SAT: A Security Architecture Achieving Anonymity and Traceability in Wireless Mesh Networks

Jinyuan Sun, *Member*, *IEEE*, Chi Zhang, *Student Member*, *IEEE*, Yanchao Zhang, *Member*, *IEEE*, and Yuguang Fang, *Fellow*, *IEEE*

Abstract—Anonymity has received increasing attention in the literature due to the users' awareness of their privacy nowadays. Anonymity provides protection for users to enjoy network services without being traced. While anonymity-related issues have been extensively studied in payment-based systems such as e-cash and peer-to-peer (P2P) systems, little effort has been devoted to wireless mesh networks (WMNs). On the other hand, the network authority requires conditional anonymity such that misbehaving entities in the network remain traceable. In this paper, we propose a security architecture to ensure unconditional anonymity for honest users and traceability of misbehaving users for network authorities in WMNs. The proposed architecture strives to resolve the conflicts between the anonymity and traceability objectives, in addition to guaranteeing fundamental security requirements including authentication, confidentiality, data integrity, and nonrepudiation. Thorough analysis on security and efficiency is incorporated, demonstrating the feasibility and effectiveness of the proposed architecture.

Index Terms—Anonymity, traceability, pseudonym, misbehavior, revocation, wireless mesh network (WMN).

1 Introduction

Wireless Mesh Network (WMN) is a promising technology and is expected to be widespread due to its low-investment feature and the wireless broadband services it supports, attractive to both service providers and users. However, security issues inherent in WMNs or any wireless networks need be considered before the deployment and proliferation of these networks, since it is unappealing to subscribers to obtain services without security and privacy guarantees. Wireless security has been the hot topic in the literature for various network technologies such as cellular networks [1], wireless local area networks (WLANs) [2], wireless sensor networks [3], [4], mobile ad hoc networks (MANETs) [5], [6], and vehicular ad hoc networks (VANETs) [7]. Recently, new proposals on WMN security [8], [9] have emerged. In [8], the authors describe the specifics of WMNs and identify three fundamental network operations that need to be secured. We [9] propose an attack-resilient security architecture (ARSA) for WMNs, addressing countermeasures to a wide range of attacks in WMNs. Due to the fact that security in WMNs is still in its infancy as very little attention has been devoted so far [8], a majority of security issues have not been addressed and are surveyed in [10].

Anonymity and privacy issues have gained considerable research efforts in the literature [7], [9], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], which have focused on investigating anonymity in different context or application scenarios. One requirement for anonymity is to unlink a user's identity to his or her specific activities, such as the anonymity fulfilled in the untraceable e-cash systems [11], [13] and the P2P payment systems [12], [14], where the payments cannot be linked to the identity of a payer by the bank or broker. Anonymity is also required to hide the location information of a user to prevent movement tracing, as is important in mobile networks [15], [16], [17] and VANETs [7]. In wireless communication systems, it is easier for a global observer to mount traffic analysis attacks by following the packet forwarding path than in wired networks. Thus, routing anonymity [18], [19], [20], [21] is indispensable, which conceals the confidential communication relationship of two parties by building an anonymous path between them. Nevertheless, unconditional anonymity may incur insider attacks since misbehaving users are no longer traceable. Therefore, traceability is highly desirable such as in e-cash systems [11], [13], where it is used for detecting and tracing double-spenders.

In this paper, we are motivated by resolving the above security conflicts, namely anonymity and traceability, in the emerging WMN communication systems. We have proposed the initial design of our security architecture in [22], where the feasibility and applicability of the architecture were not fully understood. As a result, we provide detailed efficiency analysis in terms of storage, communication, and computation in this paper to show that our SAT is a practically viable solution to the application scenario of

- *J. Sun is with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996. E-mail: jysun@eecs.utk.edu.*
- *C. Zhang and Y. Fang are with the Department of Electrical and Computer Engineering, University of Florida, PO Box 116130, Gainesville, FL 32611. E-mail: zhangchi@ufl.edu, fang@ece.ufl.edu.*
- *Y. Zhang is with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287. E-mail: yczhang@asu.edu.*

Manuscript received 4 Sept. 2008; revised 16 May 2009; accepted 7 Sept. 2009; published online 4 Dec. 2009.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-2008-09-0140. Digital Object Identifier no. 10.1109/TDSC.2009.50.

interest. Our system borrows the blind signature technique from payment systems [11], [12], [14], [23], and hence, can achieve the anonymity of unlinking user identities from activities, as well as the traceability of misbehaving users. Furthermore, the proposed pseudonym technique renders user location information unexposed. Our work differs from previous work in that WMNs have unique hierarchical topologies and rely heavily on wireless links, which have to be considered in the anonymity design. As a result, the original anonymity scheme for payment systems among bank, customer, and store cannot be directly applied. In addition to the anonymity scheme, other security issues such as authentication, key establishment, and revocation are critical in WMNs to ensure the correct application of the anonymity scheme. Moreover, although we employ the widely used pseudonym approach to ensure network access anonymity and location privacy, our pseudonym generation does not rely on a central authority, e.g., the broker in [9], the domain authority in [15], the transportation authority or the manufacturer in [7], and the trusted authority in [18], who can derive the user's identity from his pseudonyms and illegally trace an honest user. Note that our system is not intended for achieving routing anonymity, which can be incorporated as an enhancement.

Specifically, our major contributions in this paper include 1) design of a ticket-based anonymity system with traceability property; 2) bind of the ticket and pseudonym which guarantees anonymous access control (i.e., anonymously authenticating a user at the access point) and simplified revocation process; 3) adoption of the hierarchical identity-based cryptography (HIBC) for interdomain authentication avoiding domain parameter certification.

The rest of the paper is organized as follows: Section 2 introduces some preliminaries. The system model including the network architecture and trust model is described in Section 3. Section 4 elaborates on the ticket-based anonymity scheme, which is the key component of our security architecture. Security analysis, efficiency analysis, and possible enhancements pertinent to the proposed architecture are presented in Sections 5, 6, and 7, respectively. Finally, Section 8 concludes the paper.

2 Preliminaries

2.1 IBC from Bilinear Pairings

ID-based cryptography (IBC) allows the public key of an entity to be derived from its public identity information such as name and e-mail address, which avoids the use of certificates for public key verification in the conventional public key infrastructure (PKI) [24]. Boneh and Franklin [25] introduced the first functional and efficient ID-based encryption scheme based on bilinear pairings on elliptic curves. Specifically, let G_1 and G_2 be an additive group and a multiplicative group, respectively, of the same prime order p. The Discrete Logarithm Problem (DLP) is assumed to be hard in both G_1 and G_2. Let P denote a random generator of G_1 and $e : G_1 \times G_1 \rightarrow G_2$ denote a bilinear map constructed by modified Weil or Tate pairing with the following properties:

1. Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$, $\forall P, Q \in G_1$, and $\forall a, b \in Z_p^*$, where Z_p^* denotes the multiplicative group of Z_p, the integers modulo p. In particular, $Z_p^* = \{x \mid 1 \leq x \leq p-1\}$ since p is prime.
2. Nondegenerate: $\exists P, Q \in G_1$ such that $e(P, Q) \neq 1$.
3. Computable: there exists an efficient algorithm to compute $e(P, Q), \forall P, Q \in G_1$.

2.2 Blind Signature

Blind signature is first introduced by Chaum [23]. In general, a blind signature scheme allows a receiver to obtain a signature on a message such that both the message and the resulting signature remain unknown to the signer. We refer the readers to [26] for a formal definition of a blind signature scheme, which should bear the properties of verifiability, unlinkability, and unforgeability according to [23].

Brands [27] developed the first restrictive blind signature scheme, where the restrictiveness property is incorporated into the blind signature scheme such that the message being signed must contain encoded information. As the name suggests, this property restricts the user in the blind signature scheme to embed some account-related secret information into what is being signed by the bank (otherwise, the signing will be unsuccessful) such that this secret can be recovered by the bank to identify a user if and only if he double-spends. The restrictiveness property is essentially the guarantee for traceability in the restrictive blind signature systems. Partial blind signature schemes [28], [29] allow the resulting signature to convey publicly visible information on common agreements between the signer and the signee. This is useful when certain information in the signature needs to be reviewed by a third party. One example is the common agreements, the visibility of which enables the intermediate parties who examine the signature to check the compliance of the signee to the items specified in the agreements, before proceeding to the verification of the signature and other operations. Restrictive partially blind signature schemes [30], [31], [32] were derived from the aforementioned work. They are essentially blind signature schemes with restrictiveness and partial blindness properties. In the restrictive partially blind signature schemes [31], [32] that serve as a building block for our architecture, the two key concepts, namely restrictiveness and partial blindness, are defined based on [11], [28].

Restrictiveness. Let a message m be such that the receiver knows a representation $(a_1, \ldots, a_k)$ of m with respect to a generator tuple $(g_1, \ldots, g_k)$ at the beginning of a blind signature protocol. Let $(b_1, \ldots, b_k)$ be the representation the receiver knows of the blinded message m' of m after the completion of the protocol. If there exist two functions I_1 and I_2 such that $I_1(a_1, \ldots, a_k) = I_2(b_1, \ldots, b_k)$, regardless of m and the blinding transformations applied by the receiver, then the protocol is called a restrictive blind signature protocol. The functions I_1 and I_2 are called blinding-invariant functions of the protocol with respect to $(g_1, \ldots, g_k)$.

Partial Blindness. A signature scheme is partially blind if, for all probabilistic polynomial-time algorithm $\mathcal{A}$, $\mathcal{A}$ wins the game in the signature issuing protocol with probability at most $\frac{1}{2} + \frac{1}{k^c}$ for sufficiently large k and some constant ϵ. The probability is taken over coin flips of $\mathcal{KG}$, U_0, U_1, and $\mathcal{A}$, where $\mathcal{KG}$ is the key generation function defined in [32], U_0 and U_1 are two honest users following the signature issuing protocol. Due to the space limitation, interested readers are referred to [32] for complete description of the game in the signature issuing protocol.

