AC-PKI: Anonymous and Certificateless Public-Key Infrastructure for Mobile Ad Hoc Networks

Yanchao Zhang*, Wei Liu*, Wenjing Lou†, Yuguang Fang*, and Younggoo Kwon‡
*Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611
Email: {yczhang@, liuw@, fang@ece}ufl.edu
†Department of Electrical and Computer Engineering
Worcester Polytechnic Institute, Worcester, MA 01609
Email: wjlou@ece.wpi.edu
‡Department of Computer Engineering
Sejong University, Seoul 143-747, Korea
Email: ygkwon@sejong.ac.kr

***Abstract*— This paper studies public-key management, a fundamental problem in providing security support for mobile ad hoc networks. The infrastructureless nature and network dynamics of ad hoc networks make the conventional certificate-based public-key solutions less suitable. To tackle this problem, we propose a novel Anonymous and Certificateless Public-Key Infrastructure (AC-PKI) for ad hoc networks. AC-PKI enables public-key services with certificateless public keys and thus avoids the complicated certificate management inevitable in conventional certificate-based solutions. To satisfy the demand for private keys during network operation, we employ the secret-sharing technique to distribute a system master-key among a pre-selected set of nodes, called D-PKGs, which offer a collaborative private-key-generation service. In addition, we identify pinpoint attacks against D-PKGs and propose anonymizing D-PKGs as the countermeasure. Moreover, we determine the optimal secret-sharing parameters to achieve the maximum security.**

I. Introduction

Public-key cryptography (PKC) is appealing in offering security support for mobile ad hoc networks (MANETs) due to its effectiveness in facilitating essential security services such as digital signatures and key management. In a conventional public-key infrastructure (PKI), a centralized Certification Authority (CA) is indispensable for managing public key certificates used to generate confidence in the legitimacy of public keys. However, it is difficult to deploy such a certificate-based PKI in MANETs for the lack of infrastructure and other centralized services. Although the secret-sharing technique [1] could be employed to distribute the CA's role to a pre-selected set of nodes, termed distributed CAs [2]–[6], resource-constrained ad hoc networks might be still unable to afford the rather complicated certificate management, including revocation, storage and distribution, and the computational costs of certificate verification.

Identity-based public-key cryptography (ID-PKC) [7] arises as a promising candidate during our search for a realistic, lightweight and secure PKC solution for MANETs. ID-PKC simplifies public-key management by allowing a public key to be directly derived from publicly available information that uniquely and undeniably identifies users, e.g., telephone numbers, email addresses and social security numbers, thus eliminating the need for public-key certificates and CAs. This attractive feature makes ID-PKC ideal for the wireless arena, where computational power, memory and battery life are more constrained.

Inspired by ID-PKC, this paper proposes a novel *Anonymous and Certificateless Public-Key Infrastructure* (AC-PKI) for MANETs. Our contributions are mainly threefold. First, we apply Shamir's secret-sharing technique [1] to distribute the system trust, essentially a system master-key, across a pre-selected set of nodes, called distributed private-key generators (D-PKGs). D-PKGs collaboratively offer a prerequisite private-key-generation (PKG) service during network operation. Second, we propose offering D-PKGs anonymity protection to defend against pinpoint attacks that are quite easy to conduct and may cause devastating consequences in MANETs. Last, we determine the optimal secret-sharing parameters for achieving the maximum security.

The rest of this paper is structured as follows. Section II introduces the pairing technique. Section III details the AC-PKI system design. The related work is reviewed in Section IV and some concluding remarks are given in Section V.

II. Introduction to Paring

The formulation of ID-PKC can date back to 1984 [7]. However, ID-PKC underwent a rather slow development until the application of pairing. The basic concept of pairing is outlined as follows. Let $\mathbb{G}_1, \mathbb{G}_2$ be two groups of the same prime order q. We view $\mathbb{G}_1$ as an additive group and $\mathbb{G}_2$ as a multiplicative group throughout the paper. A pairing is a computable *bilinear map* $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfying the following properties:

This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N000140210464 and under grant N000140210554.

