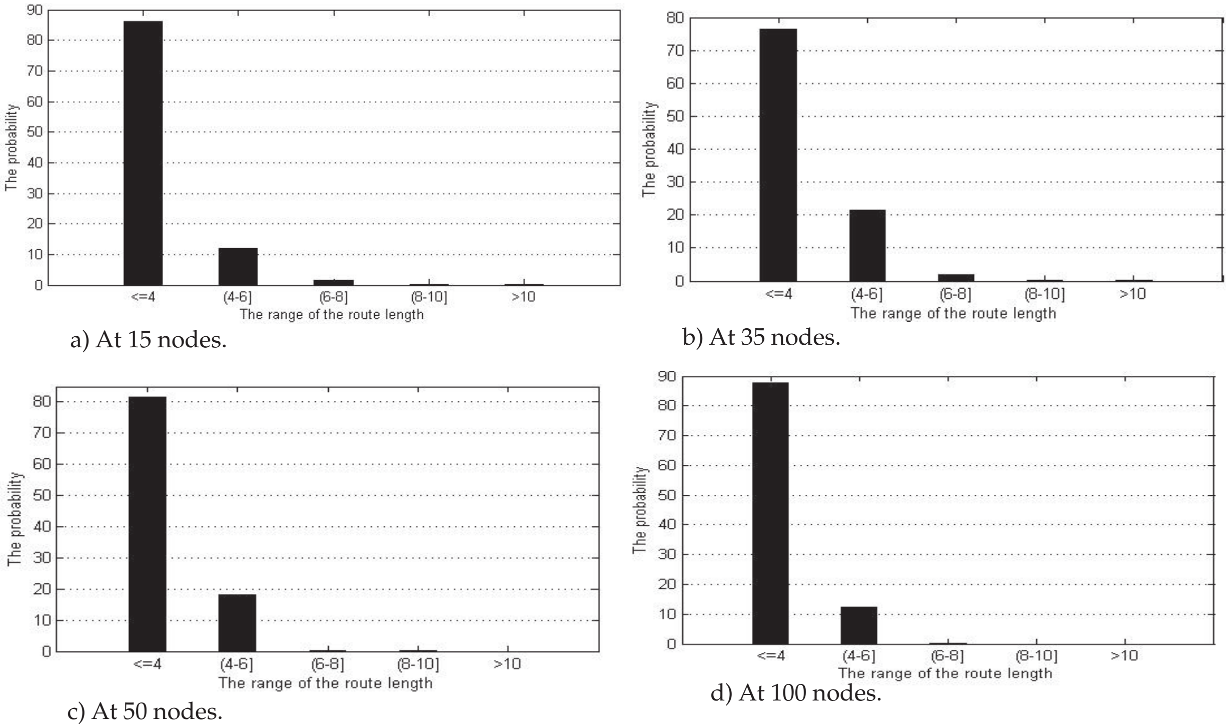


Fig. 9: The equivalent route lengths for the same security packet-overhead.

Fig. 10 shows route length distribution at different number of nodes in the simulated network. At 15 nodes, the network is lightly connected because the average connectivity is 0.66. The network connectivity is measured by the number of connected routes to the total number of routes assuming any two nodes can be the source and destination pair. As shown in Fig. 10(a), 86% of the routes have four nodes or fewer, and only 0.0238% of the routes are longer than ten nodes. At 35 nodes, the average network connectivity is 0.99, the probability a route is shorter than seven is 99.7%, and the probability a route is longer than ten nodes is 0.0151%. At 50 nodes, the probability a route is shorter than seven is 98.1%, and the probability a route is longer than ten nodes is negligible. For dense network with 100 nodes, the probability a route is shorter than seven is 99.99%, and the probability a route is longer than ten nodes is negligible. Table 2 gives the probability that a route is longer than 13 nodes (Pr(R_L > 13)) in different network parameters. *The conclusion of these results is that in realistic network parameters, the route length is less than 13 nodes with very high probability, which means that the expected security packet-overhead of ESIP is less than the DSA and RSA based protocols.*



a) At 15 nodes.

b) At 35 nodes.

c) At 50 nodes.

d) At 100 nodes.

Fig. 10: Route length distribution.

The average packet overhead is the average additional data (in bytes) attached to the message including the routing and security data. Table 3 gives the average packet overhead in ESIP and signature-based protocols. The packet overhead using RSA is much longer than DSA due to its longer signature. For signature-based protocols, the average packet overhead in the first packet is longer than the next packets due to attaching the source node's certificate. For the first packet, both ESIP and signature-based protocols attach the source node's certificate and signature but the average packet overhead of ESIP is more due to attaching

V_S^N and $HS(M_1)$. However, in the next packets, the overhead is less because ESIP has less average security packet-overhead. For the first packet, the packet overhead in ESIP is 1.18 and 1.067 times the overhead in DSA and RSA based protocols, and for a series of two packets, the ratios become 0.98 and 0.79, *so, from the second packet, we gain the revenue from the overhead investment in the first packet. Moreover, for a series of 10 packets, the data packet overhead in ESIP is 70% and 37% of that in the DSA and RSA based protocol, respectively*.

