A Survey on Cryptography Applied to Secure Mobile Ad Hoc Networks and Wireless Sensor Networks

Jianmin Chen and Jie Wu

Department of Computer Science and Engineering

Florida Atlantic University

E-mail: {jchen8}@fau.edu, {jie}@cse.fau.edu}

ABSTRACT

Some security research in Mobile Ad Hoc Networks (MANETs) and Wireless Sensor Networks (WSNs) is very closely related to cryptography. There are numerous security routing protocols and key management schemes that have been designed based on cryptographic techniques, such as public key infrastructures and identity-based cryptography. In fact, some of them are fully adapted to fit the network requirements on limited resources such as storage, CPU, and power limitations. For example, one way hash functions are used to construct disposable secret keys instead of choosing private key in public key infrastructure. To gain a quick understanding of security design, we provide a survey on cryptography applications to secure MANETs and WSNs. Through this survey, we present network security schemes and protocols according to cryptographic techniques, give a few case studies on popular techniques of cryptography application, and dissect one of the designs using cryptographic techniques.

INTRODUCTION

One of the biggest challenges when it comes to securing MANETs and WSNs is all of the factors that must be accounted for: dynamic topologies, resource constraints, no infrastructure and limited physical security. As WSNs typically have more nodes than MANETs, and sensory nodes in WSNs are more resource constrained in terms of power, computational capabilities, and memory, the security design used in WSNs has to be more specific for those areas. Much research has been conducted on routing security, key management, and trust in MANETs and WSNs; most of it is associated with cryptography, authentication, authorization, encryption, and decryption. The detailed process can be found through various surveys and development of cryptography tools for MANETs/WSNs.

Several surveys have been published on attacks and countermeasures in MANETs (Wu & Chen, 2008), key management in MANETs (Wu & Cardei, 2008), security locations in WSNs (Srinivasan, 2008), secure routing protocols in MANETs (Pervaiz, 2008), challenges and solutions in wireless security (Lou, 2003), key management schemes in WSNs (Xiao, 2007), and open issues in WSNs (Evans, 2006). From the security perspective, a cryptography technique or scheme can be applied in MANETs/WSNs in different areas. For example, ID-based cryptography (Shamir, 1984) is used to develop a new certificateless security scheme in MANETs, and it is also used as a security scheme in vehicular ad hoc networks. It can also be used for secure routing applications. Evidently, knowledge about cryptography and its special customization with MANET and WSN case studies will provide the research community with the latest updates in cryptographic techniques and bring new perspective to security, performance, and many other areas of high importance in MANETs and WSNs. The focus of cryptography and its basic applications in MANETs/WSNs will build the foundation for advanced research in security. One example is a configurable library for elliptic curve cryptography in wireless sensor networks called TinyECC (Liu, 2008). Our survey is part of the effort to promote cryptographic techniques and knowledge to prepare others for research on security design in MANETs/WSNs.

This chapter aims to explain how MANET and WSN security design can be better achieved with a broad knowledge of cryptography. The cases studies in this chapter are chosen to discuss one way hash functions, threshold cryptography, public key infrastructure (PKI), identity-based cryptography, and batch verification of signatures. The chapter is organized as follows. We give an overview of cryptographic techniques commonly used to secure MANET/WSN design in section 3.1, and continue to discuss symmetric cryptographic techniques in MANETs/WSNs based on case study LHAP (Zhu & Xu, 2003) in section 3.2. We review asymmetric cryptographic techniques applied in MANET/WSN security in section 3.3, with a special focus on composite design with different cryptographic techniques based on case study IKM (Zhang, Liu, Lou & Fang, 2006). For example, threshold cryptography is used to enable secret sharing applied in different cases for various gains, from security to performance aspects. Threshold cryptography applications are discussed in section 3.4, and other cryptographic techniques based on the case study IBV scheme (Zhang, Lu, Ho & Shen, 2008) are in section 3.5. Section 4 will present open-ended questions and future challenges. Section 5 concludes the chapter.