1. *Bilinearity*: $\forall\ P, Q, R, S \in \mathbb{G}_1$, we have

$$\hat{e}(P+Q, R+S) = \hat{e}(P,R)\hat{e}(P,S)\hat{e}(Q,R)\hat{e}(Q,S).^{1} \quad (1)$$

2. *Non-degeneracy*: If $\hat{e}(P,Q) = 1$ for all $Q \in \mathbb{G}_1$, then P must be the identity element in $\mathbb{G}_1$.
3. *Computability*: There exists an efficient algorithm to compute $\hat{e}(P,Q)$ for all $P, Q \in \mathbb{G}_1$.

Modified Weil pairing [8] and Tate pairing [9] on super-singular elliptic curves are examples of such bilinear maps, for which the *Bilinear Diffie-Hellman Problem* (BDHP) is believed to be hard. That is, it is believed that, given $< P, xP, yP, zP >$ for random $x, y, z \in \mathbb{Z}_q^{*}$[2] and $P \in \mathbb{G}_1$, there is no algorithm running in expected polynomial time, which can compute $\hat{e}(P,P)^{xyz} \in \mathbb{G}_2$ with non-negligible probability. We refer readers to [8], [9] for further details on pairing and to [10] for an exemplary pairing implementation.

III. AC-PKI System Design

A. System Overview

Our proposed AC-PKI is a lightweight and secure PKC solution for MANETs. We consider a MANET consisting of $\mathcal{N}$ non-adversarial nodes affiliated with the same party Ψ ($|\Psi| = \mathcal{N}$), where the network size $\mathcal{N}$ might dynamically changes with node join/leave. Each node i has a unique non-zero identifier ID_i , which can be any type of string that uniquely and undeniably identifies node i, e.g., its email address, its role description, its social security number, its telephone number, etc.

During the bootstrapping phase, guided by [8], a trusted authority (TA), e.g., the system administrator or network planner, first determines an appropriate prime q, two q-order cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$, a pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, and an arbitrary generator $W \in \mathbb{G}_1$. He/she then chooses a random $g \in \mathbb{Z}_q^{*}$ as the system *master-key* and determines a cryptographic hash function $H_1 : \{0,1\}^{*} \rightarrow \mathbb{G}_1^{*}$[3], mapping arbitrary strings to elements in $\mathbb{G}_1^{*}$. Each node $i \in \Psi$ is furnished with a unique *master private key* $S_i = gH_1(ID_i) \in \mathbb{G}_1$ and the public system parameters $< \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_1, W >$.

The security of any ID-PKC cryptosystem like AC-PKI relies on the secrecy of the system master-key used to generate nodal private keys. Since the Discrete Logarithm Problem (DLP)[4] is believed to be hard in $\mathbb{G}_1$ [8], given any $< ID_i, S_i >$ pair, the system master-key g cannot be deduced with non-negligible probability. It means that (1) each node i is totally blind to g, and (2) adversaries cannot determine g even after compromising an arbitrary number of legitimate nodes. The system master-key g is only known to the TA who usually does not appear in the resulting network.

Notice that we can remove the dependence on the TA by allowing non-adversarial nodes to negotiate the above system parameters using *Byzantine agreement* [11] over a particular location-limited channel, e.g., by physical contact. For the lack of space, we ignore this self-organizing extension in this paper. However, it is worth pointing out that this self-organizing way and the previous TA-dependent way both require some kind of *a priori* security context shared among all the non-adversarial nodes. The rationale here is that any security mechanism can only help in transforming and transferring the trust assumptions in the prior context, but cannot create trust from scratch.

The following example can help understand how public-key encryption/decryption services are accomplished in AC-PKI. Suppose nodes Alice and Bob are both legitimate members of Ψ. When having some secret information *msg* for Bob, Alice no longer needs to obtain from anywhere Bob's public-key certificate and verifies it in advance, as what she has to do in a conventional certificate-based PKI. Instead, she can directly use Bob's identifier ID_B as his public key and generate the ciphertext as $IBE(ID_B, msg)$. Here $IBE()$ represents an identity-based encryption (IBE) function built on the above public system parameters. Many such IBE functions as [8], [9] have been proposed in the literature, which can guarantee that no other node than Bob, which must hold the valid private key corresponding to "ID_B", can correctly decrypt the ciphertext. Similarly, signature generation/verification services can be fulfilled by harnessing any identity-based signing function, such as the one defined in [12].