TABLE 2: THE PROBABILITY THAT A ROUTE IS LONGER THAN 13

Network Dimension	Nodes' Number	Average Connectivity	Average Route Length	Pr (R_L > 13)
800 X 800	15	0.66	3.25	0
	30	0.97	3.66	0
	60	1	3.41	0
1600 X 1600	40	0.2235	3.6892	0.000444
	60	0.5394	5.5683	0.011
	100	0.9531	6.3174	0.0059
2000 X 2000	100	0.5591	7.4081	0.091
	150	0.948	7.7624	0.0539
	200	0.992	7.172	0.01225

TABLE 3: THE AVERAGE DATA PACKET OVERHEAD (IN BYTES).

		RSA	DSA
Signature-based protocols	X = 1	279	103
	X > 1	143	55
ESIP	First packet (X = 1)	297.73	121.73
	Subsequent packets (X > 1)	33.73	

7.2.2 Average End-to-End Packet Delay

The required cryptographic operations for ESIP and signature-based protocols are given in Table 4, where P, V, S, and H refer to pairing, verifying, signing, and hashing operations, respectively. It can be seen that ESIP requires more cryptographic operations in the RREP and first data packets, but from the second data packet, ESIP requires only hashing operations.

At R_L = 4 nodes, Fig. 11 shows that the ratio of the ESIP's cryptographic delay to that of the signature-based protocols are 1.4 and 1.75 using DSA and RSA, respectively for the first packet, and the ratios become 0.68 and 0.88 for two packets. Moreover, *for 13 packets, ESIP* requires *only 10% and 12% of the cryptographic delay in DSA and RSA based protocols,* *respectively*. In addition, the simulation results given in Table 5 shows that under different network parameters, *the average size of the packet series is greater than 13, and the cryptographic delay in ESIP is incomparable with that of DSA or RSA based protocols.*

TABLE 4: THE REQUIRED CRYPTOGRAPHIC OPERATIONS IN ESIP AND SIGNATURE-BASED PROTOCOLS.

	ESIP			Signature-based protocols		
	Source node	**Intermediate node**	**Destination node**	**Source node**	**Intermediate node**	**Destination node**
RREP packet	2 × V + P × (R_L-1)	2 × V × (R_L-2) + P	S + P	2 × V	2 × V × (R_L-2)	S
Data packet (X = 1)	S + H × R_L	2 × V × (R_L-2) + 2 × H × (R_L-2)	2 × V + H	S	2 × V × (R_L-2)	2 × V
Data packet (X > 1)	H × R_L	2 × H × (R_L-2)	H	S	V × (R_L-2)	V

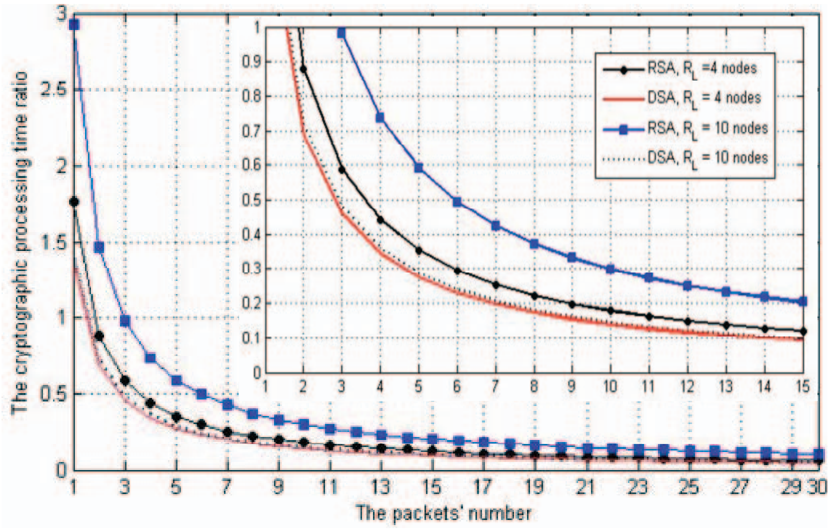
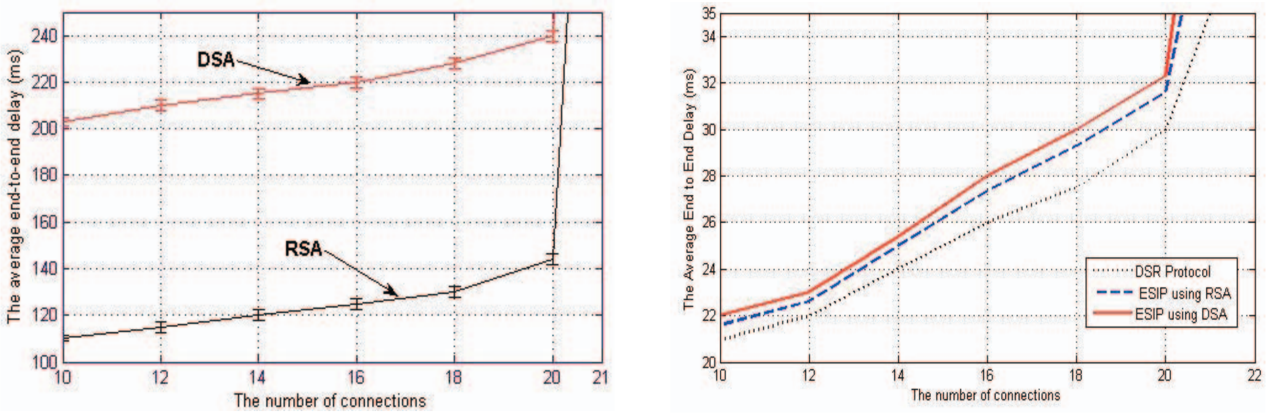


