Wireless Netw (2016) 22:2523–2535
DOI 10.1007/s11276-015-1109-7

CAKA: a novel certificateless-based cross-domain authenticated key agreement protocol for wireless mesh networks

Yanping Li[1] · Weifeng Chen[2] · Zhiping Cai[3] · Yuguang Fang[4]

Published online: 7 November 2015

Abstract Due to the flexibility of wireless mesh networks (WMNs) to form the backhaul subnetworks, future generation networks may have to integrate various kinds of WMNs under possibly various administrative domains. Aiming at establishing secure access and communications among the communication entities in a multi-domain WMN environment, in this paper, we intend to address the cross-domain authentication and key agreement problem. We present a light-weight cross-domain authentication and key agreement protocol, namely **CAKA**, under certificateless-based public key cryptosystem. CAKA has a few attractive features. First, mutual authentication and key agreement between any pair of users from different WMN domains can be easily achieved with two-round interactions. Second, no central domain authentication server is required and fast authentication for various roaming scenarios is supported by using a repeated cross-domain algorithm. Third, no revocation and renewal of certificates and key escrow are needed. Finally, it provides relatively more security features without increasing too much overhead of computation and storage. Our analysis shows that the proposed CAKA protocol is highly efficient in terms of communication overhead and resilient to various kinds of attacks.

Keywords Certificateless public key cryptography · Wireless mesh networks · Cross-domain authentication · Key agreement

✉ Yuguang Fang
fang@ece.ufl.edu

Yanping Li
lyp@snnu.edu.cn

Weifeng Chen
chen@calu.edu

Zhiping Cai
zpcai@nudt.edu.cn

[1] School of Mathematics and Information Science, Shaanxi Normal University, Xi'an 710062, China

[2] Department of Math and Computer Science, California University of Pennsylvania, California, PA 15419, USA

[3] College of Computer, National University of Defense Technology, Changsha 410073, China

[4] Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA

1 Introduction

Wireless mesh networks (WMNs) have been recognized as an innovative and competitive technology for the next generation broadband mobile communication and high-speed Internet access due to the features of self-organization, self-healing, and low costs. Numerous commercial and experimental WMNs have been deployed all over the world, ranging from metro-scale broadband municipal networks to medium-scale and small-scale community networks [1]. The major components of a wireless mesh network include mesh clients (MCs), mesh routers (MRs), and mesh gateways (MGWs) [2]. Mesh clients could be desktop, database servers, smartphones or tablets whereas mesh routers are mesh backbone providing multi-hop connectivity from one mesh client to another or to a mesh gateway for Internet access.

A typical wireless mesh network consists of several cooperating sub-networks, referred to as *mesh domains* in WMNs. Two mesh domains have been shown in Fig. 1, each of them administered by an Internet Service Provider (ISP). Secure communication is always needed in WMNs.

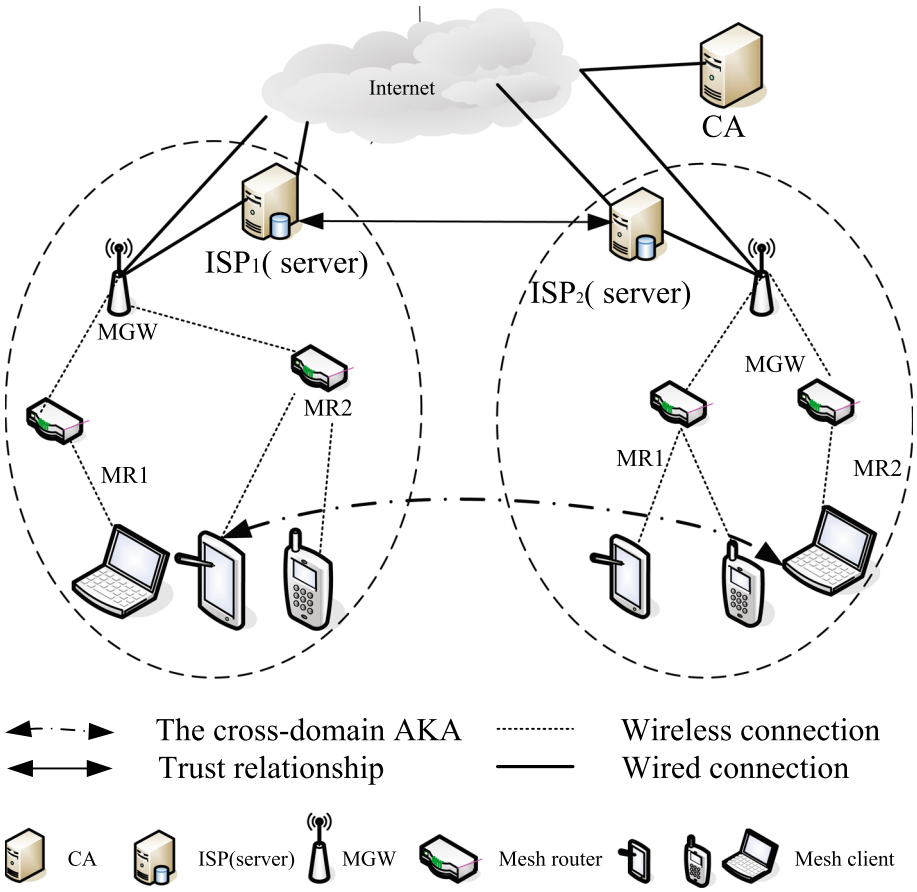


Fig. 1 Cross-domain AKA model and three-layer architecture

When secure communication is between components within the same domain, it is easy to achieve since they are administered by the same ISP. However, the problem becomes challenging when a mesh client travels to another domain, referred to as the *foreign domain*, and requires secure communication with an entity in the foreign domain. Both mutual authentication and key agreement need to be considered [1–9].

Authenticated key agreement (AKA) [10–12] incorporate authentication and key agreement in one logical step. Although two-party and three-party AKA protocols have been extensively studied and widely deployed [13], it is still not clear how they can be directly applied to establish a secure cross-domain communication channel in WMNs. A few certificateless-based AKA protocols have been proposed [14–16]. However, they do not address authentication and key agreement in the cross-domain situation. The reason is that mobile nodes (e.g., smartphones, tablets) have limited storage, computing capability and power supply. Clients' mobility demands efficient and fast handoff mechanisms, which makes the protocol design and implementation very difficult. Therefore, in this paper, we investigate how to design efficient protocols for authentication and key establishment in the cross-domain scenarios in WMNs, based on certificateless public key cryptography (CL-PKC).

1.1 Related work

A novel identity-based secure architecture to enable secure communications in large-scale multi-domain wireless mesh networks was proposed in [1]. Under this scheme, when a user U from domain A travels to a foreign domain B and requests a secure communication service, a total of eight message interactions are needed. Additionally, U needs to communicate with the trusted authority T_B in domain B, which further communicates with the trusted authority T_A. As a consequence, trusted authorities T_A and T_B will easily become the bottleneck of the system. The involvement of the trusted authorities in both domains also increases the authentication delay.