B. A Basic Private-Key-Generation Scheme

In the previous example, Bob's identifier is used as his public key. In fact, any type of string can be a public key in AC-PKI. For instance, Alice can encrypt a message using as his public key Bob's identifier concatenated with any desired information, e.g., "ID_B|| current-date || role = captain", where "||" denotes the concatenation of messages. By doing this, Alice attempts to make sure that if and only if Bob is a captain who holds the valid private key on the specified date, could he decrypt the ciphertext. An interesting property here is that Bob does not need to possess the corresponding private key beforehand. He can request the private key from the TA after receiving the ciphertext. Such on-demand means of private-key requests coincides well with the dynamic, resource-constrained nature of MANETs. Obviously, we can accomplish more nice properties that do not exist in a conventional PKI setting by concatenating the destination identifier with different information. Such a nice feature, however, poses the demand for a PKG scheme during network operation: the destination may not have the needful private key in hand so that it should be able to obtain it from somewhere.

To meet this demand, it is necessary to introduce the TA functionality into the network from which mobile nodes can request private keys whenever needed. It is, however,

[1] In particular, $\forall\ P, Q \in \mathbb{G}_1,\ \forall\ a, b \in \mathbb{Z}_q^{*},\ \hat{e}(aP, bQ) = \hat{e}(aP, Q)^b = \hat{e}(P, bQ)^a = \hat{e}(P, Q)^{ab}$ etc.

[2] $\mathbb{Z}_q^{*}$ is the *multiplicative group* of integers modulo q. In particular, if q is a prime, $\mathbb{Z}_q^{*} = \{a \mid 1 \leq a \leq q-1\}$.

[3] $\mathbb{G}_1^{*}$ denotes the set $\mathbb{G}_1^{*} \setminus \{\mathcal{O}\}$ where $\mathcal{O}$ is the identity element in the group $\mathbb{G}_1$.

[4] The DLP in the additive group $\mathbb{G}_1$ is as follows: given two group elements P and Q, find an integer $n \in \mathbb{Z}_q^{*}$ such that $Q = nP$ whenever such an integer exists.

problematic to use a single TA in MANETs because it may become the single point of failure. To enable a robust PKG scheme, Shamir's (*t,n*) secret-sharing technique [1] can be employed to distribute the TA functionality among a set of pre-selected nodes such that as long as there are no less than *t* such nodes being functional, mobile nodes can still ask for private keys from them. For this purpose, the TA needs to supplement the previous network bootstrapping process with the following operations:

1. Determine a (t-1)-degree ($1 \leq t \leq \mathcal{N}$) polynomial, $h(x) = g + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \pmod q$, with random coefficients a_i ($1 \leq i \leq t-1$) in $\mathbb{Z}_q^*$. Here g is the system master-key chosen previously.
2. Select n ($t \leq n \leq \mathcal{N}$) nodes out of the total $\mathcal{N}$ nodes, either without distinction or by considering node heterogeneity and choosing physically more secure, computationally more powerful, or more trustworthy ones. We call these nodes *D-PKG*s (distributed private key generators), denoted by $\mathcal{SH} = \{SH_k | 1 \leq k \leq n\}$ consisting of the identifiers of D-PKGs.
3. Calculate *n* shares of g as $g_k = h(k)$ for $k \in \{1...n\}$, and assign g_k to SH_k.
4. Calculate a set of share commitments as $\mathcal{SC} = \{W_k^{pub} = g_k W \in \mathbb{G}_1 | 1 \leq k \leq n\}$.

