Compact Stimulation Mechanism for Routing Discovery Protocols in Civilian Ad-hoc Networks

Huafei Zhu, Feng Bao, Tieyan Li

Department of Information Security, Institute for Infocomm Research, A-Star.
{huafei, baofeng, litieyan} @i2r.a-star.edu.sg

Abstract. In this paper, a refined sequential aggregate signature scheme from RSA that works for any modulus is presented, then a compact stimulation mechanism without a central, trusted authority for routing discovery in civilian ad hoc networks is proposed as an immediate application of this cryptographic primitive. Our protocol forces selfish nodes to cooperate and report actions honestly, thus enables our routing discovery protocol to resist selfish actions within our model.
Keywords: Network security, Routing discovery protocol, Sequential aggregate signature

1 Introduction

Civilian ad hoc networks have been a very attractive field of academic and industrial research in recent years due to their potential applications and the proliferation of mobile devices. Unfortunately, ad hoc networks are vulnerable and subject to a wide range of attacks due to the open medium, dynamically changing topology, possible node compromise, difficulty in physical protection, absence of infrastructure and lack of trust among nodes. As a result, nodes in these networks can be faulty/malicious or selfish. Although the problems of faulty/malicious nodes can be important in multi-authority applications, the focus of this paper is on selfish nodes. We expect that selfish nodes are the dominant type of nodes in a civilian Ad hoc network, where the nodes do not belong to a single authority and forwarding a message will incur a cost to a node, thus a selfish node will need incentive in order to forward others' messages. A series works of Michiardi and Molva [12] and [13] have already shown that a selfish behavior can be as harmful, in terms of the network throughput, as a malicious one. Consequently, practical incentive to stimulate cooperative behaviors such as forwarding each other's message in such emerging civilian applications are certainly welcome.

1.1 Related works

Incentives/stimulating cooperation is a serious issue in many protocols, including mobile ad hoc networks, peer-to-peer or overlay network systems, and even in traditional BGP Internet routing. This paper, however, is restrict to study the

incentive issues in routing discovery protocols and the incentive issues from these other fields are completely neglected. We thus sketch the following works that are closely related to this paper:

-In [14], Marti et al. proposed a reputation system for ad hoc networks. In their system, a node monitors the transmission of a neighbor to make sure that the neighbor forwards others' traffic. If the neighbor does not forward others' traffic, it is considered as uncooperative, and this uncooperative reputation is propagated throughout the network. Such reputation systems have several issues since there is no formal specification and analysis of the type of incentive provided by such systems and the system has not considered the possibility that even selfish nodes can collude with each other in order to maximize their welfare.

-Buttyan and Hubaux [4] proposed a stimulation approach that is based on a virtual currency, called nuglets, which is used as payments for packet forwarding. To implement the payment models, a tamper-proof hardware is required at each node to ensure the correct amount of nuglets is deducted or credited at each node. Besides the nuglets approach, the authors also proposed a scheme based on credit counter [5]. Although, this new scheme is simple and elegant, it still requires a tamper-proof hardware at each node so that the correct amount of credit credited or deducted.

- In [8], Jakobsson *et al.* proposed a micro-payment scheme for mobile ad hoc networks that encourages collaboration in packet forwarding by letting users benefit from relaying other's packets. The proposal is somewhat similar to [15] in that the originators of packet are charged per packet while users performing packet forwarding are paid per winning ticket. Although, the architecture for fostering collaboration is attractive, their approach is heuristic. Consequently, a less heuristic approach would be a great step forward. The recent work of Sprite (a simple, cheat-proof, credit-based system for mobile ad hoc networks [18]) can be viewed as such a forward step.