The ARSA protocol is based on universal passes [3] to achieve secure roaming across WMN domains administrated by different operators. However, the universal passes have expiration timers. When a pass is expired or lost, the user with the pass will not receive secure communication services.

In [4], a fast authentication scheme using tickets for a roaming user from one WMN domain to another is proposed. However, this scheme needs three types of tickets used in the authentication protocol. The scheme also requires a ticket agent that is trusted by all users to manage all kinds of tickets. Therefore, the ticket agent is vulnerable to attack. Similarly, in [5], a mesh client registered to a home location register (HLR) can prove its registration to a foreign location register (VLR) using a delegation scheme based on proxy signatures and two kinds of tickets: direct ticket and delegating ticket. In the process of delegation, however, the scheme requires complex encryption and signature verification.

The schemes in Wong and Lim [6] and Chen et al. [7] proposed a password-based authenticated inter-domain key exchange protocol based on identity-based cryptography. However, it is based on the assumption that two users have to share a password before the authentication. This assumption is impractical especially in WMNs because two users may belong to different domains. They may not agree on a password when a roaming user gets into an unfamiliar domain for the first time. It will be difficult to have a secret password in advance with a visiting mesh domain.

Authors in He and Agrawal [8] improved the scheme in Zhu et al. [1] to achieve distributed authentication between a mesh client and an access router. Unfortunately, the scheme requires a large number of message exchanges for negotiation, causing a long authentication delay that impedes real-time applications such as VoIP. Furthermore, the total authentication message overhead could be significant when the number of users grows large or when frequent roaming occurs.

The schemes in Ren and Lou [2] and Ren et al. [9] considered similar application scenarios as ours. They proposed distributed authentication and key agreement schemems based on traditional certificate-based public key crypto-systems. Both schemes require a large number of signatures, especially the complex group signature and the time-consuming inverse computation in ECDSA-160. In this paper, based on Certificateless-PKC, our CAKA protocol aims to establish an authentication session key such that two users (user-to-user, user-to-router, and router-to-router) can securely exchange information when moving from one WMN domain to another. We use a relatively simple architecture and a more realistic trust model without the key-escrow problem and the certificate-revocation problem.

1.2 Our contributions

In this paper, we describe a hierarchical security architecture and trust model for WMNs. As shown in Fig. 1, there is a central authority (CA) that will set up and manage the security parameters of the whole wireless mesh network (WMN). The WMN consists of multiple domains and one Internet Service Provider (ISP) manages all users (like MCs) and network devices (like MRs and MGWs) within its domain. All ISPs trust the CA and also trust each other. Users and devices within a domain trust the ISP managing the domain. Under this model, we propose a novel authenticated key agreement (AKA) protocol that efficiently achieves Cross-domain mutual Authentication and Key Agreement (CAKA). The CAKA protocol allows a user, when traveling to a foreign domain, to quickly accomplish mutual authentication and establish a session key with any user or device in the foreign domain.

In summary, we have made the following contributions:

- In most existing AKA protocols [1, 4, 8], when a user travels to a foreign domain and wants to achieve mutual authentication and establish a session key with another user in the foreign domain, both the home ISP and the foreign ISP need to participate. However, in our proposed CAKA protocol, none of the ISPs needs to participate in the mutual authentication and key agreement process. The roaming user can directly contact any user (a mesh client) or a device (a mesh router or a gateway) in the foreign domain to accomplish the mutual authentication and key agreement. Thus the CAKA protocol reduces the computation and communication workload from the ISPs, preventing them from becoming the system bottleneck for a large scale WMN.
- Our CAKA protocol is designed based on certificateless public key cryptography (CL-PK). Thus it does not have the key escrow problem of the identity-based (ID-based) AKA protocols. In an ID-based AKA protocol, a user's private key is generated by a trusted entity and securely delivered to the user. However, in CAKA, a user chooses its own private key by itself, not completely generated by the trusted entity.
- Compared to existing related protocols, the proposed CAKA protocol significantly reduces the number of messages exchanged to complete the mutual authentication and establish a session key. Only two messages need to be exchanged to complete this process. In the case when a user frequently crosses domains, the CAKA protocol further reduces the computation workload for encryption/decryption. Consequently, the CAKA protocol is attractive to WMNs, where mesh clients normally have limited storage, computing capability and power supply.

The rest of this paper is organized as follows. Section 2 introduces the basic knowledge and preliminaries. We describe the details of the proposed CAKA protocol in Sect. 3. Security analysis and the performance evaluation

of the CAKA protocol is given in Sect. 4. We provide concluding remarks and outline our future work in Sect. 5.

2 Preliminaries

In this section, we present some preliminaries used in the CAKA scheme.

2.1 Certificateless public key cryptography (CL-PKC)

CL-PKC was introduced by Al-Reyami and Peterson [17], which can be viewed as an intermediate model between the traditional certificate-based PKC and the identity-based PKC (ID-PKC). Different from the traditional public key cryptographic system, the CL-PKC scheme does not need certificates to authenticate public keys. It is also different from ID-PKC schemes. ID-PKC schemes completely rely on the trusted third party (TTP) and a user's private key is generated by the TTP. However, in a CL-PKC scheme, a user's private key is generated jointly by a trusted authority and the user. Nobody knows the private key except the user itself. Consequently, CL-PKC schemes are not subject to the key escrow problem, which seems to be inherent in all ID-PKC schemes and all PKC schemes. On the other hand, CL-PKC schemes also have the advantages of the ID-PKC schemes, i.e., a user's public key can be derived from its public identity information, such as names, email addresses, telephone numbers or any strings of characters. Due to these features, CL-PKC schemes have received intensive attention since it was invented. Details of CL-PKC schemes can be found in [17, 18].

2.2 Bilinear pairing

ID-PKC, including CL-PKC, has extensively utilized the bilinear pairing function. For convenience, we offer the brief definition here. Let q be a large prime, $\mathbb{G}_1$ be a q-order additive group, and $\mathbb{G}_2$ be a q-order multiplicative group. When $m \in Z_q$ and $P \in \mathbb{G}_1$, we write mP for P added to itself m times, also called scalar multiplication of P by an integer m. From a cryptographic point of view, a pairing is a map $e(\cdot,\cdot) : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties:

Bilinear: $\forall P, Q \in \mathbb{G}_1, \forall a \in Z_q, e(aP, Q) = e(P, aQ) = e(P, Q)^a$.
Nondegenerate: $e(P, Q) \neq 1_{\mathbb{G}_2}$, whenever $P, Q \neq 1_{\mathbb{G}_1}$.
Computable: for all $P, Q \in \mathbb{G}_1$, there exists an efficient algorithm to compute $e(P, Q) \in \mathbb{G}_2$.

In practice, bilinear pairing can be implemented using modified Weil and Tate pairing on elliptic curves [19], and its security is based on the Computational Diffie–Hellman Problem (CDHP:Given P, aP, bP with uniformly random choices of $a, b \in Z_q^*$, compute abP) and the Bilinear Diffie–Hellman Problem (BDHP:Given P, aP, bP, cP with uniformly random choices of $a, b, c \in Z_q^*$, compute $e(P, P)^{abc} \in \mathbb{G}_2$) [20].Currently, it is believed that CDHP and BDHP are computationally hard, leading to the CDHP assumption and BDHP assumption, respectively. For more details, please refer to [19].