3 System Model

3.1 Notation and Definitions

First, we give a list of notation and definitions that are frequently used in this paper.

3.1.1 Notation

- $\rightarrow$, $\rightarrow\!\!\rightarrow$, and $\|$: denote single-hop communications, multihop communications, and concatenation, respectively.
- CL, MR, GW, and TA: abbreviations for client, mesh router, gateway, and trusted authority, respectively.
- ID_x: the real identity of an entity x in our WMN system.
- PS_x: the pseudonym self-generated by a client x by using his real identity ID_x.
- $H_1(M)$ and $H_1'(M)$: $\{0,1\}^* \rightarrow G_1$, cryptographic hash functions mapping an arbitrary string M to G_1.
- H_2: a cryptographic secure hash function: $G_1^3 \times G_2^5 \rightarrow Z_p^*$.
- H_3: a cryptographic secure hash function: $G_2 \times G_2 \times ID_{GW} \times date/time \rightarrow Z_p^*$.
- $H_1(ID_x)/\Gamma_x$ and $H_1(IDT_x)/\psi_x$: the public/private key pairs assigned to an entity x in the standard IBC and HIBC, respectively.
- $PS_x/\widetilde{\Gamma_x}$ and $PST_x/\widetilde{\psi_x}$: the self-generated pseudonym/private key pairs based on the above public/private key pairs.
- $\mathcal{SIG}_{\Gamma_x}(m)$: the ID-based signature on a message m using the signer x's private key Γ_x.
- $\mathcal{VER}(\mathcal{SIG})$: the verification process of the above signature, which returns "accept" or "reject."
- $\mathcal{HIDS}_{\psi_x,s_x}(m)$: the hierarchical ID-based signature on a message m generated by the signer x using its secret point ψ_x and secret number s_x for interdomain authentication.
- $\mathcal{HVER}(\mathcal{HIDS}, QT)$: the verification process using the above $\mathcal{HIDS}$ and QT, which returns "accept" or "reject."
- $\mathcal{SKE}_\kappa(D)$: the symmetric key encryption on plaintext D using the shared secret key κ.
- $\mathcal{HMAC}_\kappa(m)$: the keyed-hash message authentication code on a message m using cryptographic hash functions and the symmetric key κ.

3.1.2 Definitions

- *Anonymity (Untraceability)*: the anonymity of a legitimate client refers to the untraceability of the client's network access activities. The client is said to be anonymous if the TA, the gateway, and even the collusion of the two cannot link the client's network access activities to his real identity.
- *Traceability*: a legitimate client is said to be traceable if the TA is able to link the client's network access activities to the client's real identity *if and only if* the client misbehaves, i.e., one or both of the following occurs: ticket reuse and multiple deposit.
- *Ticket reuse*: one type of misbehavior of a legitimate client that refers to the client's use of a depleted ticket ($val = 0$).

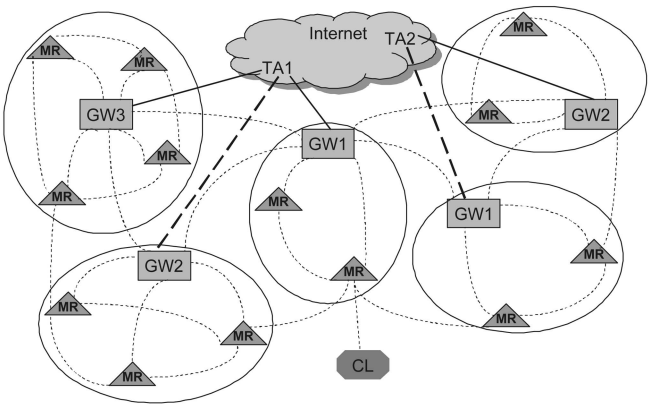


Fig. 1. Network topology of a typical WMN.

- *Multiple deposit*: one type of misbehavior of a legitimate client that refers to the client's disclosure of his valid ticket and associated secrets to unauthorized entities or clients with misbehavior history, so that these coalescing clients can gain network access from different gateways simultaneously.
- *Collusion*: the colluding of malicious TA and gateway to trace a legitimate client's network access activities in the TA's domain (i.e., to compromise the client's anonymity).
- *Framing*: a type of attack mounted by a malicious TA in order to revoke a legitimate client's network access privilege. In this attack, the TA can generate a false account number and associate it with the client's identity. The TA can then create valid tickets based on the false account number and commit fraud (i.e., misbehave). By doing so, the TA is able to falsely accuse the client to have misbehaved, and thus, to revoke his access right.

3.2 Network Architecture

Consider the network topology of a typical WMN depicted in Fig. 1. The wireless mesh backbone consists of mesh routers (MRs) and gateways (GWs) interconnected by ordinary wireless links (shown as dotted curves). Mesh routers and gateways serve as the access points of the WMN and the last resorts to the Internet, respectively. The hospital, campus, enterprise, and residential buildings are instances of individual WMN domains subscribing to the Internet services from upstream service providers, shown as the Internet cloud in Fig. 1. Each WMN domain, or trust domain (to be used interchangeably) is managed by a domain administrator that serves as a trusted authority (TA), e.g., the central server of a campus WMN. The TA and associated gateways are connected by high-speed wired or wireless links, displayed as solid and bold dashed lines, respectively. TAs and gateways are assumed to be capable of handling computationally intensive tasks. In addition, they are assumed to be protected in private places and cannot be easily compromised due to their important roles in the WMN. The WMNs of interest here are those where the TA provides free Internet access but requires the clients (CLs) to be authorized and affiliated members generally for a long term, as the employees or students in the case of enterprise and hospital WMNs or campus WMNs. Such individual

WMN domains can be building blocks of an even larger metropolitan WMN domain.

3.3 Trust Model

The trust model comprising trust relationships and the trust domain initialization will be described in this section.

3.3.1 Trust Relationship

In general, the TA is trusted within the WMN domain. There is no direct trust relationship between the client and the gateway/mesh router. We will use standard IBC for authentication and secure communications both at the backbone and during network access inside a trust domain (i.e., intradomain). We further assume the existence of preshared keys and secure communication channels between entities (TAs, gateways, mesh routers) at the backbone and will solely consider the authentication and key establishment during the network access of the clients.

The client presents his ID upon registration at the TA, which assigns a private key associated with the client's ID. The client selects a unique account number Ω computed by a randomly chosen secret number u_1 (cf., Section 4.1.1). The account number is stored with the client's ID at the TA. The TA also assigns an ID/private key pair to each gateway and mesh router in its trust domain before deployment. Advantages of this general trust relationship with the TA stem from the direct authentication of the clients traveling among gateways/mesh routers in the same domain, which reduces network access latency and communication overhead that is expected to be overwhelming in future WMNs due to the large user population and high mobility.

In accordance with the natural hierarchical architecture of the WMNs considered in this paper, we adopt the hierarchical ID-based signature scheme (HIDS) for interdomain authentication that occurs when a client affiliated with the home TA visits neighboring foreign TAs. Note that the basic HIDS [33] is suitable when the level m of the signer in the hierarchical tree (HT) is close to the root at level 0, since the number of pairing operations and the size of the signature are determined by the signer's absolute location m. If m is relatively high (i.e., the signer is located deep down the HT), the basic HIDS can be very inefficient in terms of computation and communication. In this case, Dual-HIDS [33] is more suitable if the signer and verifier share a common ancestor at level l below the root, since the number of pairing operations and the size of the signature are determined by the signer's relative location $m - l$, to the common ancestor. For instance, the two TAs in Fig. 1 can be the domain administrators of neighboring campuses or hospitals directly managed by the State Department of Education (SDE), or the State Department of Health (SDH), etc. For ease of demonstration, we use the basic HIDS for interdomain authentication in this paper. Let the SDE (or SDH) be the root at level 0 in the HT of the campus (or hospital) WMN. All the TAs in the SDE's domain are at level 1 and all gateways, mesh routers and clients in each TA's domain are at level 2. Note that in reality, the campus (or hospital) WMN may be part of the HT of a larger WMN (i.e., the SDE or SDH is a child at level n below the root). However, as long as the signer's relative location to the common ancestor of the signer/verifier pair in the HT remains unchanged, the Dual-HIDS scheme can be employed instead.

In the WMN architecture in [9], we handled a similar interdomain authentication issue with a different approach. When a client roams to a foreign TA's domain (FTD) with a different master secret, we propose to get the foreign TA's domain parameters certified by a trusted third party (TTP). The domain parameter certificate (DPC) issued by the TTP is then included in the interdomain authentication for verifying the authenticity of the domain parameters, which will later be utilized to verify the signature from the entities in FTD. Compared to that approach, the adopted HIDS scheme eliminates the requirement for the TTP and the DPCs. Furthermore, since we are concerned with the computation power of the clients, using the level assignment (levels 0-2) mentioned in the example above, the client needs to compute four pairings for verifying the signature from the access point (mesh router or gateway). In [9], the client needs also to compute four pairings, two for DPC validation and two for verifying the signature from the access point if the efficient Hess's ID-based signature [34] is used. Thus, the adopted HIDS scheme does not compromise the computation efficiency while increases the communication efficiency by the avoidance of DPCs. We argue that the computational complexity of HIDS for the WMN architecture considered here is acceptable since the client is most frequently roaming within the home domain, where the standard IBC is used.