-The basic idea of Sprite is that [18]: suppose an initiator node n_0 is to send message payload m with sequence number $seq_0(0, d)$ to a destination node n_d, through path p which is generated by routing discovery protocol DSR (Dynamic Source Routing in ad hoc wireless networks [9]). Node n_0 first computes a signature s on ($H(m)$, p, $seq_0(0, d)$). Then, n_0 transfers (m, p, $seq_0(0, d)$, s) to the next hop and increases $seq_0(0, d)$. Suppose that node n_i receives (m, p, seq, s). It first checks three conditions: 1) n_i is on the path; 2) the message has a sequence number greater than $seq_i(0, d)$; and 3) the signature is valid. If any of the conditions is not satisfied, the message is dropped. Otherwise, it saves ($H(m)$, p, seq, s) as a receipt. If n_i is not the destination and decides to forward the message, it sends (m, p, seq, s) to the next hop. In order to get credit for forwarding other's messages, a node needs to report to a Credit Clearance Service (CCS) the messages it has helped forward whenever it switches to a fast connection and has backup power (to implement this idea, Sprite assumes that a mobile node can also use a desktop computer as a proxy to report to the CCS). The CCS then determines the charge and credits to each node involved in the transmission of a message, depending on the reported receipts of a message. The

contribution of Sprite lies in that they avoid assumptions on the tamper proof hardware and the receipt submission is proved cheat-proof. Sprite works well on message forwarding protocols assuming that an originator has a path connected a destination node prior to the communication. To simulate cooperation for routing discovery, the authors further proposed the following mechanism based on DSR: when a node starts to broadcast a route request, the node signs (e.g., using RSA signature scheme) and broadcasts the message, and increases its sequence number counter by 1. Suppose a node receives a route request, it first decides whether the message is a replay by looking at the sequence number. The node saves the received route request for getting payment in the future. When the node decides to rebroadcast the route request, it appends its own address to the route request and signs the extended message. In this way, the signatures' size of a routing request may grow linearly with the inputs and increase communication overheads. Thus we need a cryptographic primitive that provides the functionality of a signature scheme and at the same time reduces the overall signature sizes.

1.2 Problem statement

Normally what makes mobile ad hoc networks interesting is that they are generally operating with extremely limited memory and CPU resources. Most serious MANET protocols completely avoid public key cryptography. It is just too expensive. However, in energy limited networks, the energy consumed to compute 1000 32-bit additive operations is approximate to that of transmission of 1 bit. Thus, the communication complexity is clearly a dominate concern in energy-consumed networks. Thus, it is not surprising, many incentive based network systems are built on top of the public key cryptography, e.g., Nuglets [4] and Sprite [18]. We will follow the public key cryptography approach throughout the paper. Although the idea for designing Sprite is interesting and attractive [18], it still suffers from the problems stated below. That is

-Problem 1: In [18], the incentive system consists of a central, trusted authority called Credit Clearance Service (CCS) and a collection of mobile nodes. Each node n_i has a pair of public/secret key (PK_i, SK_i) which is certificated by a scalable certificate authority. The nodes are equipped with network interfaces that allow them to send and receive messages through a wireless overlay network, using GPRS or 3G in a wide-area environment while switching to IEEE 802.11 or Bluetooth in an indoor environment. Normally, what makes MANETs interesting is its distributed property, thus a central, trusted authority CCS may not be available. The same problem occurs also in the recent work of Martinelli, Petrocchi, and Vaccarelli [16]. As a result, any compact stimulation mechanism without a central authority is certainly welcome.

-Problem 2: The signatures' size of a routing message (request/reply) grows linearly with the inputs and increase communication overheads since each intermediate node should signs its routing messages in [18]. Thus how to reduce the line size of signatures to the constant size of is definitely an important re-

search problem (communication complexity), i.e., the signature size should be independent of the number of intermediate nodes.