We also introduce the discrete logarithm problem (DLP) that forms the basis of security for our proposed scheme. Assume that the DLP is a hard problem in both $\mathbb{G}_1$ and $\mathbb{G}_2$. It is computationally infeasible to obtain integers a, r from given $U \in \mathbb{G}_1, V \in \mathbb{G}_2$ such that $U = aP$ and $V = e(P, Q)^r$.

2.3 Desirable security attributes

Let A and B be two mesh entities (e.g., mesh clients or mesh routers) that want to execute an authenticated key agreement (AKA) protocol to accomplish mutual authentication and establish a session key for secure communication. In this subsection, we summarize the desirable properties of any AKA protocol. More details are described in [4, 12, 20].

Known-session key secrecy (KSKS) Each execution of the AKA protocol between A and B should produce a unique and independent secret session key. Compromising one session key should not affect the secrecy of session keys produced for other sessions.

Forward secrecy (FS) Disclosure of the private keys of A and B should not affect the previous session keys established between A and B before the disclosure. *Perfect Forward Secrecy* implies that an attacker, even armed with both A and B's private keys, cannot determine previously used session keys between A and B. *Partial Forward Secrecy* implies that an attacker, armed with either private keys, but not both, cannot determine previously used session keys. Obviously, the former implies the latter.

Key-compromise impersonation (KCI) resilience If an adversary compromise user A's private key, the adversary can impersonate A, but cannot impersonate any other users, i.e., any other users should not be affected by the compromise of A's private key.

Unknown key-sharing (UKS) resilience User A cannot be coerced to share a key with any other user C while A believes that he is sharing the key with entity B. In other words, it should not be possible for A to believe that he is sharing a key with C, while B correctly thinks the key is shared with A.

No key control(NKC) When A and B want to establish a session key, none of them should be able to forge the session key with a preselected value.

3 Cross-domain authentication and key agreement (CAKA)

We now describe our CAKA protocol, beginning with the hierarchical architecture and trust model.

3.1 Hierarchical architecture and trust model

Our description in this section follows the hierarchical architecture of a WMN, as shown in Fig. 1.

Figure 1 illustrates a three-layer hierarchical security architecture for a multi-domain WMN where one ISP being the trusted authority manages one domain. The top layer of the hierarchical architecture is a central authority (CA) that runs the root Public Key Generator to generate public common parameters for the whole WMN, according to the security requirements. As will be described, the CA does not participate in the process of any two cross-domain mutual authentication and session key agreement, nor in the detailed network operations.

The second layer consists of ISPs or domain managers. Each ISP generates its domain's secret key and public key. The ISP also registers the domain information and the domain public key to the CA. We assume each ISP can be reached by all the users (e.g., mesh clients, mesh routers or mesh gateways) in its domain through either direct or multi-hop communication. Each ISP generates partial private keys for the users in its domain, during the registration process. After that, ISPs do not participate in any AKA process. This is the significant difference from existing schemes.

The third layer is mainly composed of mesh users that might roam to another domain. Mesh users in home domain provide access service for roaming MCs. Every user (including MRs and MCs) has a home WMN domain where he is registered in with his domain manager ISP and user profile information is maintained for a relatively long period of time.

The trust relationships among entities are assumed as follows. ISPs have a long-term trust relationship with the CA. All ISPs are assumed to have a long-term trust relationship with each other. For example, they may want to cooperate as a federation to provide seamless Internet access services (e.g., global roaming) to mesh clients. All users within a domain have a long-term trust relationship with the ISP managing the domain. Based on this trust model, we want to design an efficient AKA protocol that allows a roaming user in a foreign domain to establish mutual authentication and establish a session key with a user in the foreign domain.

3.2 System initialization

Before the proposed CAKA protocol can be executed when a user travels to a foreign domain, a set of initialization operations need to be conducted, described in this subsection.

Setup This operation is executed by the CA to bootstrap the system. More specifically, the CA will generates $\mathbf{params} = (q, \mathbb{G}_1, \mathbb{G}_2, e(\cdot,\cdot), P, H_1, H_2)$, where P is the generator of $\mathbb{G}_1$.$\mathbb{G}_1$ and $\mathbb{G}_2$ are a cyclic additive group and a cyclic multiplicative group of prime order q,respectively. $e(\cdot,\cdot)$ is a bilinear pairing map, and H_1, H_2 are computationally secure Hash functions with strong resistance to collision: $H_1 : \{0,1\}^* \times \mathbb{G}_1 \rightarrow \mathbb{G}_1^*, H_2 : \mathbb{G}_2 \times \mathbb{G}_1 \rightarrow \{0,1\}^l$. Here, l is the session key length of a symmetric encryption algorithm. These public parameters can be published on a bulletin board and made publicly available.

Domain-master-public/secret-key The next operation will be conducted by each ISP to generate the private key and public key of its own domain. An ISP in domain D selects its domain master secret key $s_D \in_U Z_q^*$, and computes the domain public key as $p_D = s_D P$ where P comes from the system **params**. The ISP should well safeguard and prevent unauthorized access to the domain master secret key and register the domain information and the domain public key to the CA.

Set-public-key A mesh user U executes this operation to generate its public key. More specifically, user U randomly selects its secret value $\kappa_U \in_U Z_q^*$, and computes $X_U = \kappa_U P$ where P comes from the system **params**. The public key of U is $p_U = \langle I_U, X_U \rangle$ where I_U is U's real identity such as names or email addresses. Here we assume each mesh user (an MC or an MR) has a unique identity in its domain. The private key of user U will be generated jointly by U and its ISP through the following two steps.

Partial-private-key-extract The ISP executes this algorithm to generate U's partial private key. Specifically, the ISP computes $E_U = s_D Q_U$ where $Q_U = H_1(I_U || X_U) \in \mathbb{G}_1$. E_U is sent to U on a confidential and secure channel.

Set-Private-Key Upon receiving E_U , user U can verify the correctness of E_U by checking $e(E_U, P) \stackrel{?}{=} e(Q_U, p_D)$. If the verification succeeds, U will generate its private key as $s_U = \langle \kappa_U, E_U \rangle$.

The operations described above are analogous to the ones proposed in [14, 21], which were proved to be equivalent to the computational Diffie–Hellman problem under the random oracle model with a tight reduction.

To simplify the subsequent description, we summarize some notations in Table 1, where we define $Flag_U = \langle D, p_D, p_U, Q_U \rangle$ as the identifier flag of U, which will be used frequently in a cross-domain authentication scenario.

3.3 The cross-domain authenticated key agreement (CAKA) protocol

After the *SystemInitialization* described above, we now present the CAKA protocol that allows a user, when roaming to a foreign domain, to efficiently accomplish mutual authentication and establish a session key with another user in the foreign domain. To simplify our description, consider the scenario when user A from domain D_A roams to a foreign domain D_B and conducts the CAKA protocol with user B in D_B. User B can be any entity (a mesh client, a mesh router or a mesh gateway) that is close to A.