3.3.2 Trust Domain Initialization

We apply the domain initialization of the hierarchical IBC [33]. Specifically, the root public key generator (PKG) at level 0 in the HT performs the following domain initialization algorithm when the network is bootstrapped, where P_0 is a generator of G_1:

1. Input security parameter $\xi \in Z^+$ into domain parameter generator $\mathcal{PG}$ and output the parameter tuple $(p, G_1, G_2, e, P_0, H_1)$.
2. Randomly select a domain master secret $s_0 \in Z_p^*$ and calculate the domain public key $\overline{P_{pub}} = s_0 P_0$.

The root PKG (e.g., the SDE or SDH) publishes the domain parameters $(p, G_1, G_2, e, P_0, H_1, \overline{P_{pub}})$ and maintains s_0 confidential. Suppose that a child CH_j is located at level j. The lower level setup is performed by the parent as follows:

1. compute $K_j = H_1(ID_1, ..., ID_j)$;
2. compute CH_j's private keys $\psi_j = \psi_{j-1} + s_{j-1}K_j = \sum_{i=1}^{j} s_{i-1}K_i$, and $\Gamma_j = \pi H_1(ID_j)$; and
3. distribute $QT = \{Q_l : 1 \le l < j\}$ to CH_j, where $Q_l = s_l P_0$.

In the above private key assignment, $(ID_1, \ldots, ID_i)$ for $1 \le i \le j$ is the ID tuple of CH_j's ancestor at level i. The private keys ψ_j and Γ_j are generated for the interdomain and intradomain authentication, respectively, where s_{j-1} is the parent's secret and π is the master secret of the trust domain manager (i.e., TA in this paper). In Fig. 1, TA1 is the parent of all the entities in its domain, which is located at level 1. The entities (gateways, mesh routers, clients) are TA1's children at level 2. Similarly, the SDE or SDH (root PKG in our simple illustration) at level 0 is the parent of TA1. Note that due to the hardness of DLP, it is not possible to solve

for s_{j-1} or π given any private key calculated from them with nonnegligible probability.

4 SAT Security Architecture

4.1 Ticket-Based Security Architecture

First, we restrict our discussion to within the home domain. The interdomain protocols in our security architecture, which are executed when the client roams outside his home domain, will be presented in Section 4.1.5. The ticket-based security architecture consists of ticket issuance, ticket deposit, fraud detection, and ticket revocation protocols. In what follows, we will describe these protocols in detail, together with the fulfillment of authentication, data integrity, and confidential communications that may take place during the execution of these protocols.

4.1.1 Ticket Issuance

In order to maintain security of the network against attacks and the fairness among clients, the home TA may control the access of each client by issuing tickets based on the misbehavior history of the client, which reflects the TA's confidence about the client to act properly. Ticket issuance occurs when the client initially attempts to access the network or when all previously issued tickets are depleted. The client needs to reveal his real ID to the TA in order to obtain a ticket since the TA has to ensure the authenticity of this client. Moreover, the TA should be unable to link the ticket it issued to the clients' real identities. The client thus employs some blinding technique to transform the ticket to be unlinkable to a specific execution of the ticket generation algorithm (the core of ticket issuance protocol), while maintaining the verifiability of the ticket. The ticket generation algorithm, which can be any restrictive partially blind signature scheme in the literature, takes as input the client's and TA's secret numbers, the common agreement c, and some public parameters, and generates a valid ticket $ticket = \{T_N, W, c, (U', V', X', \rho, \sigma_1', \sigma_2')\}$ at the output, where T_N is the unique serial number of the ticket that can be computed from the client's account number Ω (cf., Section 4.1.1), $(U', V', X', \rho, \sigma_1', \sigma_2')$ is the signature on (T_N, W, c) where W is necessary for verifying the validity of the signature in the ticket deposit protocol. We opt for a partially restrictive blind signature scheme with two desired features: partial blindness and restrictiveness (cf., Section 2.2), for the proposed WMN framework. Partially blind signatures alone allow the blind signature to carry explicit information on commonly agreed terms (i.e., ticket value, expiry date, misbehavior, etc.) which remains publicly visible regardless of the blinding process. Restrictive blind signatures place restrictions on the client's selection of messages being signed which contain encoded identity information (in T_N) instead of completely random numbers, allowing the TA to recover the client's identity by computing Ω *if and only if* misbehavior is detected. As a result, the anonymity of an *honest* client is unconditionally ensured. Restrictive partially blind signature schemes [31], [32] can be adopted as the building block of the ticket generation algorithm in our ticket issuance protocol.

The TA publishes the domain parameters to be used within its trust domain as $(p, G_1, G_2, e, P, P_1, P_2, H_1, H_2, H_3, P_{pub})$ using the standard IBC domain initialization, where (P, P_1, P_2) are random generators of G_1, and $P_{pub} = \pi P$. Since the scheme in [32] is selected for demonstration, G_1 here should be a Gap Diffie-Hellman (GDH) group [35], where the computational Diffie-Hellman problem (CDHP) [35] is assumed to be intractable. In addition, the TA chooses $r \in_R Z_p^*$ and $Q \in_R G_1$, and the client chooses $\alpha, \beta, \gamma, \tau, \lambda, \mu, \rho \in_R Z_p^*$. Note that if the scheme in [31] is adopted, the TA publishes $(p, G_1, G_2, e, g, g_1, g_2, H, H_0, H_1)$, where G_1 should be a GDH in which the RCDHP (reversion CDHP) is assumed to be intractable (refer to [31] for detailed definitions). We will demonstrate the following protocols based on the scheme in [32]. The application of the scheme in [31] to our protocols is straightforward following a similar procedure. The ticket issuance protocol is demonstrated as follows:

1. $CL \rightarrow\rightarrow TA$: $ID_{CL}, m, t_1, \mathcal{HMAC}_\kappa(m \parallel t_1)$;
2. $$TA \rightarrow\rightarrow CL : ID_{TA},\ X = e(m, \Gamma_{TA}),\ Y = e(P, Q),\ Z = e(m, Q),\ U = rH_1(ID_{TA}),\ V = rP, t_2, \mathcal{HMAC}_\kappa(X \parallel Y \parallel Z \parallel U \parallel V \parallel t_2);$$
3. $$CL \rightarrow\rightarrow TA : ID_{CL},\ B = \frac{1}{\lambda} H_2(m' \parallel U' \parallel V' \parallel R \parallel W \parallel X' \parallel Y' \parallel Z') + \mu, t_3, \mathcal{HMAC}_\kappa(B \parallel t_3);\ \text{and}$$
4. $$TA \rightarrow\rightarrow CL : ID_{TA},\quad \sigma_1 = Q + B\Gamma_{TA},\ \sigma_2 = (r + B)\Gamma_{TA} + rH_1(c), t_4,\ \mathcal{HMAC}_\kappa(\sigma_1 \parallel \sigma_2 \parallel t_4).$$

At the end, the client checks if the following equalities hold: $e(P, \sigma_1) = y^B Y$ and $e(m, \sigma_1) = X^B Z$, where $y = e(P_{pub}, H_1(ID_{TA}))$. If the verification succeeds, the client calculates $\sigma_1' = \gamma\sigma_1 + \tau H_1(ID_{TA})$, $\sigma_2' = \lambda\sigma_2$, $\rho = \gamma B$, and outputs the signature $(U', V', X', \rho, \sigma_1', \sigma_2')$ on (T_N, W, c), where $T_N = m'$. In Step 3 above, $m = u_1 P_1 + u_2 P_2 = \Omega + u_2 P_2 \neq 0$, where $u_1 \in_R Z_p^*$ and

$$u_2 = 1;\quad m' = \alpha m,\quad U' = \lambda U + \lambda\mu H_1(ID_{TA}) - \beta H_1(c),\ V' = \lambda V + \beta P_{pub},\quad R = e(m', H_1(ID_{TA})),\quad W = g_1^{v_1} g_2^{v_2},$$

where $g_1 = e(P_1, H_1(ID_{TA})), g_2 = e(P_2, H_1(ID_{TA}))$, and $v_1, v_2 \in_R Z_p^*$; $X' = X^\alpha$, $Y' = Y^\gamma g^\tau$, where $g = e(P, H_1(ID_{TA}))$, $Z' = Z^{\alpha\gamma} R^\tau$. In the above protocol, the TA and the client can locally derive a symmetric key $\kappa = e(\Gamma_{TA}, H_1(ID_{CL}))$, and $\kappa = e(H_1(ID_{TA}), \Gamma_{CL})$ [36], respectively, assuming that ID_{TA} is known to all the entities in the TA's domain. A time stamp t_i is included in each message exchanged to prevent the message replay attack [37]. Note that some pairings such as the those for g_1, g_2, and g in the above procedure can be precomputed once and stored for all future use, thus, alleviating the computation burden of the client.

A design issue to be pointed out is the commonly agreed information c negotiated at the beginning of the ticket generation algorithm. We define c as $(val, exp, misb)$, where val, exp, and $misb$ denote the ticket value, expiry date/time, and the client's misbehavior level, respectively. The ticket value confines the total amount of traffic that the client is allowed to generate and receive before the expiry date of the ticket. Tickets bear different values. The value val is issued by the TA and will be deducted by the gateway in the ticket deposit protocol. The client's $misb$ field conveys information on the misbehavior history of the client in the network. This information is summarized at the TA by performing the fraud detection based on the ticket records reported by gateways that have serviced this client. By placing the misbehavior information in c, the TA successfully informs gateways about the client's past misbehavior when the ticket is deposited. Note that the presence of misbehavior information in c will not leak the client's identity to any entity in the network, since $misb$ is just a quantitative level indicating the severity of the misbehavior and is not specific to a particular client. The incorporation of the $misb$ field has several merits. One possible merit would be to punish clients with misbehavior history by higher network access latency. The gateway may intend to service well-behaved clients immediately upon receiving the ticket, and report ticket records to the TA at a later time. If the client appears to have misbehaved previously, and thus, may cast a threat on network operations, the gateway will first report the ticket record to the TA and will service the client only if the TA returns positive feedback (i.e., the TA performs ticket fraud detection to check if this ticket has been deposited before). Since we assume an offline TA in our scheme, the network access delay cannot be bounded and depends on the work load of the TA. Moreover, the TA may decrease the value of the issued tickets or reduce the frequency of approving the client's ticket requests based on the misbehavior level indicated in $misb$.