Fig. 11: The ratio of ESIP's cryptographic delay to that of signature-based protocols.

TABLE 5: THE AVERAGE PACKET SERIES SIZE, AND CRYPTOGRAPHIC TIME AND ENERGY RATIOS.

Speed (m/s)	**Transmission rate** / **Nodes number**	**0.5 (packet/sec)**					**1 (packet/sec)**				
		Av. packet series size	**Cryptographic energy ratio**		**Cryptographic delay ratio**		**Av. packet series size**	**Cryptographic energy ratio**		**Cryptographic delay ratio**	
			DSA	**RSA**	**DSA**	**RSA**		**DSA**	**RSA**	**DSA**	**RSA**
[0, 2]	**15**	126.8	0.029	0.033	0.037	0.051	289.7	0.01	0.011	0.011	0.016
	35	134.15	0.015	0.018	0.019	0.027	258	0.012	0.015	0.015	0.022
[0, 10]	**15**	42.55	0.09	0.1	0.117	0.16	94.6	0.05	0.055	0.063	0.084
	35	40.425	0.098	0.11	0.13	0.173	95.4	0.05	0.056	0.064	0.088

The average end-to-end packet delay refers to the average time that packets traverse the network from the source node to the destination node. The end-to-end packet delay is due to propagation, cryptographic, queuing, etc, delays. Fig. 12 gives the average end-to-end packet delay in ESIP and signature-based protocols at different traffic load expressed in connections' number, and Table 6 give the confidence intervals for Fig. 12(b). *The simulation results demonstrate that ESIP can reduce the average end-to-end packet delay significantly comparing with DSA and RSA based protocols because the hashing operations that are computationally free (43 µs per operation) dominate the nodes' operations.* Up to 20 connections, the cryptographic delay dominates the channel contention and queuing delays, but over 20 connections, the delay significantly increases with and without incentive protocol because the channel contention and queuing delays dominate. Although the DSA has shorter signature than the RSA, it causes longer delay in signature-based protocols due to its longer verification time, but the DSA increases the delay very little in ESIP because the effect of the long-delay first packet vanishes with the dominant hashing operations. Hence, *ESIP can be implemented more efficiently using DSA because it has shorter signature and the hashing operations can alleviate the long delay of the first packet.*



a) Signature-based protocols.

b) ESIP and original DSR.

Fig. 12: The average end-to-end packet delay.

7.2.3 Packet Delivery Ratio

The packet delivery ratio (PDR) refers to the average ratio of data packets successfully delivered to the destination nodes with respect to those generated by the source nodes. Fig. 13 gives the PDR for ESIP and the original DSR at different connections' number, and Table 6 give the confidence intervals. Up to 20 connections, the PDR is quite high (above 98%).

Beyond 20 connections, the PDR starts to decrease because more packets are dropped due to increasing the number of congested nodes and packet collision. Since each node has only 50-packet queue size and increasing the connections' number increases the packet arrival rate, the node is congested and drops the packets once the buffer is full. Moreover, increasing the cryptographic delay causes more congested nodes due to increasing the packet processing (or service) time. Comparing to the original DSR protocol, ESIP has a very little effect on the PDR because the dominant hashing operations require very little computational time.

TABLE 6:. 95% CONFIDENCE INTERVAL (C.I.) FOR MEAN.