Figure 2 shows the three-step interaction between A and B. Only two messages are exchanged to accomplish mutual authentication and establish a session key k. More details of these three steps are described below.

Step 1 A randomly picks a, computes R_A, P_A, P'_A, K_A and k_1 as shown in Fig. 2, and sends the first message M_1 to B. In message M_1, T_1 is the WMN system time and N_A is a random nonce.

Step 2 Upon receiving M_2, B does as follows:

- Verify the public keys p_{D_A} and p_A of domain D_A and user A.
- If p_{D_A} and p_A are invalid, B terminates the communication. Otherwise, B calculates $P_A = \kappa_B^{-1} P'_A, k_1$ and K_A as shown in Fig. 2. Please note that κ_B^{-1} only needs to be pre-computed once and it is not time consuming. B then compares $k_1 \stackrel{?}{=} H_2(K_A)$, further decrypts $\{T_1, N_A\}_{k_1}$ to obtain T_1 and N_A. If T_1 and N_A are fresh, B believe that M_1 comes from A.
- B randomly picks b and continues to compute R_B, P_B, P'_B, K_B and K'.
- The session key k is then computed by B as $k = H_2(K_A||K_B||K')$.
- B chooses $T_2 = T_1 + \triangle T$ and $N \in \mathbb{Z}$ (the set of natural numbers). N is the maximum number of times that user A can revisit domain B within the time limit T_2. As will be described in Sect. 3.4, if a user frequently travels to a foreign domain and needs multiple mutual authentication and session keys, the CAKA protocol allows the user to save more on computation, if the visit is still within T_2 and the number of visits is smaller than N. User B then adds $(Flag_A, Flag_B, P_A, P_B, T_1, T_2, N, h, Num)$ to the cross-domain authentication list (CAL), where $h = H_2(Flag_A||T_1||T_2||N||k)$ and Num is the number of the current visits. If it is the first visit, $Num = 1$.
- B picks a random nonce N_B and send message M_2 to A.

Step 3 Upon receiving M_2, user A executes the following steps:

- Verify the public keys p_{D_B} and p_B of domain D_B and user B.
- If p_{D_B} and p_B are invalid, A terminates the communication. Otherwise, A calculates $P_B = \kappa_A^{-1} P'_B, K_B$ and K'. Similar to $\kappa_B^{-1}, \kappa_A^{-1}$ only needs to be pre-computed once.
- The session key k is obtained by A.
- Decrypt $\{T_1, T_2, N, N_A, N_B, h\}_k$ using the session key k to verify the freshness of the obtained nonce N_A. If N_A is verified, **user A authenticates B**.

Now users A and B complete the mutual authentication and agree on the session key k.

We briefly explain the rationale behind this CAKA protocol in Fig. 2. It is easy to see the following equation is true:

Table 1 Notation

Notation	Description
$\|\|$	Message concatenation
$\in_U Z_q^*$	Uniformly choose value from Z_q^*
D	Identifier of a WMN trusted domain
p_X	The public key of X, where x could be a domain D or a user U
s_X	The private key of X, where x could be a domain D or a user U
N	The maximum number that a user can repeatedly cross-domain in a specified time
ΔT	The validity period of repeated cross-domain authentication
T_1	The cross-domain application time
T_2	The expiration time of repeated cross-domain authentication
$\{M\}_k$	Symmetric encryption for message M under $k, \|k\| = l$
$Flag_U$	The identifier flag of mesh user U

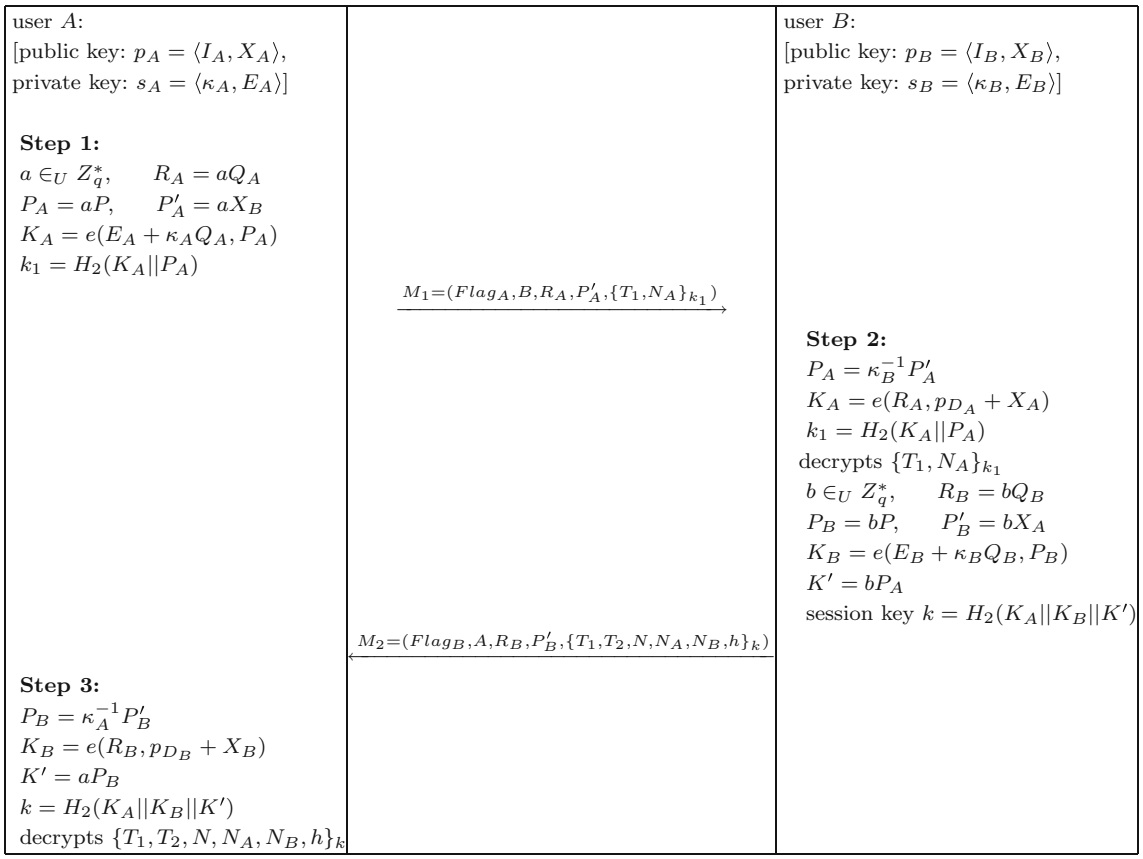


Fig. 2 Cross-domain authenticated key agreement (CAKA) protocols

$$\begin{aligned} K_A &= e(E_A + \kappa_A Q_A, P_A) \\ &= e((s_{D_A} + \kappa_A)Q_A, aP) \\ &= e((s_{D_A} + \kappa_A)P, aQ_A) \\ &= e(R_A, p_{D_A} + X_A). \end{aligned} \tag{1}$$

This equation indicates that, in *Step 2*, from the perspective of B, user A indeed comes from domain D_A with its identity $Q_A = H_1(I_A||X_A)$ and its public key $p_A = \langle I_A, X_A \rangle$. Similarly, in *Step 3*, user A also believe that message M_2 comes from user B with correct identity Q_B and public key p_B. It should be noted that if two users are from the same domain, it is easy for them to authenticate each other and establish a session key [1].