4.1.2 Ticket Deposit

After obtaining a valid ticket, the client may deposit it anytime the network service is desired before the ticket expires, using the ticket deposit protocol shown below. Our scheme restricts the ticket to be deposited only once at the first encountered gateway that provides network access services to the client according to val before exp.

1.
$$CL \rightarrow\rightarrow GW : PS_{CL}, m', W, c,$$
$$\sigma = (U', V', X', \rho, \sigma_1', \sigma_2'), t_5,$$
$$\mathcal{SIG}_{\widetilde{\Gamma_{CL}}}(m' \parallel W \parallel c \parallel \sigma \parallel t_5);$$

2. $GW \rightarrow\rightarrow CL$: ID_{GW}, $d = H_3(R \parallel W \parallel ID_{GW} \parallel T)$, t_6, $\mathcal{HMAC}_{\kappa'}(d \parallel t_6)$;
3. $CL \rightarrow\rightarrow GW$: PS_{CL}, $r_1 = d(u_1\alpha) + v_1$, $r_2 = d\alpha + v_2$, t_7, $\mathcal{HMAC}_{\kappa'}(r_1 \parallel r_2 \parallel t_7)$; and
4.
$$GW \rightarrow\rightarrow CL : ID_{GW}, misb, exp, t_8,$$
$$\mathcal{SIG}_{\Gamma_{GW}}(PS_{CL} \parallel ID_{GW} \parallel misb \parallel exp \parallel t_8);$$

At the end, the gateway checks if the equality $g_1^{r_1} g_2^{r_2} = R^d W$ holds. At the end of Step 1, the gateway will perform $\mathcal{VER}(\sigma)$ before Steps 2 and 3 can be proceeded, and R can be derived as $R = e(m', H_1(ID_{TA}))$ from the received information. T is the date/time when the ticket is deposited. A symmetric key κ' can be derived locally by the gateway and the client as $\kappa' = e(\Gamma_{GW}, PS_{CL})$, and $\kappa' = e(H_1(ID_{GW}), \widetilde{\Gamma_{CL}})$, respectively, after learning each other's ID (or pseudonym). The generation of the pseudonym will be discussed in Section 4.2.

The ticket is deemed valid if both the signature verification and the above equality check succeed. The deposit gateway (DGW), where the ticket is initially deposited, will then generate a signature on the client's pseudonym, the DGW's ID, and the associated $misb$ and exp values extracted from c. The signature is required to be present in order for other access points in the trust domain to determine whether and where to forward the client's access requests, if the deposited ticket will be further used from other access points. This is the reason why the client is not allowed to change his pseudonym while still using a deposited ticket to which the pseudonym is associated, since the DGW will refuse to offer access services to the client if the present pseudonym mismatches the one recorded with the ticket. As a result, the ticket value need to be set to a relatively small quantity in order to allow frequent update of the pseudonym if the client has high requirement on his anonymity [7], [17]. It will not place extra signaling overhead into the system since the TA can grant a batch of small-valued tickets during one single ticket issuance protocol. Due to the limited ticket value, the client is expected to have minimal mobility during the usage of the deposited ticket. However, there are also cases where the client moves to other gateways after the ticket is deposited. To address this issue, possible decision making functionalities may be incorporated into gateways. For instance, if the client temporarily moves to a new gateway in the DGW's vicinity, the new gateway can merely forward all the traffic of this client to the DGW, which then services the client based on the deposited ticket. If the client permanently moves to a new gateway, the new gateway may request the DGW to transfer the ticket record so that the new gateway can directly service the client. We do not intend to further address this issue. Instead, a simple and efficient solution can be employed to abandon the usage of the remaining ticket and deposits a new one at the new gateway since the ticket value is generally not large. This solution is also effective when the ongoing service is disrupted due to channel impairments, route failures, or mobility, as well as when the client tries to avoid mistaken multiple deposit. Adopting this solution, Step 4, in the above procedure can be omitted. On the other hand, if anonymity is not strictly required by the client, he can request tickets with higher values that can be used for longer time under a same pseudonym.

The DGW then creates a record for the deposited ticket as: $record = (ticket, r_1, r_2, T, rem, log)$, where rem and log denote the remaining value of the ticket and the logged data of the client's noncompliant behavior, respectively. The

value of rem is initially set to val. When the client uses the ticket to gain network access, the DGW initiates a traffic counter and decreases it based on the amount of traffic the client has injected and received. The remaining ticket value rem defines the amount of network access service the client will be offered before the ticket is depleted. The DGW essentially leverages rem to make sure that the client's traffic or access activity does not exceed the allowed amount defined in val. We do not constrain the number of tickets the client can request or the request frequency in the proposed scheme, rendering the opportunity for clients to inject a large amount of traffic or even to launch Denial of Service (DoS) attack, by gaining a considerable number of tickets in hand. Therefore, the log field is created to record such noncompliant behavior so that the DGW will be able to apply certain constraints on the client's bandwidth allocation based on the logged data in log. Note that the noncompliant behavior is different from misbehavior that solely refers to ticket reuse and multiple deposit. The ticket record will be deleted from the DGW's database once the ticket has expired (by checking c) and the most recent record (excluding rem) has been reported to the TA. Note that the DGW will maintain the record for the depleted tickets that have not expired in order to prevent the client from redepositing such tickets at this DGW. For clients with satisfactory $misb$ values, the ticket record is sent to the TA periodically, while it is sent to the TA before any network access service can be offered for clients with inferior $misb$ values, as mentioned before. These values are obtained and updated by the fraud detection protocol discussed next.

Note that the real ID of a client can be learned by the home TA at the time of ticket issuance due to the requirement for client authorization. However, this ID can be hidden from the access point unless this access point colludes with the home TA. The client simply deposits a ticket (using the ticket deposit protocol) for obtaining new tickets in which the ticket request is sent to the home TA in ciphertext. This activity can be exposed when the above collusion is present, which should not be a concern because it does not result in future exposure of activities performed under the new tickets. Furthermore, since a batch of tickets can be issued each time and the client may hold unused tickets, the deposit time of a specific ticket cannot be deduced by the timing analysis attack (i.e., estimating the timing relationship between ticket issuance and deposit).

4.1.3 Fraud Detection

Fraud is used interchangeably with misbehavior in this paper, which is essentially an insider attack. Ticket reuse generally results from the client's inability to obtain tickets from the TA when network access is desired, primarily due to the client's past misbehavior, which causes the TA to constrain his ticket requests. Multiple -deposit can also be termed client coalition, which is beneficial when the coalescing parties are unauthorized users or clients with misbehavior history having difficulty in acquiring tickets from the TA. Note, however, that since a client is able to obtain multiple tickets in one ticket issuance protocol and self-generate multiple pseudonyms (cf., Section 4.2), he can distribute these pseudonym/ticket pairs to other clients without being traced as long as each ticket is deposited only once. A possible remedy to this situation is to specify the nonoverlapping active period of a ticket instead of merely the expiry date/time such that each time, only one ticket can be valid. This approach, in general, requires synchronization. Another solution is to adopt the tamper-proof secure module so that a client cannot disclose his secrets to other parties since the secure module is assumed to be expensive and impractical to access or manipulate. This approach will eliminate the multiple deposit fraud but requires the deployment of secure modules. In the following discussion, we will still consider multiple deposit as a possible type of fraud (e.g., in case that secure modules are unavailable).

These two types of fraud share a common feature, that is, a same ticket (depleted or valid) is deposited more than once such that our one-time deposit rule is violated. This is where the restrictiveness of the blind signature algorithm takes effect on revealing the real identity of the misbehaving client. Specifically, when the TA detects duplicate deposits using the ticket records reported by gateways, the TA will have the view of at least two different challenges from gateways and two corresponding sets of responses from the same client. By solving the equation sets below based on these challenges and responses, the TA is able to obtain the identity information encoded in the message, and hence, the real identity of the misbehaving client. The fraud detection protocol is shown as

$$\begin{aligned} &GW \rightarrow TA : ID_{GW}, m', W, c, \\ &\quad \sigma = (U', V', X', \rho, \sigma_1', \sigma_2'), r_1, r_2, T, t_9, \\ &\quad \mathcal{HMAC}_{\kappa''}(m' \parallel W \parallel c \parallel \sigma \parallel r_1 \parallel r_2 \parallel T \parallel t_9), \end{aligned}$$

where κ'' is the preshared symmetric key between the gateway and the TA, which we have assumed for the WMN backbone. At the end, the TA performs $\mathcal{VER}(\sigma)$. If the signature can be successfully verified, the TA checks if m' (or the ticket serial number T_N) has been stored. If m' is not stored, the TA will store the following information: m', c, T, r_1, r_2 for future fraud detection. If m' has been stored, the TA will first compute the challenge $d = H_3(R \parallel W \parallel ID_{GW} \parallel T)$ and will accuse the gateway if d is the same as the stored one. If d is different, the TA can conclude that misbehavior has occurred and will reveal the identity information by constructing the following two sets of equations from two different views of the ticket records received from gateways:

$$r_1 = d(u_1\alpha) + v_1, \quad r_2 = d\alpha + v_2, \tag{1}$$

$$r_1' = d'(u_1\alpha) + v_1, \quad r_2' = d'\alpha + v_2. \tag{2}$$

The TA can solve for $u_1 = \frac{r_1 - r_1'}{r_2 - r_2'}$ and obtain the account number $\Omega = u_1 P_1$ to reveal the associated identity ID_{CL}. At this point, it is clear that the client-chosen secret $u_1 \in_R Z_p^*$ in ticket issuance serves as the embedded clue for tracing misbehaving clients.