1.3 Our works

At a high level, our approach to simulate cooperation for routing discovery can be addressed below:

-Stimulating cooperation in route discovery phase: we propose a new approach, called compact stimulation mechanism for routing discovery protocol to stimulate cooperation in routing discovery in an aggregate manner. This approach is based on endairA [6] and [3]. In endairA, the initiator of the route discovery process generates a route request, which contains the identifiers of the initiator and the target, and a randomly generated query identifier. Each intermediate node that receives the request for the first time appends its identifier to the route accumulated so far, and re-broadcasts the request. When the request arrives to the target, it generates a route reply. The route reply contains the identifiers of the initiator and the target, the accumulated route obtained from the request, and a digital signature of the target on these elements. The reply is sent back to the initiator on the reverse of the route found in the request. Each intermediate node that receives the reply verifies that its identifier is in the route carried by the reply, and that the preceding and following identifiers on the route belong to neighboring nodes. If these verifications fail, then the reply is dropped. Otherwise, it is signed by the intermediate node, and passed to the next node on the route (towards the initiator). When the initiator receives the route reply, it verifies if the first identifier in the route carried by the reply belongs to a neighbor. If so, then it verifies all the signatures in the reply. If all these verifications are successful, then the initiator accepts the route.

-Payment protocol for routing discovery: our payment protocol consists of two kinds of fees − on one hand, n_0 and n_d should pay SMALL amount fees to all intermediates nodes who are cooperated to establish multi-path from n_0 to n_d; on the other hand n_0 or n_d should pay LARGE amount fees to all intermediate nodes in a path which is uniquely selected by n_d since this path will be used to transform data between n_0 and n_d. Since the later case is dependent on the amount of data transmitted along the path thus we ignore this case. In the rest of our works we only consider the selfish actions in the routing discovery case.

A selfish node in civilian ad hoc networks is an economically rational node whose objective is to maximize its own welfare. As a result, a selfish node can exhibit selfish actions below:

-Type-1 selfish action: after receiving a message, the node saves a receipt but not forward the message;

-Type-2 selfish action: the destination node has received a message but does not report the receipt to the initiator;

-Type-3 selfish action: the node does not receive a message but falsely claim that it has received a message;

To protect our payment protocol from selfish actions, we force a destination node n_d to report back all participating nodes $(n_d, \cdots, n_0)$ to the initial node

n_0. Since each intermediate node n_i who helped to propagate routing request is explicitly listed in the aggregate signature, it follows that any node who contributed to discover routing will be credited (which is determined by n_0 as well as n_d, possibly with the help of other auxiliary information, say, the number of hop). If a intermediate node n_i who contributed to establish a path successfully from n_0 to n_d, does not receive its credit, it can report its witness (a valid aggregate signature from n_d to n_0) to the n_0 and then obtains its credit from n_0. In this case, n_0 will be over charged by means of the punishment policy.

In summary, the contributions of this paper are follows. We first propose a new solution framework for designing compact stimulation routing discovery protocols in civilian ad hoc networks based on our newly constructed sequential aggregate signature schemes and then show that our incentive mechanism is secure against selfish actions within our model.

2 Building block

Our compact stimulation mechanism for routing discovery protocol heavily relies on our newly constructed sequential aggregate signature scheme. The application of (sequential) aggregate signatures to other settings can be found in [1], [2], [11] and [17].

2.1 Syntax and security definition

A sequential signature scheme (KG, $\mathsf{AggSign}$, AggVf) consists of the following algorithms [11]:

-A Key generation algorithm (KG): On input 1^k, KG outputs system parameters param (including an initial value $\mathcal{IV}$, without loss of generality, we assume that $\mathcal{IV}$ is a zero strings with length l-bit), on input param and user index $i \in \mathcal{I}$, it outputs a public key and secret key pair (PK_i, SK_i) for a user i.

-Aggregate signing algorithm ($\mathsf{AggSign}$): Given a message m_i to sign, and a sequential aggregate σ_{i-1} on messages $\{m_1, \cdots, m_{i-1}\}$ under respective public keys $PK_1, \cdots, PK_{i-1}$, where m_1 is the inmost message. All of $m_1, \cdots, m_{i-1}$ and $PK_1, \cdots, PK_{i-1}$ must be provided as inputs. $\mathsf{AggSign}$ first verifies that σ_{i-1} is a valid aggregate for messages $\{m_1, \cdots, m_{i-1}\}$ using the verification algorithm defined below (if i=1, the aggregate σ_0 is taken to be zero strings 0^l). If not, it outputs $\perp$, otherwise, it then adds a signature on m_i under SK_i to the aggregate and outputs a sequential aggregate σ_i on all i messages $m_1, \cdots, m_i$.