Because user B has its partial private key κ_B, in *Step 2*, only B can compute $P_A = \kappa_B^{-1} P'_A$ and $k_1 = H_2(K_A||P_A)$ and then decrypt $\{T_1, N_A\}_{k_1}$. For the same reason, user A can recover in *Step 3* $P_B = \kappa_A^{-1} P'_B, K' = aP_B$ and compute $k = H_2(K_A||K_B||K')$. User A decrypts $\{T_1, T_2, N, N_A, N_B, h\}_k$ using the computed session key k to accomplish the mutual authentication with user B because it knows that only B can generate the correct key k. Therefore only two messages M_1 and M_2 are needed for users A and B to authenticate each other and agree on a session key.

3.4 User *A* repeated cross-domain$_B$ authentication

Consider the situation where user A needs to enter foreign domain D_B multiple times within a short period of time. Imagine that every time user A will communicate with user B in D_B since B is the closest to A in a short time.

In this situation, our CAKA protocol is even simpler for user A to authenticate user B on each visit. If A crosses D_B repeatedly in a time period, say, $\triangle T < 12$ hours, A and B can perform the following steps:

1. A sends $M_3 = (Flag_A, \{T_1, T_2, N, k, h\}_{\overline{k_1}})$ to B, where $\overline{P_A} = \overline{a}P, \overline{P'_A} = \overline{a}X_B, \overline{a} \in_U Z_q^*, \overline{k_1} = H_2(\overline{P_A})$.
2. Upon receiving M_3, B computes $\overline{P_A} = \kappa_B^{-1}\overline{P'_A}, \overline{k_1} = H_2(\overline{P_A})$, and decrypts $\{T_1, T_2, N, k, h\}_{\overline{k_1}}$ by using $\overline{k_1}$. B checks the CAL and matches $h = H_2(Flag_A||T_1||T_2||N||k)$ in the CAL list. If the match is successful, then B sets $Num := Num + 1$ and returns $M_4 = \{OK\}_k$

to A. Then B can communicate with A by using k if necessary. Otherwise, B interrupts the communication and returns $M_5 = \text{NULL}$ to A. Usually, the time interval $\triangle T$ allowed for repeated domain crossing with the above simpler authentication may not be too long, but N can be large. If the cross-domain authentication time is valid, B should update Num. The update algorithm of the repeated cross-domain AKA is shown by the following pseudo-code (namly Algorithm 1):

Algorithm 1: : The process of B verifying T_2 and N

Require: $T_1, T_2, T_{delay}, N, Num$

1: B gets the current time T_{now} and the average time delay over the WMN network T_{delay}.
2: **if Timecheck**$(T_{now} + T_{delay} < T_2)$ and $Num < N$ **then**
3: B verifies the validity of p_{D_A}, I_A and matches h on the CAL.
4: **if** The verification holds **then**
5: Let $Num := Num + 1$ and B updates the new Num on the CAL.
6: **end if**
7: **end if**

3. A decrypts $\{OK\}_k$ with session key k.

This algorithm could save computation time and communication throughput of CAKA. It can greatly improve the protocol efficiency when A repeatedly crosses D_B over a short time period. To conclude this section, we summarize the properties of the proposed CAKA protocol.

1. The CAKA protocol is based on certificateless public key cryptography, thus does not have the key escrow problem.
2. Two users only need to exchange two messages to achieve mutual authentication and establish a session key.
3. No central authentication server is needed for the authenticated key agreement process, which avoids the bottleneck problem at a central server. Any MRs or MCs can carry out authentication and key agreement with low communication overheads.
4. The CAKA protocol is more efficient, with less computation and communication overhead, when a user frequently crosses the same domain.

4 Protocol analysis

4.1 System security analysis

This section analyzes how the desired security attributes listed in Section II-C are achieved, and how various attacks against our scheme are mitigated. The section consists of "System Security Analysis", "Other Security Consideration" and "Performance Analysis", a similar structure used in [2, 9].

Known-session key secrecy As demonstrated in Fig. 2, each session key k co-produced by users A and B is determined by random $a, b \in_U Z_q^*$ and $H_2(\cdot)$. Only the designated user B who has κ_B can compute the k_1 and get P_A, k_1, T_1, and N_A. Moreover, from the onewayness of hash functions, $k = H_2(e(R_A, p_{D_A} + X_A)||e(R_B, p_{D_B} + X_B)||abP) \neq H_2(e(R'_A, p_{D_A} + X_A)||e(R'_B, p_{D_B} + X_B)||a'b'P)$ if and only if $a \neq a'$ or $b \neq b'$. Since a and b are randomly selected by users A and B, respectively, k can be considered as a random output uniformly distributed over $\{0,1\}^l$. Furthermore, even if an adversary $\mathcal{A}$ gets k , he cannot infer a and b because of the onewayness of hash functions and the DLP assumption in $\mathbb{G}_2$. Thus, the compromise of one session key should not affect the secrecy of other session keys.

Forward secrecy Suppose that user A's long-term private key $s_A = \langle \kappa_A, E_A \rangle$ is leaked to an adversary $\mathcal{A}$. There are two possible consequences: (1) When $\mathcal{A}$ masquerades user A as a CAKA requester or a cross-domain AKA receiver (i.e., user A in Fig. 2), it can intercept previous conversation messages and compute $K_A = e(R_A, p_{D_A} + X_A), K_B = e(R_B, p_{D_B} + X_B)$, but it cannot calculate K' since the adversary cannot obtain a from R_A or obtain b from R_B according to the DLP and CDHP assumptions in $\mathbb{G}_1$. When $\mathcal{A}$ masquerades user B as (i.e., user B in Fig. 2), it cannot obtain b from R_B either because the same assumptions.Thus, the secrecy of previous session keys is not compromised even if user A's private key or user B's private key is compromised, ensuring the *Perfect Forward Security*.

Key-compromise impersonation resilience In the CAKA protocol, compromising one of the two parties' private keys does not affect the secrecy of the session keys established previously. Suppose that an adversary obtains user A's long-term private key $s_A = \langle \kappa_A, E_A \rangle$. The adversary can impersonate user A to initiate a CAKA requester

or masquerade an AKA receiver. However, the adversary cannot get other users' long-time private keys from the interactions. Thus, the CAKA protocol ensure that even if a user's private key is compromised, security of other users' private keys will not be affected.

Unknown key-sharing resilience In our CAKA protocol, the session key can only be calculated by the CAKA requester and receiver, because of the technique of the designated verifier. Any other third party cannot force user A to share a key with other users unless both sides conduct a CAKA protocol in Fig. 2. Since only the designated receiver B can get P_A, k_1, session key k and decrypt message M_1. The decryption of M_1 needs implicit authentication. It has been proven that the unknown key-sharing resilience is implied by the implicit authentication property [16].