The connections' number	C. I. for mean	End-to-end delay			Packet delivery ratio		
		ESIP using DSA	ESIP using RSA	DSR	ESIP using DSA	ESIP using RSA	DSR
12	Upper limit	23.1	22.8	22.05	99.9	99.99	99.997
	Mean	23.03	22.6	22	99.93	99.95	99.994
	Lower limit	22.96	22.4	21.95	99.9	99.93	99.991
14	Upper limit	25.95	25.5	24.21	99.98	99.967	99.989
	Mean	25.4	25	24	99.94	99.96	99.989
	Lower limit	24.85	24.5	23.79	99.9	99.959	99.989
16	Upper limit	28.61	27.68	26.2	99.92	99.96	99.988
	Mean	28.01	27.36	26	99.9	99.95	99.985
	Lower limit	27.41	27.04	25.8	99.88	99.94	99.982
18	Upper limit	30.21	29.5	28	99.64	99.92	99.994
	Mean	30.02	29.3	27.5	99.6	99.88	99.989
	Lower limit	29.83	28.1	27	99.56	99.84	99.985
20	Upper limit	32.39	31.82	29.86	98.56	98.93	99.8
	Mean	32.32	31.6	30	98.5	98.9	99.6
	Lower limit	32.25	31.38	30.14	98.44	98.87	99.4

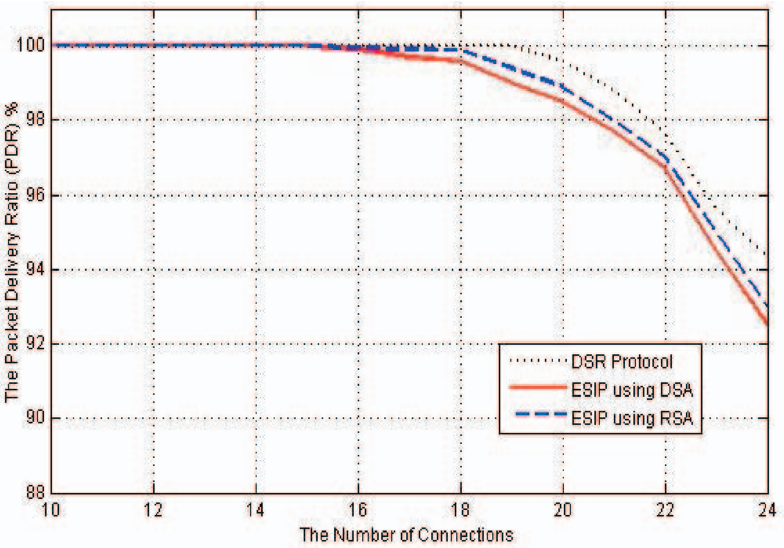


Fig. 13: The packet delivery ratio.

7.2.4 Average Network Throughput

The average network throughput is computed by dividing the size of the received data by all the nodes over the simulation time. Since the end-to-end packet delay and the PDR in ESIP are close to those of the DSR, it is expected that the throughput of ESIP is close to that of the DSR. The simulation results shown in Fig. 14 demonstrate that ESIP has very little effect on the throughput comparing with the original DSR protocol. Up to 20 connections, the throughput increases with increasing the number of connections, but beyond 20 connections, the increasing rate starts to decrease because the network starts to enter its maximum capacity, i.e., above 20 connections, the PDR decreases and the end-to-end packet delay increases as discussed in Sections 7.2.2 and 7.2.3, respectively.

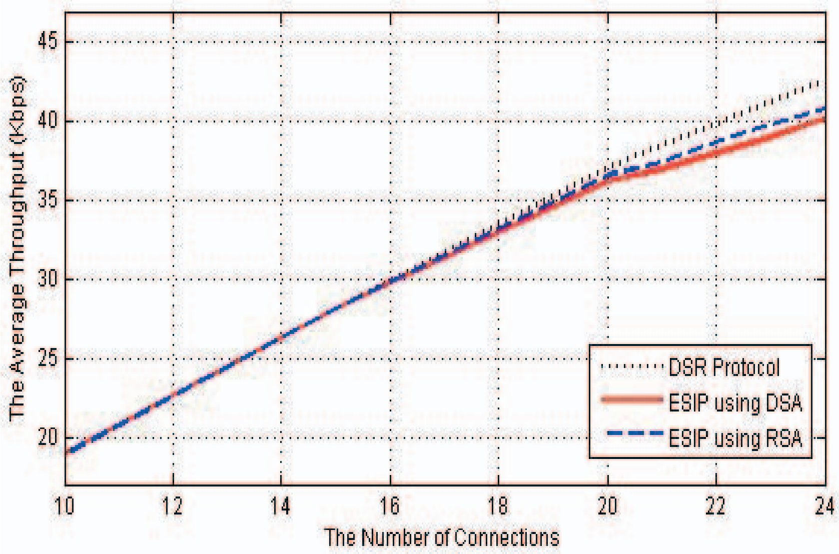


Fig. 14: The average throughput.