-Aggregate verifying algorithm (AggVf): Given a sequential aggregate signature σ_i on the messages $\{m_1, \cdots, m_i\}$ under the respective public keys $\{PK_1, \cdots, PK_i\}$. If any key appears twice, if any element PK_i does not describe a permutation or if the size of the messages is different from the size of the respective public keys reject. Otherwise, for $j = i, \cdots, 1$, set $\sigma_{j-1} = \mathsf{Evaluate}(PK_1, \cdots, PK_j, \sigma_j)$. The verification of σ_{i-1} is processed recursively. The base case for recursion is $i = 0$, in which case simply check that σ_0. Accepts if σ_0 equals the zero strings.

To define the security of sequential aggregate signature scheme, we allow the adversary to play the following game [11].

-The aggregate forger $\mathcal{A}$ is provided with a initial value $\mathcal{IV}$, a set of public keys $PK_1, \cdots, PK_{i-1}$ and PK, generated at random. The adversary also is provided with $SK_1, \cdots, SK_{i-1}$; PK is called target public key.

-$\mathcal{A}$ requests sequential aggregate signatures with PK on messages of his choice. For each query, he supplies a sequential aggregate signature σ_{i-1} on some messages $m_1, \cdots, m_{i-1}$ under the distinct public keys $PK_1, \cdots, PK_{i-1}$, and an additional message m_i to be signed by the signing oracle under public key PK.

-Finally, $\mathcal{A}$ outputs a valid signature σ_i of a message m_i which is associated with the aggregate σ_{i-1}. The forger wins if $\mathcal{A}$ did not request (m_i, σ_{i-1}) in the previous signing oracle queries.

By $\mathsf{AdvAggSign}_{\mathcal{A}}$, we denote the probability of success of an adversary. We say a sequential aggregate signature scheme is secure against adaptive chosen-message attack if for every polynomial time Turing machine $\mathcal{A}$, the probability $\mathsf{AdvAggSign}_{\mathcal{A}}$ that it wins the game is at most a negligible amount, where the probability is taken over coin tosses of KG and $\mathsf{AggSign}$ and $\mathcal{A}$.

2.2 Construction and the proof of security

We further propose a refined scheme that works for any RSA moduli, and is provably secure in the sense of [11] and thus can be applied for our compact incentive routing discovery protocol. More precisely,

Let H: $\{0,1\}^* \rightarrow \{0,1\}^l$ be a cryptographic hash function and $\mathcal{IV}$ be the initial vector that should be pre-described by a sequential aggregate signature scheme. The initial value could be a random l-bit string or an empty string. Without loss of generality, we assume that the initial value $\mathcal{IV}$ is 0^l. Our sequential aggregate signature scheme is described as follows:

- Key generation: Each user i generates an RSA public key (N_i, e_i) and secret key (N_i, d_i), ensuring that $|N_i| = k_i$ and that $e_i > N_i$ is a prime. Let G_i: $\{0,1\}^{t_i} \rightarrow \{0,1\}^{k_i}$, be cryptographic hash function specified by each user i, $t_i = l - k_i$.
- Signing: User i is given an aggregate signature g_{i-1} and $(b_1, \cdots, b_{i-1})$, a sequence of messages $m_1, \cdots, m_{i-1}$, and the corresponding keys (N_1, e_1), $\cdots$, (N_{i-1}, e_{i-1}). User i first verifies σ_{i-1}, using the verification procedure below, where $\sigma_0 = 0^l$. If this succeeds, user i computes $H_i = H(m_1, \cdots, m_i, (N_1, e_1), \cdots, (N_i, e_i))$ and computes $x_i = H_i \oplus g_{i-1}$. Then it separates $x_i = y_i || z_i$, where $y_i \in \{0,1\}^{k_i}$ and $z_i \in \{0,1\}^{t_i}$, $t_i = l - k_i$. Finally, it computes $g_i = f_i^{-1}(y_i \oplus G_i(z_i)) || z_i$. By $\sigma_i \leftarrow (g_i, b_i)$, we denote the aggregate signature(if $y_i \oplus G_i(z_i) > N_i$, then b_i =1, if $y_i \oplus G_i(z_i) < N_i$, then $b_i = 0$; again we do not define the case $y_i \oplus G_i(z_i) = N_i$ since the probability the event happens is negligible), where $f_i^{-1}(y) = y^{d_i} \bmod N_i$, the inverse of the RSA function $f_i(y) = y^{e_i} \bmod N_i$ defined over the domain $Z_{N_i}^*$.