No key control Since the session key k is determined by the random values a, b and $H_2(\cdot)$ $(k = H_2(e(aQ_A, p_{D_A} + X_A)||e(bQ_B, p_{D_B} + X_B)||abP))$, neither the requester nor the receiver can control k to be a preselected value. Although the requester can select an ideal hash value by changing random number a, the avalanche effect of hash functions will produce great difference with a slight difference in a or b.

4.2 Other security considerations

Even if the security properties of the previous section are attained, additional considerations must be made to ensure secure implementations. The man-in-the-middle attack(the MITM attack) is one of the most difficult attacks to resist and one of the most vulnerable to two-party protocol. In CL-PKC, we often consider two types of adversaries who would mount the MITM attack. A type I adversary $\mathcal{A}_1$ can change public key of clients at will, but does not access to the system master key. The type II adversary $\mathcal{A}_2$ is equipped with master key, but is not allowed to replace the public key of clients. These adversary models is to capture the attacks from an eavesdropping ISP (or say PKG/KGC in different literatures).In the following the security analysis of our CAKA, we consider the above two types of adversaries.

Assume that the most common adversary $\mathcal{A}$ mounts an MIMT attack on the key exchange in an undetectable way. For instance, $\mathcal{A}$ can replace $\mathbf{M_1}$ sent by U_A with $\mathbf{M_1'} = \{Flag_A, R_A', P_A'', T_1', N_A'\}_{k_1'}$, and similarily can substitute $\mathbf{M_2}$ with $\mathbf{M_2'} = \{Flag_A, R_B, P_B', T_1', T_2, N, N_A', N_B, h\}_{k'}$ sent by U_B . However, $\mathcal{A}$ do not know κ_A^{-1} and hence cannot recover P_B from P_B' because only designated client can correctly decrypt the encrypted messages. Therefore, $\mathcal{A}$ acts as a relay node and cannot get the session key or private keys.

The MITM attack may be mounted by the ISP or the adversary $\mathcal{A}_1$ or $\mathcal{A}_2$ who has obtained the ISP's master key. This can be easily launched because the CL-AKA is easy to go through the public-key replacement attack or say that the key replacement attack is one basic attack against a certificateless scheme. One way to fight against the public key replacement attack is to bind a user's public and private keys, as noted in [16]. We adopt such a technique by incorporating user $_A$'s identity I_A and fixed public key X_A into the partial private key $E_A = s_{D_A} H_1(I_A||X_A)$. Therefore, an adversary should not be able to obtain the partial private key s_A from the replaced public key without knowing the private key. Even the adversary $\mathcal{A}_1$ can replace the public key $X_A' = \kappa'_A P, Q_A' = H_1(I_A||X_A'), R_A = a'Q_A'$ and impersonate user $_A$ successfully, he still cannot compute correct P_B from P_B' , $a'bP$ under the assumption of CDHP. For the adversary $\mathcal{A}_2$, he can access the domain master key s_D and does not allow to replace the public keys of clients. Even if $\mathcal{A}_2$ knows the partial private key E_A, he still cannot masquerade user $_A$ to compute K_A without knowing s_A. Hence, both typed adversaries cannot compute the session key without knowing the full private keys of both clients. Therefore, the CAKA can withstand the MITM attack.

4.3 Performance analysis

In this section, an efficiency analysis on communication and computation complexities of the proposed CAKA protocol is given. Communication complexity is measured by number of messages sent from one user to another user, the number of the communication hops, and the bytes of the message. We compare our CAKA protocol to the ZFW [1] and HA [8] protocols, both of which provide cross-domain authenticated key agreement for multi-domain WMNs with detailed algorithms that are most similar to our scheme.

Communication overhead Compared to ZFW [1] and HA [8], our CAKA protocol is more efficient with less number of messages sent and less number of communication hops. *Number of messages sent*: As demonstrated in Fig. 3, the ZFW [1] protocol needs a total of eight messages sent to accomplish a cross-domain authenticated key agreement and the HA [8] protocol needs a total of six messages sent. However, the CAKA only needs two messages. *Number of communication hops*: Our CAKA protocol can be seen as one-hop communication, considering the link between MRs are usually high speed wireless links. Even if the CAKA requester and receiver cannot communicate in one-hop, the intermediate nodes MRs just relay and forward the message without doing any computation. However, in ZFW [1] and HA [8], all cross-domain AKA protocols have to be performed by the home authentication

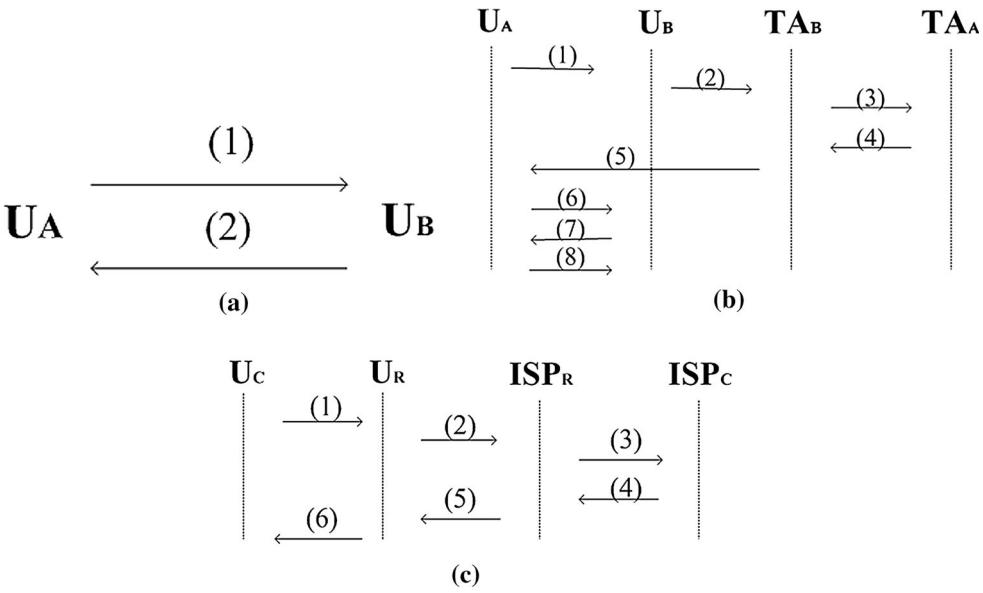


Fig. 3 Communication costs comparison of CAKA , ZFW [1] and HA [8]. (**a**) The flowchart of our protocol, (**b**) The flowchart of ZFW [1] protocol, (**c**) the flowchart of HA [8] protocol

server, requiring multi-hop communications in most cases. No involvement of ISPs is required in our CAKA protocol and hence there is no system bottleneck problem. So the AKA performs more efficiently with less propagation and smaller handover delay. This will definitely facilitate a fast hand-off process for real-time service.