CRYPTOGRAPHY TECHNIQUES OF SECURED MANETS/WSNS DESIGN

Security is the combination of processes, procedures, and systems used to ensure confidentiality, authentication, integrity, availability, access control, and non-repudiation.

- **Confidentiality:** The goal of confidentiality is to keep sent information from being read by unauthorized users or nodes. MANETs/WSNs use an open medium, so usually all nodes within the direct transmission range can obtain the data. One way to keep information confidential is to encrypt the data. In WSNs, confidentiality is achieved to protect information from disclosure when communication is between one sensor node and another sensor node or between the sensors and the base station. Compromised nodes may be a threat to confidentiality if the cryptographic keys are not encrypted and stored in the node.

- **Authentication:** The goal of authentication is to be able to identify a node or a user and to prevent impersonation. In wired networks and infrastructure-based wireless networks, it is possible to implement a central authority at a router, base station, or access point. However, there is no central authority in MANETs/WSNs, and it is much more difficult to authenticate an entity. Confidentiality can be achieved via encryption. Authentication can be achieved by using a message authentication code (MAC) (Menezes, Oorschot & Vanstone, 1996).
- **Integrity:** The goal of integrity is to keep a sent message from being illegally altered or destroyed during transmission. When the data is sent through the wireless medium, the data can be modified or deleted by malicious attackers. The malicious attackers can also resend it, an action known as a *replay attack*. Integrity can be achieved through hash functions.
- **Non-repudiation:** The goal of non-repudiation is related to the fact that if an entity sends a message, the entity cannot deny that it sent the message. By producing a *signature* for the message, the entity cannot later deny having sent that message. In public key cryptography, a node, A, signs the message using its private key. All other nodes can verify the signed message by using A's public key, and A cannot deny that its signature is attached to the message.
- **Availability:** The goal of availability is to keep the network service or resources available to legitimate users. It ensures the survivability of the network despite malicious incidents. In a WSN, the examples of risk of loss of availability can be sensor node capturing and denial of service attacks. One solution could be to provide alternative routes in the protocols employed by the WSN to mitigate the effect of outages.
- **Access control:** The goal of access control is to prevent unauthorized use of network services and system resources. Obviously, access control is tied to authentication attributes. In general, access control is the most commonly needed service in both network communications and individual computer systems.

Cryptography is very strongly tied to mathematics and number theory. It is, therefore, difficult to create a new design using composite cryptographic techniques without the sound security analysis behind it, usually based on cryptographic reasoning. One way to reach this goal is to learn from others by reviewing the current MANET/WSN security schemes, and also to understand the network to further understand how cryptographic techniques combine with MANETs/WSNs to provide a security service with reasonable network performance, scalability, storage, and synchronization. Certainly the security design can be evaluated using different techniques. Our goal is to provide perspective using cryptographic techniques and study basic cryptographic techniques (as seen in Figure 1) when applied to authentication, trust, and key management in MANETs/WSNs. Furthermore, we can study several of the most commonly-used cryptographic techniques and see how they are employed to deal with different tasks and balance security and performance.

It is a common approach today to use software engineering design patterns to illustrate the design of object-oriented programming. Likewise, in security and performance of MANETs/WSNs, cryptographic techniques can successfully be used in different stages of network bootstrap, packet communication, and factors to be evaluated. These techniques can certainly be reused after the analysis as known techniques from the cryptography perspective. One of the approaches we take here is to break down the design using cryptographic techniques and do some reverse engineering, then see how the new design is formed using different cryptographic techniques.