$\mathcal{SH}$ and $\mathcal{SC}$ are appended to the public system parameters known to every node. Based on the Lagrange interpolation, any combination of t D-PKGs with indices x_i ($1 \leq i \leq t, 1 \leq x_i \leq n$) in $\mathcal{SH}$ can collectively reconstruct the system master-key g by computing

$$g = \sum\nolimits_{i=1}^{t} \lambda_i g_{x_i}, \text{ where } \lambda_i = \prod\nolimits_{j=1, j \neq i}^{t} \frac{x_j}{x_j - x_i}. \quad (2)$$

However, any less than *t* D-PKGs do not suffice to determine g and hence to generate valid private keys. Obviously, our scheme can tolerate the compromise of up to *t* nodes at any point of time. We will discuss how to choose the values of t and n to achieve the highest level of security in Section III-D. The following example is used to illustrate how these D-PKGs collaboratively provide the PKG service.

Suppose node Bob (ID_B) receives some messages encrypted under the public key "$ID_B || otherInfo$" to which he has no corresponding private key in hand. He can easily obtain it by randomly choosing *t* D-PKGs from $\mathcal{SH}$ and then sending each of them a private-key sub-request containing "$ID_B||$ otherInfo". The sub-requests can be sent in either plaintexts or ciphertexts generated with the chosen D-PKGs' identifiers as their respective public keys if needed.

Upon receiving such a sub-request, each chosen D-PKG, say SH_{x_i}, sends back to Bob a sub-reply containing a partial private key, $S_{B,x_i}^r = g_{x_i} H_1(ID_B || otherInfo)$. Our scheme has the *verifiable* feature that, for each sub-reply, Bob can utilize Eq. (1) and the public share commitment $W_{x_i}^{pub}$ of SH_{x_i} to verify its authenticity by checking

$$\hat{e}(S_{B,x_i}^r, W) = \hat{e}(H_1(ID_B || otherInfo), W_{x_i}^{pub}). \quad (3)$$

If the condition does not hold for any D-PKG SH_{x_i}, Bob knows that there must be something wrong with it. For example, the sub-reply from SH_{x_i} might have undergone transmission errors, or even SH_{x_i} itself might have been physically or logically controlled by adversaries. Bob can then request a new private-key piece from another unused D-PKG. After obtaining *t* authentic private-key pieces, Alice can plug them into Eq. (2) and calculate the complete private key desired as

$$S_B^r = \sum\nolimits_{i=1}^{t} \lambda_i S_{B,x_i}^r = g H_1(ID_B || otherInfo) \in \mathbb{G}_1 \quad (4)$$

In the above process, we assume that there exist out-of-band mechanisms or policies for D-PKGs to determine whether Bob is still a qualified party member to be issued the requested private key. For example, if D-PKGs have found the misbehavior of Bob through some means, they might reject his private-key request. How to establish such mechanisms or policies is application-dependent and outside the scope of this paper. In addition, since malicious eavesdropping is easy to conduct in the open wireless arena, each chosen D-PKG should encrypt the sub-reply using Bob's identifier ID_B as his public key so that Bob can decrypt the ciphertext with his master private key. Otherwise, adversaries may be able to intercept all the partial private keys so as to construct S_B^r that can be used to decrypt the previously eavesdropped messages sent to Bob.

Notice that it is possible that Bob cannot receive enough t sub-replies in a timely manner due to the error-prone nature of wireless links and the possibly poor network connectivity. If this happens, he would fail to construct the desired private key. Several methods can be employed to address this situation. First, Bob can send private-key sub-requests to $(t+e)$ instead of t D-PKGs, where e is a tunable integer. Second, Bob can utilize as many D-PKGs as possible to which routing paths can be found in its routing table, instead of purely randomly picking t D-PKGs from the entire set $\mathcal{SH}$. Last, Bob can resend in the face of failure private-key sub-requests to D-PKGs from which no sub-replies are received until the allowed maximum number of retry rounds is reached. For the lack of space, we will report the impact of these measures on private-key requests in a separate paper.