TABLE 7: THE AVERAGE RECEIPT SIZE.

	Average receipt size (Bytes)
Sprite using DSA	61.83
Sprite using RSA	149.83
DSC	87.83
ESIP	101.83

7.2.5 Average Receipt Overhead

In sprite, a receipt is composed per packet, but DSC and ESIP generate a receipt when the route is broken or N (hash chain length) packets are sent. Table 7 gives the expected receipt size for Sprite, DSC, and ESIP. The receipt size in ESIP is larger than that of DSC because ESIP attaches two hash values from the source nodes' hash chain. 1MB storage capacity may hold up to 11,938.7 and 10,297.32 receipts in DSC and ESIP, respectively. The little bit increase of the receipt size in ESIP is not comparable by the improvement in the end-to-end delay and the PDR.

TABLE 8: THE AVERAGE RECEIPTS' SIZE FOR 10 MIN. DATA TRANSMISSION (KBYTES)

Node speed	Transmission rate (packets/sec)	Sprite (DSA)	DSC			ESIP		
			N = 10	N = 20	N = 30	N = 10	N = 20	N = 30
2 m/s	0.5	29.88	4.29	3.36	3.28	4.98	3.898	3.81
	2	124.74	17.65	8.825	7.65	20.46	10.23	8.87
10 m/s	0.5	53.03	7.75	7.52	7.42	8.99	8.72	8.60
	2	212.17	30.95	15.47	15.02	35.88	17.94	17.41

We run a simulation to estimate the expected size of the generated receipts in a session held for ten minutes at different transmission rates and random source and destination pair. From Table 8, the receipts' size in Sprite is much larger due to generating a receipt per packet, and the size significantly increases with high transmission rate due to generating more packets during the 10-min data transmission. For DSC and ESIP, less receipts' size is required with the increase of N because a receipt can contain payment data for more packets. However, the increase of N beyond 20 cannot reduce the receipts' size significantly because the routes are broken before releasing the 20 hash values. The optimal value for N depends on the number of transmitted packets before a route is broken, which is related to the nodes' speed and the channel quality. In DSC and ESIP, the receipts' size increases witht high node mobility because more receipts are generated due to breaking the routes more frequently.

7.2.6 Energy Consumption

Energy is consumed in relaying the packets and executing the cryptographic primitives. As discussed in Sections 7.2.1, ESIP can reduce the packet overhead with very high probability. On the other hand, the signing and verifying energy are (546.5 and 15.97) mJ and

(313.6 and 338.02) mJ using RSA and DSA, respectively, and the required energy for hashing 512 byte-message using MD5 and pairing operation are 0.302 mJ and 25.5 mJ, respectively [35], [36]. *These results indicate that the consumed energy for hashing operations is incomparable with those of signing and verifying operations, which supports our approach of replacing the signature with the hashing operations*. Fig. 15 gives the relation between the ratio of the required cryptographic energy in ESIP to that of DSA and RSA based protocols, and the packets' number. At $R_L = 4$, ESIP requires 1.025 and 1.175 of the consumed cryptographic energy for DSA and RSA based protocols, respectively for the first packet. However, from the second packet ESIP requires less cryptographic energy, e.g., for *10 packets, ESIP requires around 10% of the cryptographic energy consumed in DSA and RSA based protocols at R_L=4.* In addition, the simulation results given in Table 5 demonstrate that the average cryptographic energy consumed in ESIP is incomparable with those consumed in DSA and RSA based protocols.

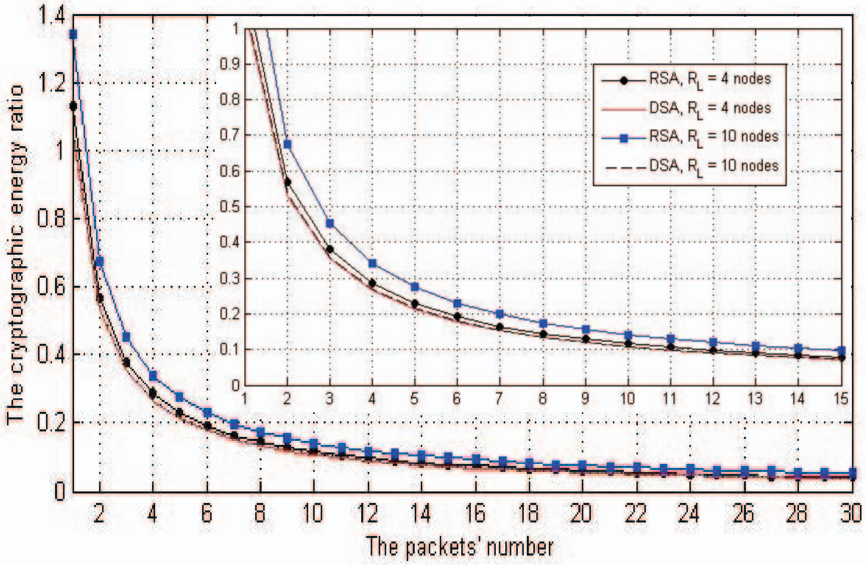


Fig. 15: The ratio of ESIP's cryptographic energy to that of signature-based protocol.