– Verifying: The verification is given as input an aggregate signature g_i, $(b_1, \cdots, b_i)$, the messages $m_1, \cdots, m_i$, the correspondent public keys (N_1, e_1), $\cdots$, (N_i, e_i) and proceeds as follows. Check that no keys appears twice, that $e_i > N_i$ is a prime. Then it computes:
  - $H_i = H(m_1, \cdots, m_i, (N_1, e_1), \cdots, (N_i, e_i))$;
  - Separating $g_i = v_i || w_i$;
  - Recovering x_i form the trapdoor one-way permutation by computing $z_i \leftarrow w_i$, $y_i = \mathcal{B}_i(f_i(v_i) + b_i N_i) \oplus G_i(z_i)$, and $x_i = y_i || z_i$, where $\mathcal{B}_i(x)$ is the binary representation of $x \in \mathcal{Z}$ (with k_i bits).
  - Recovering g_{i-1} by computing $x_i \oplus H_i$. The verification of (g_{i-1}, b_{i-1}) is processed recursively. The base case for recursion is $i = 0$, in which case simply check that $\sigma_0 = 0^l$.

Lemma [17]: The sequential aggregate signature scheme described above is provable secure in the sense of [11] in the random oracle model.

2.3 Comparison and open problem

We compare our sequential aggregate signature schemes with Kawauchi, Komano, Ohta and Tada's (KKOT) scheme [10], and Lysyanskaya et al's scheme [11] below;

-All three signatures are based on the hardness of RSA problem. For the i-th users, each signing processing needs one exponent computation while the verification processing needs $(i-1)$ exponent computations. Thus all three schemes have approximate computational complexity;

-Lysyanskaya et al's first scheme can be viewed as improvement of of KKOT scheme [10]. The restriction of modulus in the KKOT's scheme $|N_i|$-$|N_{i-1}| = 1 + k_1 + k_2$ is replaced by users's moduli to be arranged in increasing order: $N_1 < N_2 < \cdots < N_t$ in Lysyanskaya et al's scheme.

-The second approach of Lysyanskaya et al's scheme does not require the modulus to be arranged in increasing order, however they are required to be of the same length. The signature will expanded by n bits $b_1, \cdots, b_n$, where n is the total number of users. Namely, during signing, if $\sigma_i \geq N_{i+1}$, let $b_i = 1$; else, let $b_i = 0$. In our scheme, the modulus are not required to be of the same length. We emphasize that in our scheme N_i is chosen by each user independently, thus our construction is the first scheme from RSA that works for any modulus. However as Lysyanskaya et al's scheme, our sequential aggregate signature is expanded by n bits $b_1, \cdots, b_n$, where n is the total number of users.

Following from the above discussion, we here present an interesting open problem: can we propose a sequential aggregate signature scheme such that N_i is chosen by each user independently, and at the same time no single bit of signature size will be expanded?

3 Stimulating cooperation for routing discovery

In this section, we propose a compact stimulation mechanism for routing discovery in ad hoc networks. Suppose that a source node n_0 sends a routing request

message $RREQ$ (it may include the maximum number of hops that is allowed to reach the destination node) to the destination n_d, where $RREQ$ is formatted by the endairA protocol (this protocol has nice features, for example, it ensures that once a path is outputted by endairA, it is always correct one). When the initiator n_0 starts to broadcast a route request, it signs and broadcasts route request message by an ordinary signature scheme specified [1]. When an intermediate node n_i decides to rebroadcast the routing request message if the signature of $RREQ$ is valid, it then appends its own address/identity to the received routing request and then rebroadcasts the $RREQ$ until the the destination node n_d is reached.