By far, we have presented the techniques to resolve the conflicts between anonymity and traceability. As long as the client is a well-behaved user in this network, his anonymity can be fully guaranteed. This is achieved by the blinding process of the ticket issuance protocol, which breaks the linkage between the ticket and the identity, i.e., the TA

knows the client's real ID but does not know which ticket/pseudonym pairs belong to this client, while the gateway knows the linkage between the ticket and the pseudonym but learns no information on the real identity of the owner of these pairs. On the other hand, if the client misbehaves (i.e., fraud occurs), the client's anonymity can no longer be guaranteed since the TA may tend to identify this client, and subsequently, punish him possibly by revoking the client's network access privilege, leveraging the traceability property offered by our security architecture. In addition, our system enables authentication at the access points and meets the access control security requirement that is not satisfied in [16], where no authentication of the client is performed at the access point in the controlled connection protocol.

4.1.4 Ticket Revocation

Ticket revocation is necessary when a client is compromised, and thus, all his secrets are disclosed to the adversary. In our system, the adversary is motivated by gaining network services using tickets once the ticket-associated secrets are obtained from the compromised clients. Therefore, the compromised client needs to be able to revoke the ticket and prevent the adversary from acquiring benefits. The compromised client and the adversary are the only two parties that are in possession of the ticket-related secrets, a valid revocation request must be sent by the compromised client for genuine revocation purpose since the adversary gains nothing in doing so. The ticket revocation protocol consists of two cases as follows:

1. Revocation of new tickets: the client may store a number of unused tickets, as mentioned previously. When revoking these tickets that have not been deposited, the client sends PS_{CL}, T_N, t_{10}, $\mathcal{SIG}_{\widetilde{\Gamma_{CL}}}(T_N \parallel t_{10})$ in the revocation request to any encountered gateway. This gateway authenticates the client using PS_{CL} and records the ticket serial number T_N as revoked.
2. Revocation of deposited tickets: the client simply sends PS_{CL}, ID_{DGW}, t_{11}, $\mathcal{SIG}_{\widetilde{\Gamma_{CL}}}(ID_{DGW} \parallel t_{11})$ in the revocation request to the DGW. The DGW authenticates the client and marks the associated ticket revoked.

When gateways have records in the revocation database, they immediately report the revocations to the home TA, which will update and distribute the revocation list for all gateways in the trust domain to reference.

4.1.5 Accessing the Network from Foreign Domains

The access services the visiting (foreign) trust domain provided the ticket-based security architecture can take place in two ways including the following:

- A foreign mesh router $\overline{MR}$ (or foreign access point) forwards the client's new ticket request to the home domain when there is no available ticket for accessing the network from the foreign domain.
  - $$CL \rightarrow\rightarrow \overline{MR} : PST_{CL}, aP_0, t_{12}, \mathcal{HIDS}_{\psi_{CL}, s_{CL}}(H_1'(PST_{CL} \parallel aP_0 \parallel t_{12}));$$
  - $$\overline{MR} \rightarrow CL : IDT_{\overline{MR}}, bP_0, t_{13}, \mathcal{HIDS}_{\psi_{\overline{MR}}, s_{\overline{MR}}}(H_1'(IDT_{\overline{MR}} \parallel bP_0 \parallel t_{13}));$$ and
  - $CL \rightarrow\rightarrow \overline{MR}$: PST_{CL}, PS_{CL}, $\mathcal{SKE}_{\kappa}(ID_{CL} \parallel m)$, t_{14}, $\mathcal{HMAC}_{\bar{\kappa}}(PS_{CL} \parallel \mathcal{SKE} \parallel t_{14})$.
- $\overline{MR}$ (or an access point) forwards the client's ticket deposit request to the home domain when the client owns available new tickets issued by the home TA. The first two steps of the procedure are exactly the same as the above. The last step in this case will be $CL \rightarrow\rightarrow \overline{MR}$: PST_{CL}, PS_{CL}, $ticket$, t_{15}, $\mathcal{HMAC}_{\bar{\kappa}}(PS_{CL} \parallel ticket \parallel t_{15})$.

At the end, $\overline{MR}$ will forward the network access request consisting of $(PS_{CL}, \mathcal{SKE}_{\kappa}(ID_{CL} \parallel m))$ with κ the symmetric key between the client and his home TA, or $(PS_{CL}, ticket)$, to an access point (a gateway or mesh router) in the client's home domain if $\mathcal{HVER}(\mathcal{HIDS} \parallel QT)$ outputs "accept" in Steps 1 and 2. The symmetric key between the client and $\overline{MR}$ is $\bar{\kappa} = abP_0$, where $a, b \in_R Z_p^*$ and P_0 are the public domain parameter of the root PKG (cf., Section 3.3).

It is noted that the above triangular traffic forwarding via the home domain can be cumbersome if the client will stay at a foreign domain for a long term (e.g., not temporarily visiting). It is recommended that the client registers with the foreign TA to become an affiliated user of the foreign domain. Consequently, all the network access-related operations including ticket issuance, deposit, revocation, and fraud detection will follow the same procedures as in the home domain case, which greatly reduces the communication overhead in the system.

4.2 Pseudonym Generation and Revocation

The use of pseudonyms has been shown in the ticket-based protocols. This section copes with the pseudonym generation technique and the related revocation issue. The pseudonym is used to replace the real ID in the authentication, which is necessary for both anonymous network access and location privacy. In the intradomain authentication in our system, the client generates his own pseudonym by selecting a secret number $\varpi \in_R Z_p^*$ and computing the pseudonym $PS_{CL} = \varpi H_1(ID_{CL})$. The corresponding private key can be derived as $\widetilde{\Gamma_{CL}} = \varpi \Gamma_{CL} = \varpi \pi H_1(ID_{CL}) = \pi \cdot PS_{CL}$, in a similar way to that of [38]. Compared to [7], [9], [15] where a batch of pseudonyms are assigned to each client by the TA, the self-generation method vastly reduces the communication overhead in the system. Moreover, the client is able to frequently update his pseudonyms (with tickets) to enhance anonymity by using this inexpensive method.

When accessing the network from a foreign domain, suppose a client CL_j residing at level j is requesting network access from a foreign mesh router $\overline{MR}$ in a visiting

trust domain. After obtaining the private key ψ_j associated with the ID tuple $IDT_j = (ID_1, \ldots, ID_j)$ as $\psi_j = \psi_{j-1} + s_{j-1}H_1(IDT_j)$ from the parent (i.e., the home TA), the client CL_j derives the self-generated pseudonym tuples $\{PST_i : 1 \leq i \leq j\}$ as follows: CL_j selects a random secret $\varpi \in Z_p^*$ and computes the pseudonym tuples $PST_i = \varpi K_i = \varpi H_1(IDT_i)$ $(1 \leq i \leq j)$. The associated private key can be computed as $\widetilde{\psi}_j = \varpi\psi_j = \varpi \sum_{i=1}^{j} s_{i-1}K_i = \sum_{i=1}^{j} s_{i-1}\varpi K_i = \sum_{i=1}^{j} s_{i-1}\varpi H_1(IDT_i) = \sum_{i=1}^{j} s_{i-1} \cdot PST_i$. By substituting $PST_j/\widetilde{\psi}_j$ for $H_1(IDT_j)/\psi_j$ in the HIDS scheme [33], the signing and verification can be correctly performed.

As a final note on the self-generation algorithm, it would render the pseudonym revocation impossible by using the pseudonym alone. The reason is that any adversary who has compromised a client can generate valid pseudonym/key pairs that are only known to the adversary by running the self-generation algorithm. However, this pseudonym self-generation technique is appropriate in our system because the pseudonym revocation can be realized via revoking the associated ticket since the pseudonym is active only when its associated ticket is actively in use (deposited and not depleted). Therefore, the revocation process described in Section 4.1.4 for ticket revocation automatically revokes ticket-bound pseudonyms. If we employ the pseudonym assignment as in [7], [9], [15], in addition to the ticket-related operations, the TA will be required to generate and update the pool of pseudonyms for the client and to distribute the revocation list for revoking all effective pseudonyms in the active pool during a specific period, which induces significantly higher signaling overhead. The TA will also be able to derive the real identity corresponding to the assigned pseudonyms, which destroys the anonymity for honest clients.

5 Security Analysis

In this section, we analyze the security requirements our system can achieve as follows: Again, we use theorems in [32] for demonstration and the analysis using theorems in [31] can be carried out in a similar fashion.

Fundamental security objectives. It is trivial to show that our security architecture satisfies the security requirements for authentication, data integrity, and confidentiality, which follows directly from the employment of the standard cryptographic primitives, namely digital signature, message authentication code, and encryption, in our system. We are only left with the proof of nonrepudiation in this category. A fraud can be repudiated only if the client can provide a different representation (u_1, u_2) he knows of m from what is derived by the TA. If the client has misbehaved, the representation he knows will be the same as the one derived by the TA which ensures nonrepudiation.

Anonymity. First of all, it can be easily shown that a gateway cannot link a client's network access activities to his real identity. Due to the use of pseudonyms in authentication which reveals no information on the real ID, the gateway learns nothing about the identity of the client requesting network access. Since the pseudonym is generated by the client using his secret number, solving for the real identity from the pseudonym is equivalent to solving the DLP. Furthermore, the client's deposit gateway (DGW) cannot deduce the client's ID from the deposited ticket, which has been blinded by the client and does not reveal any identification information unless misbehavior occurs.