With high node mobility, Table 5 indicates that the average cryptographic delay and energy increase, and Fig. 16 shows that the end-to-end packet delay increases. That is because the size of the packet series decreases with node high mobility, and thus the effect of the heavy-weight first packet increases. From Table 8, the receipts' number increases with high node mobility because the routes are broken more frequently, and a new receipt is generated when the route is broken. However, the simulation results demonstrate that the overhead of ESIP is still incomparable with those of the DSA and RSA based protocols because only the free computational hashing operations are used after the first packet.

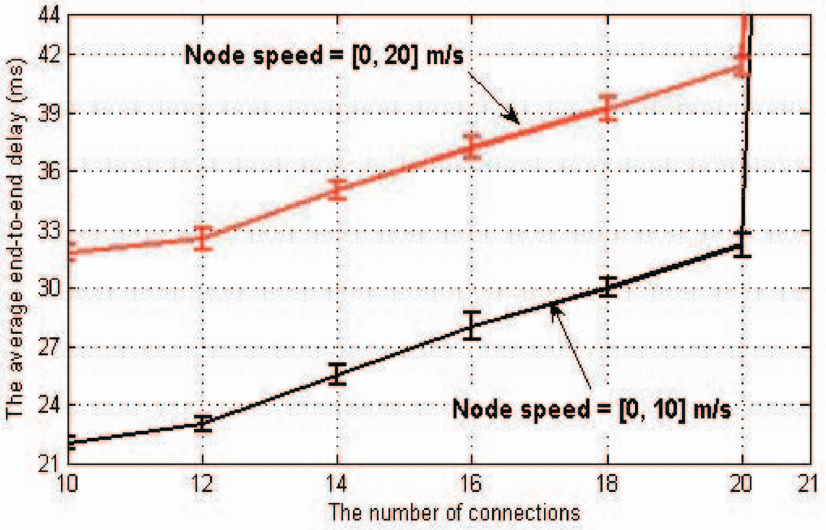


Fig. 16: The impact of mobility on the end-to-end packet delay.

8 Conclusion and Future Work

In this paper, we have proposed secure cooperation incentive protocol with limited use of public key cryptography for multi-hop wireless networks. The public-key operations are required only for the first packet, and then it is replaced with the efficient hashing operations, so for a series of packets, the heavy overhead of the first packet vanishes and the overall overhead converges to that of the lightweight hashing operations. Our security analysis and performance evaluations demonstrate that ESIP can secure the payment, and improve the network performance significantly because the hashing operations dominate the nodes' operations. For a series of two packets, ESIP has lower cryptographic delay and energy than DSA and RSA based protocols, and for a series of 13 packets, ESIP requires around 10% of the cryptographic delay and energy in DSA and RSA based protocols. Moreover, the packet overhead in ESIP is less than that of the DSA and RSA based protocols with very high probability, e.g., for a series of 10 packets, the data packet overhead in ESIP is 70% and 37% of that in the DSA and RSA based protocol, respectively.

In this work, we have implemented virtual currency in the multi-hop wireless network to stimulate the rational packet droppers to cooperate. However, the irrational packet droppers, e.g., compromised or faulty nodes, sacrifice their resources such as energy, bandwidth, credits, etc to harm the network, i.e., they attempt to degrade the network performance by involving themselves in communication sessions and then dropping the packets intentionally. Since the sessions may be broken normally, e.g., due to mobility, or intentionally due to malicious actions, statistical methods are required to identify the irrational attackers that drop the packets more than the normal rate. In ESIP, the receipt format can reveal the node at which the route was broken, so in our future work, we will extend this work to consider the irrational packet droppers. The AC can inspect the submitted receipts to build a reputation system to identify the irrational packet droppers. The reputation system should be carefully designed to identify the attackers in short time to reduce their harm, and to avoid falsely identifying honest nodes as irrational packet droppers.

REFERENCES

[1] G. Shen, J. Liu, D. Wang, J. Wang, and S. Jin, “Multi-hop relay for next-generation wireless access networks”, Bell Labs Technical Journal, vol. 13, no. 4, pp. 175-193, 2009.

[2] X. Li, B. Seet, and P. Chong, “Multihop cellular networks: technology and economics”, Computer Networks, vol. 52, no. 9, pp. 1825–1837, June 2008.