Computation overhead We choose the modified Tate pairing on an MNT curve with embedding degress 6 and 160-bit q [22]. For simplicity, let T_m and T_p denote the times to perform one point multiplication and pairing evaluation in $\mathbb{G}_1$, respectively. Similarly, T_e and T_h are the times for performing an AES symmetric encryption and a H_2 hash operation in our protocol. The computation overhead is compared to the latest cross-domain AKA protocols given in ZFW [1] and HA [8] . The results are listed in Table 2 (HA [8] presented a general protocol. Usually a signature and an encryption based on bilinear pairing at least contain one T_m and one T_e, respectively).

We benchmarked all of these operations using the PBC library [23] (version pbc-0.4.7) on a 32-bit, 2.13 GHz Intel core based, dual-core processor machine with 2GB main memory, running Debian Linux. The computation time is recorded as below. User A needs 48.42 ms (*Step 1*) + 63.19 ms (*Step 3*) and user B needs 93.34 ms (*Step 2*), in the CAKA protocol shown in Fig. 2. Since the time for performing the most time-consuming *MapToPoint* H_1 hash function only is 0.6 ms [24], and AES encryption algorithm needs 1s to encrypt 6.556 M files. So computation time of AES and hash algorithm is negligible. From Table 2, we can see the computation cost of the CAKA is approximately higher than that of the ZFW [1] and the HA [8]. However, the proposed CAKA protocol has much smaller number of messages sent (Fig. 3), the communication latency of CAKA is significantly smaller than those of the other two protocols.

Storage overhead In CAKAnetwork users maybe carry resource-constrained devices such PDAs and smart phones to cross-domain. Hence, storage overhead should be affordable to the modern pervasive devices. Similar to other existing work [24], we assume the bytes of difference piece of information to be those shown in Table 3. The message M_1 sent from CAKA requester to CAKA receiver in Fig. 2 has 102 bytes and message M_2 on the reverse direction has 134 bytes. In the case when user A revisits a

Table 2 Performance and security properties of three schemes

Protocol	User A	User B	Key escrow	Involvement of TTP[a]	Flow	Domain authentication server[b]
CAKA	$1T_e + 5T_m + 2T_p + 2T_h$	$1T_e + 3T_m + 2T_p + 1T_h$	No	No	2	No
ZFW [1]	$1T_e + 1T_m + 2T_p + 4T_h + 1T_{sig}$	$1T_e + 1T_m + 2T_p + 4T_h + 1T_{sig}$	Yes	Yes	8	Yes
HA [8]	$2T_e + 2T_m + 1T_h$(at least)	$2T_e + 2T_m + 1T_h$(at least)	Yes	Yes	6	Yes

[a] ZFW's TTP is TA and HA's TTP is ISP

[b] ZFW needs shared keys between router and TA; HA needs a secure channel between different ISPs

Table 3 Message types sent by two communicating parties

Domain name D (bytes)	User I_u (bytes)	$\|P\|$ in $\mathbb{G}_1$ (bytes)	AES encryption (bytes)	SHA-1 (bytes)	Nonce (bytes)	Timestamp (bytes)
2	4	20	16	20	4	4

foreign domain multiple times within a short period, as described in Sect. 3.4, the messages are even shorter since $|M_3| = 118$ bytes, $|M_4| = 2$ bytes and $|M_5| = 1$ byte, which should be affordable to most of the modern pervasive devices.

We also consider the energy loss for mobile terminals. From [25], we know the energy consumption for multiplication, pairing [25], AES encryption and hash SHA1 are 30.02, 423.87 mJ, 1.62 and 5.90 uJ/byte, respectively. Data in [25] showed even with 5 % of the energy available from a miniature 30 mAh battery, a node can perform 173 ECC-160 handshakes. Our scheme only needs energy consumption about 998.02 mJ (0.998 J) for user A and 937.96 mJ (0.93796 J) for user B to complete the CAKA protocol in Fig. 2, which is insignificant to today's mobile tablets and smartphones with strong ARM processors.

5 Conclusion and ongiong work

The major concern we have addressed in this paper is to enable the secure communication between two users from different WMN administrative or security domains potentially managed by different operators. Although this is not a new problem, current approaches fall short of being able to offer a satisfactory solution. In this paper, we have presented a CL-based hierarchical security architecture and trust model to develop a novel mutual authentication and key agreement protocol for cross-domain network environment such as WMNs. The architecture is simple and the trust model is practical without key-escrow problem and certificate revocation. Each user firstly registers certain domain parameters with the help of trusted domain server ISP. Then, a user in one domain can prove his identity to another user another domain and negotiate the session key for subsequent communications. The mutual AKA can be efficiently performed by only two-round interactions. Each client can be the verifier and no central verification servers are needed. Security analysis has demonstrated that the proposed scheme exhibits excellent security property. Compared with the existing cross-domain AKA protocols, our CAKA is very efficient in terms of computation and communication overheads. Thus it is suitable for various kinds of roaming scenarios.

In term of future work, we plan to incorporate other security attributes such as user privacy and user accountability (revocable anonymity) [2, 9, 24] into our approach to make CAKA more comprehensive.

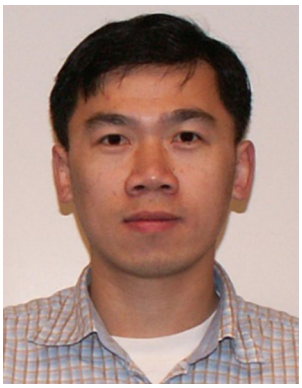
Acknowledgments This work was partly supported by the National Natural Science Foundation of China under Grants 61402275, 61373150, 61379145, 61232016, U1405254, 61202317, 61272436, Shaanxi Province Natural Science Basic Research Program Funded Project 2015JM6263, the PAPD fund, the Fundamental Research Funds for the Central Universities under Grant GK201402004.

References

1. Zhu, X., Fang, Y., & Wang, Y. (2010). How to secure multi-domain wireless mesh networks. *Wireless Networks*, *16*(5), 1215–1222.
2. Ren, K., & Lou, W. (2008). A sophisticated privacy-enhanced yet accountable security framework for metropolitan wireless mesh networks. In *The 28th international conference on distributed computing systems, 2008. ICDCS'08* (pp. 286–294). New York: IEEE.
3. Zhang, Y., & Fang, Y. (2006). ARSA: An attack-resilient security architecture for multihop wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, *24*(10), 1916–1928.
4. Li, C., & Nguyen, U. T. (2010). Fast authentication for mobile clients in wireless mesh networks. In *2010 23rd Canadian conference on electrical and computer engineering (CCECE)* (pp. 1–8). New York: IEEE.
5. Gao, T., Guo, N., & Yim, K. (2012). Delegation-based mutual authentication scheme for multi-operator wireless mesh network. In *2012 sixth international conference on innovative mobile and internet services in ubiquitous computing (IMIS)* (pp. 143–147). New York: IEEE.
6. Wong, F.L., & Lim, H.W. (2007). Identity-based and inter-domain password authenticated key exchange for lightweight clients. In *21st International conference on advanced information networking and applications workshops, AINAW'07* (vol. 1, pp. 544–550). New York: IEEE.
7. Chen, L., Lim, H. W., & Yang, G. (2013). Cross-domain password-based authenticated key exchange revisited. In *2013 Proceedings IEEE INFOCOM* (pp. 1052–1060). IEEE.
8. He, B., Agrawal, D. P. (2010). An identity-based authentication and key establishment scheme for multi-operator maintained wireless mesh networks. In *2010 IEEE 7th international conference on mobile adhoc and sensor systems (MASS)* (pp. 71–78). New York: IEEE.
9. Ren, K., Yu, S., Lou, W., & Zhang, Y. (2010). Peace: A novel privacy-enhanced yet accountable security framework for metropolitan wireless mesh networks. *IEEE Transactions on Parallel and Distributed Systems*, *21*(2), 203–215.
10. Shim, K. (2003). Efficient ID-based authenticated key agreement protocol based on weil pairing. *Electronics Letters*, *39*(8), 653–654.
11. Wang, S., Cao, Z., Choo, K.-K. R., & Wang, L. (2009). An improved identity-based key agreement protocol and its security proof. *Information Sciences*, *179*(3), 307–318.