Next, we will show that the client's home TA cannot perform such linking either, which follows directly from Theorem 3 of [32] that the restrictive partially blind signature scheme used as a building block for our security architecture is partially blind. Specifically, any view of the ticket issuance protocol $(U, V, X, Y, Z, B, \sigma_1, \sigma_2, m)$ is unlinkable to any valid signature $(U', V', X', \rho, \sigma_1', \sigma_2', m')$ because it is proved in [32] that the blinding factors $(\alpha, \gamma, \tau, \lambda, \mu, \beta)$ always exist, which maps $(U, V, X, Y, Z, B, \sigma_1, \sigma_2, m)$ to $(U', V', X', \rho, \sigma_1', \sigma_2', m')$. Therefore, even an infinitely powerful $\mathcal{S}^*$ wins the game with probability $\frac{1}{2}$ (see [32]), which is equivalent to random guessing.

Finally, we show that even the collusion of the home TA and the DGW cannot carry out the linking. It is obvious that by collusion, the TA can learn no more from the gateway than the client's pseudonym used in association with a deposited ticket. The hardness of deducing a real identity from a pseudonym has been mentioned above. On the other hand, the gateway can learn the following from the TA: the private key, the account number, and the view of the ticket issuance protocol $(U, V, X, Y, Z, B, \sigma_1, \sigma_2, m)$ of a randomly chosen target ID_{CL}. Thus, the maximal amount of information the TA and gateway can exploit by collusion is 1) from the gateway: a ticket $m', W, c, (U', V', X', \rho, \sigma_1', \sigma_2')$ deposited by some client with an unknown-yet-authentic pseudonym for network access; 2) from the TA: a randomly chosen target identity, its associated private key, account number, and the view $(U, V, X, Y, Z, B, \sigma_1, \sigma_2, m)$ of the ticket issuance protocol. It is straightforward that because of the partial blindness of the adopted signature scheme and the hardness of solving DLP as shown above, the information pieces in points 1 and 2 are unlinkable other than random guessing.

Traceability (conditional anonymity). According to its definition, this requirement is twofold: 1) Anonymity for honest clients is unconditional, which can be proved following [27, Propositions 10 and 13]; 2) A misbehaving client is traceable where the identity can be revealed. The proof of point 2 follows from [32, Theorem 2] that the adopted restrictive partially blind signature scheme in our security architecture achieves restrictiveness. In other words, point 2 states that the client can only obtain signatures on messages of which the client knows a representation for which the structure in the representation (where the identity information is encoded) remains, proved by using [27, Proposition 12] and two extra requirements on the representations the client knows of m and m' (see [27] for detailed description of the two requirements).

Framing resistance. If the client is honest, with overwhelming probability, the representation (u_1, u_2) he knows is different from that the malicious TA falsely generated. Since the client could not have come up with this representation by himself, it proves that the TA attempts to frame the client. Therefore, innocent clients can exculpate themselves to prevent malicious TAs from revoking their network access privilege.

Unforgeability. The proof of unforgeability (formally defined in [32]) is essentially the proof of [32, Theorem 4]

that the adopted restrictive partially blind signature scheme is existentially unforgeable against adaptively chosen message and ID attacks under the assumption of the intractability of CDHP in G_1 and the random oracle.

We conclude that *the proposed security architecture satisfies the security requirements for anonymity, traceability, framing resistance, and unforgeability, in addition to the fundamental objectives including authentication, data integrity, confidentiality, and nonrepudiation, under the assumption that CDHP in G_1 is hard and the random oracle.*

6 Efficiency Analysis

Most pairing-based cryptosystems need to work in 1) a subgroup of the elliptic curve $E(F_q)$ of sufficiently large prime order p, and 2) a sufficiently large finite field F_{q^k}, where q is the size of the field over which the curve is defined and k is the embedding degree. For current minimum levels of security, we require $p > 2^{160}$ and $q^k > 2^{1,024}$ [39] to ensure the hardness of the DLP in G_1 and G_2. To improve the computation and communication efficiency when working with $E(F_q)$, we tend to keep q small while maintaining the security with larger values of k. According to [39], a popular choice is to work with points in $E(F_q)$, where $q \approx 2^{170}$, and to have a curve with embedding degree $k = 6$ so that $q^k \approx 2^{1,024}$. In the following analysis, we will use the parameter values given above, resulting in the elements in G_1 and G_2 to be roughly 171-bit (using point compression) and 1,024-bit, respectively. We further assume that SHA-1 [40] is used to compute the keyed-hash message authentication code (HMAC), which yields a 160-bit output.

We have mentioned that the interdomain access scenario described in Section 4.1.5 is expected to occur infrequently. The corresponding overhead is thus not a major concern and is discussed here briefly. Recall that the interdomain access is enabled by the hierarchical ID-based cryptosystem, the implementation of which largely determines the efficiency of the interdomain access. The communication and computation efficiency is best achieved using the Dual-HIDS introduced in Section 3.3. The client transmits approximately 148 bytes ($5 \times |G_1|element + 160bit\ HMAC\ output$) and 446 bytes ($5 \times |G_1|element + 2 \times |G_2|element + 160bit\ HMAC\ output$), respectively, for a new ticket request and a ticket deposit request. They correspond to the transmission time of 1.18 and 3.57 ms, respectively, assuming a 1 Mbps communication link between the client and the gateway. In the new ticket request, the client needs to perform an HIDS signing and verification, a symmetric-key encryption, and an HMAC, among which the HIDS operations dominate the computation costs. The signing involves only four point multiplications (three for HIDS and one for deriving the symmetric key), one point addition, and one hash evaluation, and can be efficiently carried out. The verification, however, requires four pairing operations, which are more expensive than signing but still acceptable given the advances in elliptic curve arithmetics (to be discussed in Section 6.3). In the ticket deposit request, the client is required to perform an HMAC and no more complex computations. The dominating storage overhead results from the hierarchical system-related parameters, specifically, the ID tuple and associated private key, and the derived pseudonym tuple and private key. The client stores an ID tuple/private key pair of $(3j + |G_1|element)$ bytes (assuming that each ID is represented by 3 bytes, which yields a system capacity of 16 million users including clients, gateways, and mesh routers), where j is the level the client resides in the hierarchical tree. The client further stores pregenerated pseudonym tuple/private key pairs for future use. Each such pair takes a storage space of $2 \times |G_1|element \approx 43$ bytes. The storage overhead experienced in the interdomain scenario is thus quite trivial compared to that in the intradomain scenario that is demonstrated next.

In the rest of this section, we will elaborate on the overhead incurred in the intradomain-based protocols, namely ticket issuance, ticket deposit, fraud detection, and ticket revocation.

6.1 Communication

Our ticket-based security architecture consists of four intradomain protocols in which ticket deposit involves only clients and gateways. This protocol is distributed in nature, and thus, the communication cost incurred is more affordable. In contrast, protocols involving interactions with the centralized TA contribute largely to the expensive communication costs in the system. In the fraud detection protocol, gateways report accumulated ticket records to the TA periodically instead of in real time. Reports from gateways can be scheduled at different time intervals, avoiding a sudden increase in the communication overhead caused by simultaneous transmissions. For each record, a gateway transmits roughly 443 bytes, including five G_1 elements, two G_2 elements, and four 160-bit elements. Other parameters in the record transmission are negligible compared to the above elements. Assuming that the communication channel between gateways and the TA has a bandwidth of 10 Mbps, each record takes 0.35 ms to be transmitted.

Ticket issuance and revocation may take place in real time. The associated communication overhead depends on how frequent 1) the clients use up issued tickets and 2) the clients misbehave. One can expect minimal real-time interaction with the TA for systems where ticket issuance is based on the client's usage trend (such that ticket requests other than scheduled will be infrequent) and there is a well-behaving majority. Since multiple tickets are issued to the client at each scheduled interval, the average communication cost can be further reduced because some parameters need only be transmitted once. In a single ticket issuance, the client sends roughly 60 bytes (i.e., three 160-bit elements) to the TA. The TA sends to the client approximately 128 bytes (i.e., four G_1 elements and two 160-bit HMACs). Note that we have excluded the information that need only be transmitted once. In the ticket revocation protocol, the gateway sends revocations in real time to the TA to ensure a timely update and distribution of the ticket revocation list. However, a small amount of communication overhead is incurred in this real-time interaction, which involves the gateway transmitting the 171-bit ticket serial number T_N and a 160-bit HMAC for each revoked ticket (not shown in the paper).

Note that in our protocols, symmetric keys are derived locally, and hence, no extra communication overhead due to the key agreement is introduced to the system. In addition, the role of TA may be split into several servers in reality. The communication overhead in our ticket-based system is considered acceptable.

6.2 Storage

As mentioned above, the TA may consist of several servers to store necessary information from all clients during protocol executions. The storage capability of these high-end servers is usually not a concern, and therefore, we focus on the storage overhead encountered at the low-end client side. Note that there is a trade-off between storage and computation overhead. In our protocols, the client has to perform pairing computations frequently, which is impractical due to the high cost of pairings and limited power of clients. Fortunately, many pairing operations in the protocols can be computed once and stored for future use. Furthermore, some stored information remains unchanged for all instances of protocol execution (e.g., all tickets issued in the ticket issuance protocol). As a result, we need merely take into account the effective storage overhead (i.e., information that is changed and has to be stored at each protocol instance).

In ticket issuance, the client stores for each protocol instance 621 bytes precomputed information ($3 \times |G_1|element + 4 \times |G_2|element + |Z_p^*|element$) and 43 bytes ($2 \times |G_1|element$) after-protocol information for future use. Information such as the symmetric key κ, and other protocol parameters m, X, Y, Z, g, g_1, g_2, etc., will only be stored once by the client for all protocol instances. The ticket issuance protocol contributes to most of the storage overhead at the client side, since all ticket-related information necessary for the remaining protocols will be stored by the end of ticket issuance. The only requirement left for intradomain protocols is the storage of the 43-byte pseudonym/private key pair and the corresponding symmetric keys (128 bytes each) with gateways and mesh routers. The size of this storage depends on the frequency of the client's pseudonym update. The trade-off between storage and computation overhead arises again where the symmetric key storage can be eliminated if digital signatures replace HMAC for integrity check in our protocols.