[3] A. Abdrabou and W. Zhuang, "Statistical QoS routing for IEEE 802.11 multihop ad hoc networks", IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1542-1552, March 2009.

[4] Y. Jiang, M. Shi, X. Shen, and C. Lin, "BAT: A robust signature scheme for vehicular networks using binary authentication tree", IEEE Transactions on Wireless Communications, vol. 8, no. 4, pp. 1974 - 1983, 2009.

[5] P. Gupta and P. Kumar, “The capacity of wireless networks”, IEEE Transactions on Information Theory, vol. 46, no. 2, pp. 388-404, March 2000.

[6] S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks”, Proc. of IEEE/ACM MobiCom’00, pp. 255–265, Boston, MA, August 6-11, 2000.

[7] D. B. Johnson, D. A. Maltz, and Y. C. Hu. “The dynamic source routing protocol for mobile ad hoc networks (DSR)”, Technical report, IETF MANET Working Group, February 2007.

[8] P. Michiardi and R. Molva, “Simulation-based analysis of security exposures in mobile ad hoc networks”, Proc. of European Wireless Conference, Florence, Italy, February 25–28, 2002.

[9] J. Hu, “Cooperation in mobile ad hoc networks”, Technical report (TR-050111), Computer Science Department, Florida State University, Tallahassee, January 2005.

[10] G. Marias, P. Georgiadis, D. Flitzanis, and K. Mandalas, “Cooperation enforcement schemes for MANETs: a survey”, Wiley's Journal of Wireless Communications and Mobile Computing, vol. 6, issue 3, pp. 319–332, 2006.

[11] J. Jaramillo and R. Srikant, “DARWIN: distributed and adaptive reputation mechanism for wireless networks”, Proc. of ACM MobiCom’07, pp 87–98, Montreal, QC, Canada, September 9-14, 2007.

[12] Q. He, D. Wu, and P. Khosla, “A secure incentive architecture for ad-hoc networks”, Wireless Communications and Mobile Computing, vol. 6, no. 3, pp. 333-346, May 2006.

[13] N. Jiang, K. Hua, and D. Liu, “A scalable and robust approach to collaboration enforcement in mobile ad-hoc networks”, Communication and Networks, vol. 9, part 1, pp. 56-66, 2007.

[14] L. Xie and S. Zhu, “Message dropping attacks in overlay networks: attack detection and attacker identification”, ACM Transactions on Information and System Security, vol. 11, issue 3, March 2008.

[15] Y. Zhang and Y. Fang, “A secure authentication and billing architecture for wireless mesh networks”, ACM Wireless Networks, vol. 13, no. 5, pp. 569–582, October 2007.

[16] L. Buttyan and J. Hubaux, “Nuglets: a virtual currency to stimulate cooperation in self organized ad hoc networks”, Technical Report DSC/2001/001, Swiss Federal Institute of Technology, Lausanne, January 2001.

[17] L. Buttyan and J. Hubaux, “Stimulating cooperation in self-organizing mobile ad hoc networks”, Mobile Networks and Applications, vol. 8, no. 5, pp. 579-592, October 2004.

[18] A. Weyland and T. Braun, “Cooperation and accounting strategy for multi-hop cellular networks”, Proc. of IEEE Workshop on Local and Metropolitan Area Networks (LANMAN), pp. 193-198, Mill Valley, CA, USA, April 25-28, 2004.

[19] A. Weyland, “Cooperation and accounting in multi-hop cellular networks”, Ph.D. thesis, University of Bern, November 2005.

[20] Y. Zhang, W. Lou, and Y. Fang, “A secure incentive protocol for mobile ad hoc networks”, ACM Wireless Networks, vol. 13, no. 5, pp. 569-582, October 2007.

[21] J. Pan, L. Cai, X. Shen, and J. Mark, “Identity-based secure collaboration in wireless ad hoc networks”, Computer Networks (Elsevier), vol. 51, no. 3, pp. 853-865, 2007.

[22] S. Zhong, J. Chen, and R. Yang, “Sprite: a simple, cheat-proof, credit based system for mobile ad-hoc networks”, Proc. of IEEE INFOCOM’03, vol. 3, pp. 1987-1997, San Francisco, CA, March 30- April 3, 2003.

[23] N. Salem, L. Buttyan, J. Hubaux, and M. Jakobsson, “Node cooperation in hybrid ad hoc networks”, IEEE Transactions on Mobile Computing, vol. 5, no. 4, April 2006.

[24] M. Jakobsson, J. Hubaux, and L. Buttyan, “A micro-payment scheme encouraging collaboration in multi-hop cellular networks”, Proc. of the 7th Financial Cryptography (FC'03), pp. 15–33, La Guadeloupe, January 2003.