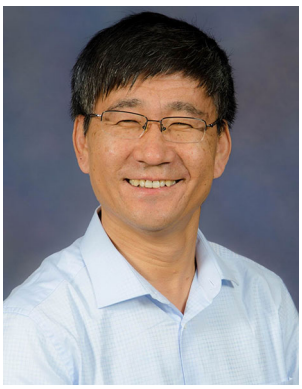
12. Wang, S., Cao, Z., Cheng, Z., & Choo, K.-K. R. (2009). Perfect forward secure identity-based authenticated key agreement protocol in the escrow mode. *Science in China Series F: Information Sciences*, *52*(8), 1358–1370.
13. I. C. S. L. M. S. Committee et al. (2009). Ieee p802.11s/d2.06: Part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications. Amendment 10: Mesh networking.
14. Zhang, L., Zhang, F., Wu, Q., & Domingo-Ferrer, J. (2010). Simulatable certificateless two-party authenticated key agreement protocol. *Information Sciences*, *180*(6), 1020–1030.
15. Mokhtarnameh, R., Ho, S. B., & Muthuvelu, N. (2011). An enhanced certificateless authenticated key agreement protocol. In *13th International conference on advanced communication technology (ICACT)* (pp. 802–806). New York: IEEE.
16. Shi, Y., & Li, J. (2007). Two-party authenticated key agreement in certificateless public key cryptography. *Wuhan University Journal of Natural Sciences*, *12*(1), 71–74.
17. Al-Riyami, S.S., & Paterson, K.G. (2003). Certificateless public key cryptography. In *Advances in cryptology-ASIACRYPT 2003* (pp. 452–473). Berlin: Springer.
18. Cheng, Z., & Comley, R. (2005). Efficient certificateless public key encryption. *IACR Cryptology ePrint Archive*, *2005*, 12.
19. Boneh, D., & Franklin, M. (2001). Identity-based encryption from the weil pairing. In *Advances in Cryptology—CRYPTO 2001* (pp. 213–229). Berlin: Springer.
20. Guo, H., Li, Z., Mu, Y., & Zhang, X. (2011). Provably secure identity-based authenticated key agreement protocols with malicious private key generators. *Information Sciences*, *181*(3), 628–647.
21. Zhang, Z., Wong, D. S., Xu, J., & Feng, D. (2006). Certificateless public-key signature: Security model and efficient construction. In *Applied cryptography and network security* (pp. 293–308). Berlin: Springer.
22. http://crypto.stanford.edu/pbc/times.html.
23. http://crypto.stanford.edu/pbc/.
24. Zhu, X., Jiang, S., Wang, L., & Li, H. (2013). Efficient privacy-preserving authentication for vehicular ad hoc networks. *IEEE Transaction on Vehicular Technology*, *63*(2), 907–919.
25. Wander, A. S., Gura, N., Eberle, H., Gupta, V., & Shantz, S. C. (2005). Energy analysis of public-key cryptography for wireless sensor networks. In *Third IEEE international conference on pervasive computing and communications, PerCom 2005* (pp. 324–328). New York: IEEE.

Yanping Li received her B.S. degree from DaTong University, DaTong, China, her M.S. degree in Applied Mathematics from Shaanxi Normal University, Xi'an, China, in 2004, and her PhD degree in Cryptography from Xidian University, Xi'an, China, in 2009. She is currently an associate professor with the School of Mathematics and Information Science, Shaanxi Normal University, Xi'an, China. She was a visiting research scholar with Department of Electrical and Computer Engineering at University of Florida, Gainesville, Florida, USA, from September 2013 to September 2014. Her research interests include public key cryptography and its applications.



Weifeng Chen received his B.S. from Beijing University, China, his M.S. from Chinese Academy of Sciences, Beijing, China, and his Ph.D. degree from University of Massachusetts at Amherst, Massachusetts, USA, all in computer science. He is currently an associate professor in the Department of Mathematics, Computer Science and Information Systems at California University of Pennsylvania. His research interests include network security, privacy and protocol design.

Zhiping Cai received his B.S., M.S. and Ph.D. degrees in computer science and technology with honor all from National University of Defense Technology (NUDT), Changsha, Hunan, in July 1996, April 2002 and December 2005, respectively. He is currently an associate Professor in Department of Networking Engineering, College of Computer, NUDT, Changsha, Hunan, China. His doctoral dissertation won the Outstanding Dissertation Award of the China PLA. His research interests include network security, network measurement and network virtualization. He is a member of ACM and IEEE.



Yuguang Fang received an M.S. degree from Qufu Normal University, Shandong, China in 1987, a Ph.D. degree from Case Western Reserve University in 1994 and a Ph.D. degree from Boston University in 1997. He joined the Department of Electrical and Computer Engineering at University of Florida in 2000 and has been a full professor since 2005. He held a University of Florida Research Foundation (UFRF) Professorship (2006–2009), a Changjiang Scholar Chair Professorship with Xidian University, China (2008–2011), and a Guest Chair Professorship with Tsinghua University, China, from (2009–2012). Dr. Fang received the US National Science Foundation Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002, 2015 IEEE Communications Society CISTC Technical Recognition Award, 2014 IEEE Communications Society WTC Recognition Award, and the Best Paper Award from IEEE ICNP (2006). He has also received a 2010–2011 UF Doctoral Dissertation Advisor/Mentoring Award, 2011 Florida Blue Key/UF Homecoming Distinguished Faculty Award, and the 2009 UF College of Engineering Faculty Mentoring Award. He is the Editor-in-Chief of IEEE Transactions on Vehicular Technology, was the Editor-in-Chief of IEEE Wireless

Communications (2009–2012), and serves/served on several editorial boards of journals including IEEE Transactions on Mobile Computing (2003–2008, 2011-present), IEEE Transactions on Communications (2000–2011), and IEEE Transactions on Wireless Communications (2002–2009). He has been actively participating in conference organizations such as serving as the Technical Program Co-Chair for IEEE INOFOCOM'2014 and the Technical Program Vice-Chair for IEEE INFOCOM'2005. He is a fellow of the IEEE.