When multi routing pathes associated with the original request messages $RREQ$ arrive to the target node, it chooses a proper route path (e.g., with least hope number). Then it sends back the route reply to the initiator node. Each routing reply message $RREP$ contains identifiers of the initiator and the target, as well as that of all intermediate nodes, together with a sequential aggregate signature on $RREP$ which starts to sign from the target node n_d. Each intermediate node that receives the reply first check that its identifier is listed in the $RREP$ and then verifies the correctness of the received sequential aggregate signature on the message $RREP$. If both checks are valid, the message $RREQ$ is further signed by this intermediate node using its own secret key, and then send it to its successive node; Otherwise, the reply is dropped. When the initiator receives the route reply message, it verifies the correctness of sequential aggregate signature scheme, and if the verification is successful, then the initiator accepts the route and then pays the credential to each intermediate node according to its payment strategy [2].

3.1 Secure against selfish actions

We now consider three types of selfish actions in our routing discovery protocol below:

-In the Type-1 selfish action, a node say n_i saves a valid aggregate signature (the receipt or the witness) but does not sign and forward the signature. In this case, each node along the path cannot be credited as there is no actual routing from n_d to n_0 (and hence from n_0 to n_d is established), thus violates the selfish action of n_i.

-In the Type-2 selfish action, the destination node n_d has received a valid $RREQ$ from n_{d-1}, but it does not send back the routing reply message $RREP$ (including its signature to $RREQ$) to the initiator n_{d-1}. In this case, there is no routing path available from n_0 to n_d. Thus, such a selfish behavior is completely avoided by n_d unless n_d refuses to receive any message from n_0.

[1] To resist DoS attack, we assume that the initiator signs the routing request message $RREQ$ and the intermediate nodes to verify this signature. Notice that the ordinary signature of $RREQ$ can be absorbed by the underlying sequential aggregate signature

[2] How to specify the payment strategy is a complex issue, possibly the credential may be related to n_0 and n_d as well as the number of hop in a given routing, however we ignore the details of the payment protocol in this paper.

-In the Type-3 selfish action, a node n_i does not receive a message but falsely claims it has signed and forwarded the aggregate signature to its successor; In this case the identity of n_i is not listed in the routing reply message $RRER$. Since the underlying sequential aggregate signature scheme is secure in the sense of [11], the selfish node can forge a valid signature with at most negligible amount. Thus, the selfish action of can be captured.

In summary, we have the desired statement − assuming the underlying sequential aggregate signature scheme is secure in the sense of [11], our routing discovery is secure against selfish actions defined in Section 1.

3.2 Unsolved problems

Notice that the reply attack does not work in our setting. The reply attack means that n_i stores a $RREP$ and reuses it later when it receives a new $RREQ$ and n_0 is fooled to think there exists a path with n_d. Since a $RREP$ message in our formate must contain the request message $RREQ$ and its signature which is signed by n_0. As a result, the replay attack does not work in our model. However we should point out the fact that our routing discovery protocol does not resist a selfish node n_i to introduce more intermediate nodes in a routing path explicitly. We therefore classify two potential selfish actions below.

-Greedy attack: for example, instead n_i broadcasts and forwards the $RREQ$ to n_{i+1}, it may intended to insert a set of redundant nodes, say $n_{i,1}$, $n_{i,2}$, $\cdots$, $n_{i,k}$, between n_i and n_{i+1}. This selfish action does not always work since the destination node is allowed to choose a path with less hop number. Thus, a remedy scheme maybe insert the maximum hop number in each $RREQ$ message. This is a possible solution to resist such a greedy attack.