[25] B. Lamparter, K. Paul, and D. Westhoff, “Charging support for ad hoc stub networks”, Computer Communications, vol. 26, no. 13, pp. 1504–1514, 2003.

[26] M. Mahmoud and X. Shen, “DSC: cooperation incentive mechanism for multi-hop cellular networks”, Proc. of IEEE ICC’09, Dresden, Germany, June 14-18, 2009.

[27] M. Mahmoud and X. Shen, "Stimulating cooperation in multi-hop wireless networks using cheating detection system", Proc. IEEE INFOCOM'10, San Diego, California, USA, March 14-19, 2010.

[28] L. Anderegg and S. Eidenbenz, “Ad hoc-VCG: a trustful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents”, Proc. of ACM MobiCom, San Diego, CA, USA, September 2003.

[29] H. Pagnia and F. Gartner, “On the impossibility of fair exchange without a trusted third party”, Technical Report TUD-BS-1999-02, Darmstadt University of Technology, March 1999.

[30] D. Boneh and M. Franklin, “Identity based encryption from the weil pairing”, Proc. of Crypto’01, LNCS, Springer-Verlag, vol. 2139, pp. 213-229, 2001.

[31] National Institute of Standards and Technology (NIST), “Recommendation for key management - part 1: general (revised)”, Special Publication 800-57 200, 2007.

[32] A. Menzies, P. Orschot, and S. Vanstone, “Handbook of applied cryptography”, CRC Press, http://www.cacr.math.uwaterloo.ca/hac, Boca Raton, Fla., 1996.

[33] J. Yoon, M. Liu, and B. Nobles, “Sound mobility models”, Proc. of ACM MobiCom, San Diego, CA, USA, September 2003.

[34] W. Dai, “Crypto++ Library 5.6.0”, http://www.cryptopp.com, 2009.

[35] N. Potlapally, S. Ravi, A. Raghunathan, and N. Jha, “A Study of the energy consumption characteristics of cryptographic algorithms and security protocols”, IEEE Transactions on Mobile Computing, vol. 5, no. 2, pp. 128-143, March-April 2006

[36] Y. Zhang, W. Liu, W. Lou, and Y. Fang, “Location-based compromise-tolerant security mechanisms for wireless sensor networks”, IEEE Journal Selected Areas Communication, vol. 24, no.2, pp.247-260, 2006.

Mohamed Elsalih Mahmoud received the B.Sc. and M.Sc. degrees (with honors) in electrical communications engineering from Banha University, Cairo, Egypt, in 1998 and 2003, respectively. He is currently working toward the Ph.D. degree with the Centre forWireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

He is also currently a member of the Broadband Communications Research Group, University of Waterloo. His research interest includes wireless network security, privacy, anonymous and secure routing protocols, trust and reputation systems, cooperation incentive mechanisms, and cryptography.

Dr. Mahmoud received the Best Paper Award from the IEEE International Conference on Communications, Dresden, Germany, June 14-18, 2009. This award is one of 14 awards among 1046 papers presented and more than 3000 total paper submissions and is the unique award for the Communication and Information Systems Security Symposium. He is the first author of more than 13 papers in major IEEE conferences and journals. He also served as a Technical Program Committee member for the Ad-Hoc and Sensor Networks track and the Mobile Applications and Services track at the IEEE Vehicular Technology Conference, which is to be held in Ottawa, ON, Canada on September 6–9, 2010.

Xuemin Shen (M'97–SM'02–F'09) received the B.Sc. degree in electrical engineering from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees in electrical engineering from Rutgers University, Camden, NJ, in 1987 and 1990, respectively.

He is currently a Professor and the University Research Chair with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is the author or a coauthor of three books and more than 400 papers and book chapters on wireless communications and networks, control, and filtering. He serves as the Editor-in Chief for Peer-to-Peer Networking and Application and an Associate Editor for Computer Networks, ACM/Wireless Networks, and Wireless Communications and Mobile Computing. He has also served as a Guest Editor for ACM Mobile Networks and Applications. His research focuses on mobility and resource management in interconnected wireless/wired networks, ultra wideband wireless communications networks, wireless network security, wireless body area networks, and vehicular ad hoc and sensor networks.

Dr. Shen is a Registered Professional Engineer in the Province of Ontario, and a Distinguished Lecturer of IEEE Communications Society. He received the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier's Research Excellence Award in 2003 from the Province of Ontario, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He served as the Tutorial Chair for the 2008 IEEE International Conference on Communications, the Technical Program Committee Chair for the 2007 IEEE Global Telecommunications Conference, the General Cochair for the 2007 International Conference in Communications and Networking in China and the 2006 International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, and the Founding Chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He also serves as a Founding Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the KICS/IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS. He has also served as a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, and the IEEE Communications Magazine.