C. Providing Anonymity Protection for D-PKGs

The shared wireless medium of MANETs introduces abundant opportunities for passive eavesdropping on data communications. Adversaries can easily overhear all the messages "flying in the air" without physically trespassing a node. On the other hand, node identifiers are left bare without any protection in common ad hoc routing protocols such as AODV. One of the severe consequences is that, for a given routing/data packet, adversaries are able to collaboratively ascertain (without much effort) the identifiers of both the local transmitter/receiver and the end-to-end source/ destination. Even worse, they might locate and trace certain critical nodes, such as distributed CAs [2] or D-PKGs in our AC-PKI, based on the node identifier information leaked in routing/data packets. This facilitates the pinpoint attacks against the locked critical targets. In contrast to active attacks like radio jamming

or other forms of more "visible" DoS attacks, such once-passive-then-active attacks, called *traffic analysis attacks* in [10], are more dangerous because they are much more subtle, "invisible", and difficult to detect before severe damage occurs.

Assume that adversaries are powerful enough in the sense that they can physically or logically trespass any given node soon or later if they intend to do so. Also, a node is said to be *compromised* if adversaries trespass it and retrieve all its secret information, and *corrupted* if adversaries paralyze its correct functioning but cannot steal its secret information. It is easy to see that both schemes [2]–[6] and our basic PKG scheme do not work once adversaries compromise no less than *t* distributed CAs [2]–[6] or D-PKGs, or corrupt no less than (*n-t*) distributed CAs or D-PKGs. Since the identifiers of distributed CAs or D-PKGs are assumed to be public knowledge, adversaries can easily achieve their objective by performing the traffic analysis attacks described above. It is, therefore, necessary to figure out effective measures to protect such critical nodes as distributed CAs or D-PKGs from traffic analysis attacks.

Our countermeasure is to make D-PKGs *anonymous* in the sense that the whereabout of any D-PKG $SH_k \in \mathcal{SH}$ is well hidden from all the other nodes, including other D-PKGs, legitimate non-D-PKG nodes, and adversarial nodes. In other words, even though all the other nodes know that there exists such a D-PKG with identifier SH_k, they cannot match SH_k to any of the nodes in the network. Our method can be applied to protect distributed CAs in certificate-based schemes [2]–[6] as well.

This seemingly difficult task can be accomplished on the basis of our previously proposed MASK [10], an anonymous on-demand routing protocol designed for MANETs. MASK can nicely fulfill the routing task without disclosing the real identifiers of packet sources and destinations and all the intermediate nodes. The basic idea of MASK is to use dynamically changing node *pseudonyms* instead of their real identifiers in the routing process and packet transmissions by utilizing the aforementioned pairing technique in an intelligent way. In addition, MASK is shown to have comparable routing efficiency to that of classic ad hoc routing protocols such as AODV [13]. More details of MASK can be found in [10].

When MASK is employed to protect D-PKGs, there are two cases. If both the PKG scheme and all the other types of MANET communications desire anonymity protection, MASK can be simply used as the unique network routing protocol. Otherwise, the whole network can run on MASK when any node requests private keys, while on AODV when performing other normal network functions that do not need anonymity protection. For the latter case to work, each node of both D-PKGs and legitimate non-D-PKG nodes should have two identifiers, of which one is exclusively used with MASK and the other is used in other normal communications. In both cases, each node can still request private keys following the procedure described in Section III-B, that is, first sending requests to any *t* D-PKGs randomly chosen from $\mathcal{SH}$ and constructing the private key using the responses from them. The difference lies in that no nodes can determine where and which nodes the chosen *t* D-PKGs are, even when the D-PKGs are their neighboring nodes.

In brief, with MASK in place, adversaries cannot ascertain where the D-PKGs are or whether one node belongs to $\mathcal{SH}$ or not just based on passive eavesdropping without truly trespassing them. Although we cannot eliminate node compromise or corruption, which is believed to be impossible for any cryptographic solution, we do make it much more difficult for adversaries to carry out devastating pinpoint attacks. That is to say, since adversaries has no cleverer way to locate D-PKGs than random guessing, they have to resort to the naive *random attacks* whereby they randomly select one node and attempt to trespass it to see whether it is a D-PKG or not. In addition to their ineffectiveness of attack, such random attacks may expose adversaries themselves more easily before they breach the PKG scheme, provided that intrusion detection systems with certain capabilities are available (it requires much more capable intrusion detection systems to detect pinpoint attacks than to detect random attacks).