-Collude attack: for example, a intermediate node n_{i-1}, n_i and the destination node n_d are collude to foolish n_0. In this case, n_d intends to insert a set of redundant intermediate nodes $n_{i,1}$, $n_{i,2}$, $\cdots$, $n_{i,k}$ between n_{i-1} and n_i. This collude is powerful and our routing discovery protocol fails in such an attack.

To best of our knowledge, all incentive based routing discovery protocol, say, [16] and [18] also suffer from the above attacks, we thus leave two open problems to the research community.

4 Conclusion

In this paper, we have presented a new solution to improve incentive-compatible routing discovery protocols in civilian networks based on our sequential aggregate signature scheme and have shown that our compact stimulation mechanism for routing discovery protocol resist certain selfish actions within our model.

References

1. Dan Boneh, Craig Gentry, Ben Lynn, Hovav Shacham: Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. EUROCRYPT 2003: 416-432.

2. Dan Boneh, Craig Gentry, Ben Lynn, Hovav Shacham: A Survey of Two Signature Aggregation Techniques. In CryptoBytes Vol. 6, No. 2, 2003.
3. L. Buttyán and I. Vajda, Towards Provable Security for Ad Hoc Routing Protocols, 2nd ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN 2004) Washington DC, USA, October 25, 2004.
4. L. Buttyán and J. P. Hubaux. Enforcing service availability in mobile ad hoc WANs, in IEEE/ACM Workshop on Mobile Ad hoc Networking and Computing (MobiHOC), Boston, MA, August 2000.
5. L. Buttyán and J. P. Hubaux, Stimulating cooperation in self-organizing mobile Ad hoc networks, ACM Journal for Mobile Networks (MONET), special issue on Mobile Ad hoc Networks, summer 2002.
6. Yih-Chun Hu, Adrian Perrig, David B. Johnson: Ariadne: a secure on-demand routing protocol for ad hoc networks. MOBICOM 2002: 12-23
7. J. Coron: On the Exact Security of Full Domain Hash. CRYPTO 2000: 229-235
8. M. Jakobsson, J. P. Hubaux, and L. Buttyan. A micropayment scheme encouraging collaboration in multi-hop cellular networks, in Proceedings of Financial Crypto 2003, La Guadeloupe, January 2003.
9. D. B. Johnson and D. A. Malt, Mobile Computing. Kluwer Academic Publishers, 1996, Dynamic Source Routing in Ad hoc Wireless Networks, Chapter 5.
10. K. Kawauchi, Y. Komano, K. Ohta and M. Tada: Probabilistic multi-signature schemes using a one-way trapdoor permutation, IEICE transactions on fundamentals, vol.E87-A, no5, pp.1141-1153, 2004.
11. Anna Lysyanskaya, Silvio Micali, Leonid Reyzin, Hovav Shacham: Sequential Aggregate Signatures from trapdoor one-way permutations. EUROCRYPT 2004: 74-90.
12. P. Michiardi, and R. Molva. Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks. In Proc. of CMS'02.
13. P.Michiardi, and R. Molva. A Game Theoretical Approach to Evaluate Cooperation Enforcement Mechanisms in Mobile Ad hoc Networks. In Proc. of WiOpt03 of the IEEE Computer Society.
14. S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile Ad hoc networks, in Proceedings of The Sixth International Conference on Mobile Computing and Networking 2000, Boston, MA, Aug. 2000.
15. Silvio Micali, Ronald L. Rivest: Micropayments Revisited. CT-RSA 2002: 149-163.
16. F. Martinelli, M. Petrocchi, and A. Vaccarelli, Local management of credits and debits in mobile ad hoc networks. Conference on Communications and Multimedia Security, CMS 2004.
17. Huafei Zhu et al, Constructing Sequential Aggregate Signatures for Secure Wireless Routing Protocols, IEEE WCNC'05, New Orleans, 13-17 March, 2005, New Orleans, LA, USA.
18. Sheng Zhong, Jiang Chen, and Yang Richard Yang. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad hoc Networks.Proceedings of IEEE INFOCOM '03, San Francisco, CA, April 2003.