The mesh routers need not store any information during the protocol executions except for the interdomain access shown in Section 4.1.5, where the derived secret key abP_0 (around 21 bytes) may be stored for each client. However, since the interdomain access occurs infrequently, such storage overhead is negligible. Each mesh router also needs to store a shared key with each of other mesh routers, gateways, and the TA in the same domain, which are of limited number. Similarly, each gateway needs to store a shared key with each other entity in the domain. In addition, the gateway must store a record for each deposited ticket which is roughly 423 bytes, for future verification and report. This record is temporary and will be deleted once it is reported to the TA. The gateways are distributed across the system, and thus, will not become the bottleneck in terms of storage.

A large portion of storage overhead is due to parameters with size of $|G_2|$ element, which in our case is roughly 128 bytes. Although techniques in [41] can be employed to compress the size by a factor of three, it adds computation complication to the client. Whether to sacrifice storage or computation in a real scenario depends on the design goals and is a parameter to be considered by system designers.

6.3 Computation

Following the claims from the storage analysis, we are mainly interested in the computation overhead experienced at the client side and will count solely the effective overhead (i.e., the overhead that is varying for each protocol instance or cannot be precomputed). The computation tasks for clients include pairing operations (basic pairing and finite field exponentiation), point multiplications and additions, hash operations, etc., among which pairing operations are undoubtedly the most time-consuming task. An example can be found in [42, Tables 4.3 and 5.2], where pairing operations count for all the high computation costs.

In ticket issuance, the client only computes two basic pairings in real time for each protocol instance. The remaining pairing operations can either be computed once or be precomputed and stored for all protocol instances. Several HMAC operations also need to be performed in real time, which is considered computationally efficient. In ticket deposit, one signing, one verification, and two HMAC operations are performed in real time by the client for each ticket deposited. All pairings involved in this protocol can be precomputed except one for the verification. A finite field exponentiation is needed for the signing. Similarly, in ticket revocation, a client has to compute one signature in real time for each revoked ticket, which requires no basic pairings but a finite field exponentiation. If the Tate pairing is used for the basic pairing operation, it is shown in [43] that the time taken for computing a Tate pairing is 20 ms, 23 ms, and 26 ms, in the underlying base field of F_p (where $|p| = 512$-bit), $F_{2^{271}}$, and $F_{3^{97}}$, respectively. The first two fields have similar levels of security to 1,024-bit RSA while the last field has effective 922-bit security. Recent progress [44] shows that the computation time of Tate pairing on elliptic curves in characteristics 2 and 3 has been significantly improved, rendering pairing-based cryptosystems more realistic in security applications. We conclude from the analysis that the real-time computation intensity in our protocol is totally acceptable even on the low-end mobile device.

7 Security Enhancements

In addressing privacy and anonymity on the Internet, Dingledine [45] argues that cryptography alone will not hide the existence of confidential communication relationships and implemented an anonymous communication overlay network, Tor [21], based on the anonymous routing protocol, i.e., the onion routing [20]. In addressing the privacy preserving issue in vehicular ad hoc networks (VANETs) where the vehicles enjoy various VANET applications, Raya and Hubaux [7] claim that all vehicle identifiers, in particular, the MAC and IP addresses, must change over time, in addition to the frequent update of the anonymous keys (pseudonyms). Analogously, the proposed ticket-based anonymity system relies on effective anonymous routing protocols to construct anonymous communication paths and guarantee unlinkability. Unlinkability is a requirement for preserving user privacy in addition to anonymity. It refers to the property that multiple packets cannot be linked to have originated from a same client. For instance, if the network ID (i.e., IP address, MAC address) of a client's device is fixed and exposed in packet forwarding, the packets sent by a same client can be linked, which will enable the attackers to profile the client through traffic analysis attacks. By incorporating anonymous routing protocols suitable for

WMNs [46], [47], [48], [49] into our system, the real network ID will be effectively concealed rendering it difficult to profile the client and to discover the confidential relationship of the communicating parties. Note that anonymous routing serves as an enhancement to user privacy. The anonymity guarantee of our architecture will not be undermined even if anonymous routing protocols are absent.

Another possible enhancement is to incorporate peer-to-peer cooperation. In the WMNs considered here, the uplink from the client to the mesh router may rely on multihop communications. Peer clients act as relaying nodes to forward each other's traffic to the mesh router, which forms a P2P network. The notorious problem common in P2P communication systems is the free-riding, where some peers take advantage of the system by providing little or no service to other peers or by leaving the system immediately after the service needs are satisfied. Peer cooperation is thus the fundamental requirement for P2P systems to operate properly. Since peers are assumed to be selfish, incentive mechanisms become essential to promote peer cooperation in terms of both cooperativeness and availability [14]. Typical incentive mechanisms for promoting cooperativeness include reputation-based [50], [51] and payment-based [52], [53] approaches. In the reputation-based systems, peers are punished or rewarded based on the observed behavior. However, low availability remains an unobservable behavior [14] in such systems, which hinders the feasibility of the reputation-based mechanism in improving peer availability. By contrast, the payment-based approach provides sufficient incentives for enhancing both cooperativeness and availability, and thus, is ideal to be employed in multihop uplink communications among peer clients in our WMN system.

8 Conclusion

In this paper, we propose SAT, a security architecture mainly consisting of the ticket-based protocols, which resolves the conflicting security requirements of unconditional anonymity for honest users and traceability of misbehaving users. By utilizing the tickets, self-generated pseudonyms, and the hierarchical identity-based cryptography, the proposed architecture is demonstrated to achieve desired security objectives and efficiency.

Acknowledgments

This work was supported in part by the US National Science Foundation under grants CNS-0721744, CNS-0716450, CNS-0626881, CNS-0916391, CNS-0844972, and CNS-0716302. The work of Yuguang Fang was also supported in part by the National Science Council (NSC), R.O.C., under the NSC Visiting Professorship with contract number NSC-96-2811-E-002-010. The authors would like to thank Kenneth G. Paterson for helpful discussions and suggestions.

References

[1] European Telecomm. Standards Inst. (ETSI), "GSM 2.09: Security Aspects," June 1993.
[2] P. Kyasanur and N.H. Vaidya, "Selfish MAC Layer Misbehavior in Wireless Networks," *IEEE Trans. Mobile Computing*, vol. 4, no. 5, pp. 502-516, Sept. 2005.
[3] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," *Comm. ACM*, vol. 47, no. 6, pp. 53-57, 2004.
[4] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," *ACM Trans. Sensor Networks*, vol. 2, no. 4, pp. 500-528, Nov. 2006.
[5] W. Lou and Y. Fang, *A Survey on Wireless Security in Mobile Ad Hoc Networks: Challenges and Possible Solutions*, X. Chen, X. Huang, and D.-Z. Du, eds., Kluwer Academic Publishers/Springer, 2004.
[6] L. Zhou and Z.J. Haas, "Securing Ad Hoc Networks," *IEEE Network Magazine*, vol. 13, no. 6, pp. 24-30, Dec. 1999.
[7] M. Raya and J-P. Hubaux, "Securing Vehicular Ad Hoc Networks," *J. Computer Security*, special issue on security of ad hoc and sensor networks, vol. 15, no. 1, pp. 39-68, 2007.
[8] N.B. Salem and J-P. Hubaux, "Securing Wireless Mesh Networks," *IEEE Wireless Comm.*, vol. 13, no. 2, pp. 50-55, Apr. 2006.
[9] Y. Zhang and Y. Fang, "ARSA: An Attack-Resilient Security Architecture for Multihop Wireless Mesh Networks," *IEEE J. Selected Areas Comm.*, vol. 24, no. 10, pp. 1916-1928, Oct. 2006.
[10] I.F. Akyildiz, X. Wang, and W. Wang, "Wireless Mesh Networks: A Survey," *Computer Networks*, vol. 47, no. 4, pp. 445-487, Mar. 2005.
[11] S. Brands, "Untraceable Off-Line Cash in Wallets with Observers," *Proc. 13th Ann. Int'l Cryptology Conf. Advances in Cyptology (CRYPTO '93)*, pp. 302-318, Aug. 1993.
[12] K. Wei, Y.R. Chen, A.J. Smith, and B. Vo, "Whopay: A Scalable and Anonymous Payment System for Peer-to-Peer Environments," *Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS)*, July 2006.
[13] D. Chaum, A. Fiat, and M. Naor, "Untraceable Electronic Cash," *Proc. Conf. Advances in Cryptology (CRYPTO '88)*, 2002.
[14] D. Figueiredo, J. Shapiro, and D. Towsley, "Incentives to Promote Availability in Peer-to-Peer Anonymity Systems," *Proc. IEEE Int'l Conf. Network Protocols (ICNP)*, pp. 110-121, Nov. 2005.
[15] G. Ateniese, A. Herzberg, H. Krawczyk, and G. Tsudik, "Untraceable Mobility or How to Travel Incognito," *Computer Networks*, vol. 31, no. 8, pp. 871-884, Apr. 1999.
[16] Q. He, D. Wu, and P. Khosla, "Quest for Personal Control over Mobile Location Privacy," *IEEE Comm. Magazine*, vol. 42, no. 5, pp. 130-136, May 2004.
[17] A.R. Beresford and F. Stajano, "Location Privacy in Pervasive Computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46-55, Jan.-Mar. 2003.
[18] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "MASK: Anonymous On-Demand Routing in Mobile Ad Hoc Networks," *IEEE Trans. Wireless Comm.*, vol. 5, no. 9, pp. 2376-2385, Sept. 2006.
[19] S. Seys and B. Preneel, "ARM: Anonymous Routing Protocol for Mobile Ad Hoc Networks," *Proc. 20th Int'l Conf. Advanced Information Networking and Applications (AINA)*, pp. 133-137, Apr. 2006.
[20] M.G. Reed, P.F. Syverson, and D.M. Goldschlag, "Anonymous Connections and Onion Routing," *IEEE J. Selected Areas Comm.*, vol. 16, no. 4, pp. 482-494, May 1998.
[21] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," *Proc. USENIX Security Symp.*, pp. 303-320, Aug. 2004.
[22] J. Sun, C. Zhang, and Y. Fang, "A Security Architecture Achieving Anonymity and Traceability in Wireless Mesh Networks," *Proc. IEEE INFOCOM*, pp. 1687-1695, Apr. 2008.
[23] D. Chaum, "Blind Signatures for Untraceable Payments," *Advances in Cryptology—Crypto '82*, pp. 199-203, Springer-Verlag, 1982.
[24] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing Mobile Ad Hoc Networks with Certificateless Public Keys," *IEEE Trans. Dependable and Secure Computing*, vol. 3, no. 4, pp. 386-399, Oct. 2006.
[25] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairings," *Advances in Cryptology-Asiacrypt 2001*, pp. 514-532, Springer-Verlag, 2001.
[26] A. Juels, M. Luby, and R. Ostrovsky, "Security of Blind Digital Signatures," *Advances in Cryptology—Crypto '97*, pp. 150-164, Springer-Verlag, 1997.
[27] S. Brands, "An Efficient Offline Electronic Cash System Based on the Representation Problem," CWI Technical Report CS-R9323, 1993.
[28] M. Abe and T. Okamoto, "Provably Secure Partially Blind Signatures," *Advances in Cryptology—Crypto 2000*, pp. 271-286, Springer-Verlag, 2000.