D. Determining Optimal Secret-Sharing Parameters

By offering D-PKGs anonymity protection, we are able to determine optimal secret-sharing parameters (t, n) for achieving the maximum security. To the best of our knowledge, no similar result has been reported in the literature.

Assume that time is divided into *time periods* and adversaries are *t-limited*, which means that they can only trespass no more than *t* nodes in each time period. This is a common assumption made about adversaries' capabilities in proactive secret-sharing schemes such as [14]. We define Pr_{comp} as the probability that adversaries happen to pick up and compromise *t* D-PKGs in one time period so as to reconstruct the system master-key, and Pr_{para} as the probability that adversaries happen to pick up (*n-t+1*) D-PKGs and corrupt them in one time period so that there are no enough *t* D-PKGs to collaboratively provide the PKG service. We then have

$$\begin{cases} \Pr_{comp} = \frac{\binom{n}{t}}{\binom{\mathcal{N}}{t}} = \prod_{i=0}^{t-1} \frac{n-i}{\mathcal{N}-i} \\ \Pr_{para} = \frac{\binom{n}{n-t+1}}{\binom{\mathcal{N}}{n-t+1}} = \prod_{j=0}^{n-t} \frac{n-j}{\mathcal{N}-j} \end{cases}, \tag{5}$$

where $\mathcal{N} = |\Psi|$ and $n = |\mathcal{SH}|$ denote the numbers of nodes and D-PKGs in the network, respectively.

In practice, both metrics are equally important and expected to be as low as possible. To reflect this fact, we then define a new metric ***Security Level*** as $SL_n(t) = 1 - 0.5 * Pr_{comp} - 0.5 * Pr_{para}$. It can be easily shown that $SL_n(t)$ is a concave function maximized at $t = \lceil n/2 \rceil$ for any given n. For the choice of n, we have shown that[5], when n is equal to either $2\left\lceil \frac{\mathcal{N}-2}{5} \right\rceil - 1$ or $2\left\lfloor \frac{\mathcal{N}+3}{5} \right\rfloor - 1$, $SL_n(\lceil n/2 \rceil)$ attains the maximum value. Based on this result, the TA is able to select an appropriate number of nodes as D-PKGs and determine the optimal secret-sharing threshold to achieve the maximum security during the network bootstrapping phase.

[5]Due to space limitations, the derivation process is omitted.

E. Discussion

Accommodation of dynamic node join/leave

Our AC-PKI allows dynamic node join/leave when the network is in operation. For example, when a new node joins the network, it can utilize its pre-equipped private key obtained from the TA to achieve mutual authentication with existing nodes and then obtain useful information from them. However, in this case, the total number of nodes in the network, i.e., $\xi = |\Psi|$, becomes a dynamically changing parameter. To achieve the highest level of security, it is necessary to adjust the size n of the shareholder set $\mathcal{SH}$ and the threshold value t based on the previous result. We have developed a pairing-based scheme to dynamically adjust the shareholder set and the secret sharing threshold so as to accommodate dynamic node join/leave in MANETs while maintaining the highest level of security. For the lack of space, we will report this result in a a separate paper.

Proactive secret sharing

We notice that if adversaries have the entire lifetime of the system master-key to mount attacks, they may reconstruct the master-key or break down the PKG scheme some time or other after compromising or corrupting enough D-PKGs. For this reason, we have also designed a suite of pairing-based proactive secret-sharing schemes to dynamically refresh secret shares of D-PKGs and to detect maliciously (or accidentally) corrupted shares, as well as to securely recover the correct shares when modification is detected. Due to space limitations, such results are left to the extension of this paper.

IV. Related work

In this section, we briefly review some of the important works mostly related to our paper. In their seminal paper [2], Zhou and Hass proposed to apply the secret-sharing technique [1] to distribute the CA's private key among a pre-selected subset of nodes, called servers. Then any combination of t servers can jointly issue public-key certificates to mobile nodes. Subsequently, several other proposals [3]–[6] were proposed to improve their scheme from different facets. These proposals all belong to the category of certificate-based PKI approaches and thus suffer from the burden of complicated certificate management, which is unfavorable in resource-constrained MANETs. In addition, none of them consider offering anonymity protection for distributed CAs. As a result, they are vulnerable to the traffic analysis attacks pointed out in Section III-C.

Another notable approach was proposed by Hubaux *et al.* in [15], in which each node acts its own CA and issues certificates to other nodes. Their approach has no trust authority like the TA in AC-PKI and is only loosely related to our work in this paper.

In addition, Khalili *et al.* [16] suggested the possible application of ID-PKC combined with secret-sharing technique in MANETs. Nevertheless, their work still remains on a conceptual level.

V. Conclusion

This paper presented our preliminary results about the applications of identity-based public-key cryptography in MANETs. Specifically, we proposed AC-PKI, a novel Anonymous and Certificateless Public-Key Infrastructure to efficiently and securely provide public-key services without using public-key certificates. To satisfy the demand of private keys during network operation, we designed a distributed private-key-generation scheme by utilizing Shamir's *(t,n)* secret-sharing technique to distribute a system master-key among a set of pre-selected nodes, called D-PKGs. In addition, D-PKGs were offered anonymity protection to defend against pinpoint attacks, which makes AC-PKI more secure than previous applications of the secret-sharing technique in MANETs. We also determined the optimal secret-sharing parameters (t, n) to achieve the maximum security and designed a novel protocol to dynamically adjust (t, n) to accommodate dynamic node join/leave. As the future research, we intend to evaluate and justify the efficacy of the proposed schemes through simulations and practical implementations.

References

[1] A. Shamir, "How to Share a Secret," *Communications of the ACM*, 22(11):612-613, 1979.

[2] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks," *IEEE Networks*, 13(6): 24-30, 1999.

[3] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad hoc networks," *IEEE ICNP'01*, Riverside, CA, Nov. 2001.

[4] S. Yi and R. Kravets, "Moca: mobile certificate authority for wireless ad hoc networks," in *PKI'03*, Gaithersburg, MD, Apr. 2003.

[5] M. Narasimha, G. Tsudik and J.H. Yi, "On the utility of distributed cryptography in p2p and manets: the case of membership control," in *IEEE ICNP'03*, Atlanta, GA, Nov. 2003.

[6] M. Bechler, H.-J. Hof, D. Kraft, F. Pahlke, and L. Wolf, "A cluster-based security architecture for ad hoc networks," in *IEEE INFOCOM'04*, Hong Kong, China, Mar. 2004.

[7] A. Shamir, "Identity based cryptosystems and signature schemes," in *Proc. CRYPTO'84*, Springer LNCS Vol. 0196, 1984.

[8] D. Boneh and M. Franklin, "Identify-based encryption from the weil pairing," in *Proc. CRYPTO'01*, pages 213-219, Springer-Verlag, 2001.

[9] P. S. L. M. Barreto, H. Y. Kim, B. Bynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *Proc. CRYPTO'02*, Springer Verlag, August 2002.

[10] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Anonymous communications in mobile ad hoc networks," in *IEEE INFOCOM'05*, Miami, FL, Mar. 2005.

[11] M. Pease, R. Shostak and L. Lamport, "Reaching agreement in the presence of faults," *Journal of the ACM*, 27:228-234, Apr. 1980.

[12] J.C. Cha and J.H. Cheon, "An identity-based signature from Gap Diffi-Hellman groups," in *PKC'03*, 2567:18-30, LNCS, Springer-Verlag, 2003.

[13] C. Perkins, E. Belding-Royer and S. Das, "Ad hoc on-demand distance vector (AODV) routing," RFC 3561, July 2003.

[14] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: how to cope with perpetual leakage," extended abstract, IBM T.J. Watson Research Center, Nov. 1995.

[15] S. Capkun, L. Buttyan and J.-P. Hubaux, "Self-organized public key management for mobile adhoc networks," *IEEE Transactions on Mobile Computing*, 2(1):52-64, Jan/Mar 2003.

[16] A. Khalili, J. Katz and W. Arbaugh, "Toward secure key distribution in truly ad-hoc networks," in *IEEE SASN'03*, Orlando, FL, Jan. 2003.