[29] S.M. Chow, C.K. Hui, S.M. Yiu, and K.P. Chow, "Two Improved Partially Blind Signature Schemes from Bilinear Pairings," *Proc. Australasian Conf. Information Security and Privacy (ACISP '05)*, pp. 316-328, 2005.

[30] G. Maitland and C. Boyd, "A Provably Secure Restrictive Partially Blind Signature Scheme," *Lecture Notes in Computer Science*, pp. 99-114, Springer-Verlag, 2002.

[31] X. Chen, F. Zhang, Y. Mu, and W. Susilo, "Efficient Provably Secure Restrictive Partially Blind Signatures from Bilinear Pairings," *Proc. 10th Conf. Financial Cryptography and Data Security (FC '06)*, pp. 251-265, Feb. 2006.

[32] X. Chen, F. Zhang, and S. Liu, "ID-Based Restrictive Partially Blind Signatures and Applications," *J. Systems and Software*, vol. 80, no. 2, pp. 164-171, Feb. 2007.

[33] C. Gentry and A. Silverberg, "Hierarchical Id-Based Cryptography," *Proc. ASIACRYPT*, pp. 548-556, Dec. 2002.

[34] F. Hess, "Efficient Identity-Based Signature Schemes Based on Pairings," *Selected Areas in Cryptography (SAC 2002)*, pp. 310-324, Springer-Verlag, 2002.

[35] R. Dutta, R. Barua, and P. Sarkar, *Pairing-Based Cryptography: A Survey*, Cryptology ePrint Archive, Report 2004/064, http://eprint.iacr.org/2004/064.pdf, 2004.

[36] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems Based on Pairing," *Proc. Symp. Cryptography and Information Security (SCIS)*, Jan. 2000.

[37] A. Menezes, P.V. Oorschot, and S. Vanston, *Handbook of Applied Cryptography*. CRC Press, 1996.

[38] S.M.M. Rahman, A. Inomata, T. Okamoto, M. Mambo, and E. Okamoto, "Anonymous Secure Communication in Wireless Mobile Ad-Hoc Networks," *Proc. First Int'l Conf. Ubiquitous Convergence Technology*, pp. 131-140, Dec. 2006.

[39] S.D. Galbraith, "Pairings," *Advances in Elliptic Curve Cryptography*, I.F. Blake, G. Seroussi, and N.P. Smart, eds., pp. 183-213, chapter 9, Cambridge Univ. Press, 2005.

[40] NIST, *Digital Hash Standard*, Fed. Information Processing Standards (FIPS) Publication 180-1, Apr. 1995.

[41] R. Granger, D. Page, and M. Stam, "A Comparison of CEILIDH and XTR," *Algorithmic Number Theory: Sixth Int'l Symp., ANTS-VI*, pp. 235-249, Springer, 2004.

[42] H.W. Lim, "On the Application of Identity-Based Cryptography in Grid Security," PhD thesis, Univ. of London, 2006.

[43] P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott, "Efficient Algorithms for Pairing-Based Cryptosystems," *Advances in Cryptology—CRYPTO 2002*, pp. 354-368, Springer-Verlag, 2002.

[44] P.S.L.M. Barreto, S.D. Galbraith, C. ÓhÉigeartaigh, and M. Scott, "Efficient Pairing Computation on Supersingular Abelian Varieties," Cryptology ePrint Archive, Report 2004/375, http://eprint.iacr.org/2004/375.pdf, Sept. 2005.

[45] R. Dingledine, "Tor: An Anonymous Internet Communication System," *Proc. Workshop Vanishing Anonymity, the 15th Conf. Computers, Freedom, and Privacy*, Apr. 2005.

[46] M. Blaze, J. Ioannidis, A.D. Keromytis, T. Malkin, and A. Rubin, "Anonymity in Wireless Broadcast Networks," *Int'l J. Network Security*, vol. 8, no. 1, pp. 37-51, Jan. 2009.

[47] X. Wu and N. Li, "Achieving Privacy in Mesh Networks," *Proc. Fourth ACM Workshop Security of Ad Hoc and Sensor Networks (SASN '06)*, pp. 13-22, Oct. 2006.

[48] T. Wu, Y. Xue, and Y. Chi, "Preserving Traffic Privacy in Wireless Mesh Networks," *Proc. Int'l Symp. World of Wireless, Mobile and Multimedia Networks (WoWMoM '06)*, 2006.

[49] Z. Wan, K. Ren, B. Zhu, B. Preneel, and M. Gu, "Anonymous User Communication for Privacy Protection in Wireless Metropolitan Mesh Networks," *Proc. ASIAN ACM Symp. Information, Computer and Comm. Security (ASIACCS '09)*, pp. 368-371, Mar. 2009.

[50] S. Buchegger and J.L. Boudec, "The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-Hoc Networks," *Proc. Workshop Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '03)*, Mar. 2003.

[51] Y. Zhang and Y. Fang, "A Fine-Grained Reputation System for Reliable Service Selection in Peer-to-Peer Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 8, pp. 1134-1145, Aug. 2007.

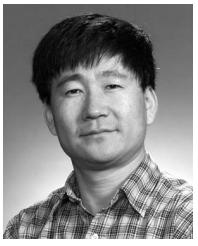
[52] S. Zhong, J. Chen, and Y. Yang, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM*, vol. 3, pp. 1987-1997, Apr. 2003.

[53] Y. Zhang, W. Lou, W. Liu, and Y. Fang, "A Secure Incentive Protocol for Mobile Ad Hoc Networks," *ACM Wireless Networks*, vol. 13, no. 5, pp. 569-582, Oct. 2007.

Jinyuan Sun received the BSc degree in computer information systems from Beijing Information Technology Institute, China, in 2003, the MASc degree in computer networks from Ryerson University, Canada, in 2005, and the PhD degree in electrical and computer engineering from the University of Florida, in 2010. She was a Network Test Developer at RuggedCom Inc., Ontario, Canada, 2005-2006. She has been an assistant professor in the Department of Electrical Engineering and Computer Science at University of Tennessee Knoxville since August 2010. Her research interests include the security protocol and architecture design of wireless networks. She is a member of the IEEE.

Chi Zhang received the BE and ME degrees in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in July 1999 and January 2002, respectively. Since September 2004, he has been working toward the PhD degree in the Department of Electrical and Computer Engineering at the University of Florida, Gainesville. His research interests include network and distributed system security, wireless networking, and mobile computing, with emphasis on mobile ad hoc networks, wireless sensor networks, wireless mesh networks, and heterogeneous wired/wireless networks. He is a student member of the IEEE.

Yanchao Zhang received the BE in Computer Science & Technology from Nanjing University of Posts & Telecom, China, in 1999, the ME in Computer Science & Technology from Beijing University of Posts & Telecommunications, China, in 2002, and the PhD in Electrical and Computer Engineering from the University of Florida in 2006. He joined Arizona State University (ASU) in June 2010 as an associate professor of the School of Electrical, Computer, and Energy Engineering. Before ASU, he was an assistant professor of electrical and computer engineering at New Jersey Institute of Technology from August 2006 to June 2010. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is an associate editor of *IEEE Transactions on Vehicular Technology* and a feature editor of *IEEE Wireless Communications*. He received the National Science Foundation Faculty Early Career Award in 2009. He is a member of the IEEE.



Yuguang Fang received the PhD degrees in systems engineering and electrical engineering from Case Western Reserve University in January 1994 and Boston University in May 1997, respectively. He was an assistant professor in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology from July 1998 to May 2000. He then joined the Department of Electrical and Computer Engineering at the University of Florida in May 2000 as an assistant professor. He received an early promotion to an associate professor with tenure in August 2003 and a professor in August 2005. He has published more than 300 papers in refereed professional journals and conferences. He received the US National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He has served on many editorial boards of technical journals including the *IEEE Transactions on Communications*, the *IEEE Transactions on Wireless Communications*, the *IEEE Transactions on Mobile Computing*, and the *ACM Wireless Networks*. He is currently the Editor-in-Chief for *IEEE Wireless Communications*. He is a fellow of the IEEE.