Overview of cryptographic techniques

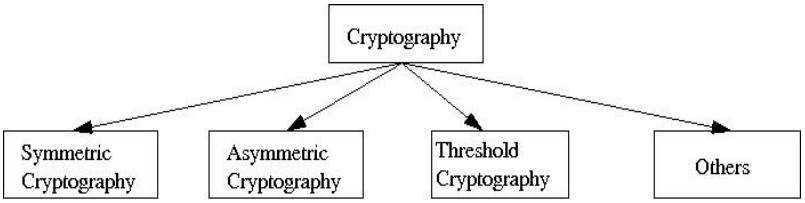
It is difficult decision to make which cryptographic techniques should be used, how often they are used, which network performance metrics are used to evaluate the design, and security analysis. The first choice may be "when does one use symmetric cryptography and when does one use asymmetric cryptography?" For example, in order to get better performance, a hash key chain can sometimes be a better choice than an asymmetric private key for encryption due to MANETs' and WSNs' dynamic changes. Specifically, alternative temporary symmetric secret keys may be better than asymmetric 1024 bit public keys.

Many researchers have proposed the use of asymmetric cryptography such as public keys using RSA (Mehuron, 94) or Elliptic Curve Cryptography (ECC) (Salomaa, 1996) to secure wireless ad hoc network routing protocols (Zhou & Haas, 1999; Yi, Naldurg, & Kravets, 2002; Zapata, 2002). But considering the ad hoc network computation cost to verify asymmetric signatures and how many times the verification has to be done, there are also several securing routing protocols (Hu, Perrig, & Johnson, 2002; Zhu & Xu, 2003) that are proposed to use symmetric keys to encrypt and authenticate, etc. One of the more commonly-used cryptographic techniques is the one-way hash function, which derives other techniques(ie, hash chain, TESLA key, Merkle hash tree and hash tree chain). The cryptographic techniques used in select MANET/WSN security research work is shown in Figure 1. The detail of the scheme's reference will be given in the following description and Table 1.

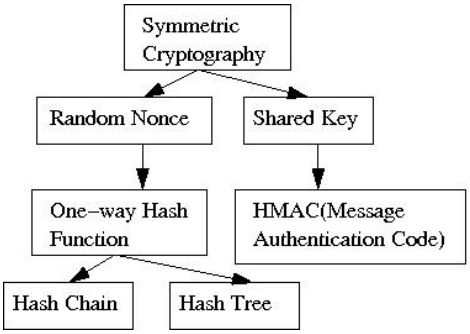
Digital signatures, hash functions, and hash functions based on a message authentication code (HMAC) (Menezes, Oorschot, & Vanstone, 1996) are techniques used for data authentication or integrity purposes in securing MANETs/WSNs. A digital signature is usually signed by a private key and can be verified by its public key. In further detail, a public key is protected by the public-key certificate, in which a trusted entity called the *certification authority* (CA) in public key infrastructure (Menezes, Oorschot, & Vanstone, 1996) vouches for the binding of the public key with the owner's identity. Those cryptographic techniques are used in most security schemes in MANET/WSN design, for example, SOLSR (Clausen, 2003), ARAN (Sanzgiri, 2002).

It is very challenging to use different cryptographic techniques to deal with different tasks. The best example is illustrated in the countermeasure resource consumption error, where the LHAP scheme has to show the art of using composite techniques.

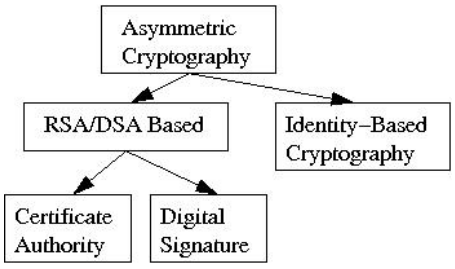
Another popular topic of discussion is determining how to build up MANETs or WSNs and how to maintain the network. For example, in order to use one-way hash chain techniques in MANETs and WSNs, the network also has to be considered when determining how to bootstrap the network, how to deliver the key chain, how to let nodes join the network, and how the nodes communicate with neighbors and countermeasure certain attacks. Other cryptographic techniques have to be considered in the design to establish trust relationships and authentication keys in MANETs in order to complement the use of techniques.



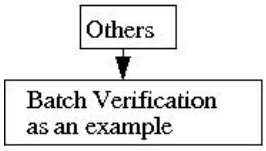
(a) Cryptography major components applied in MANETs/WSNs.



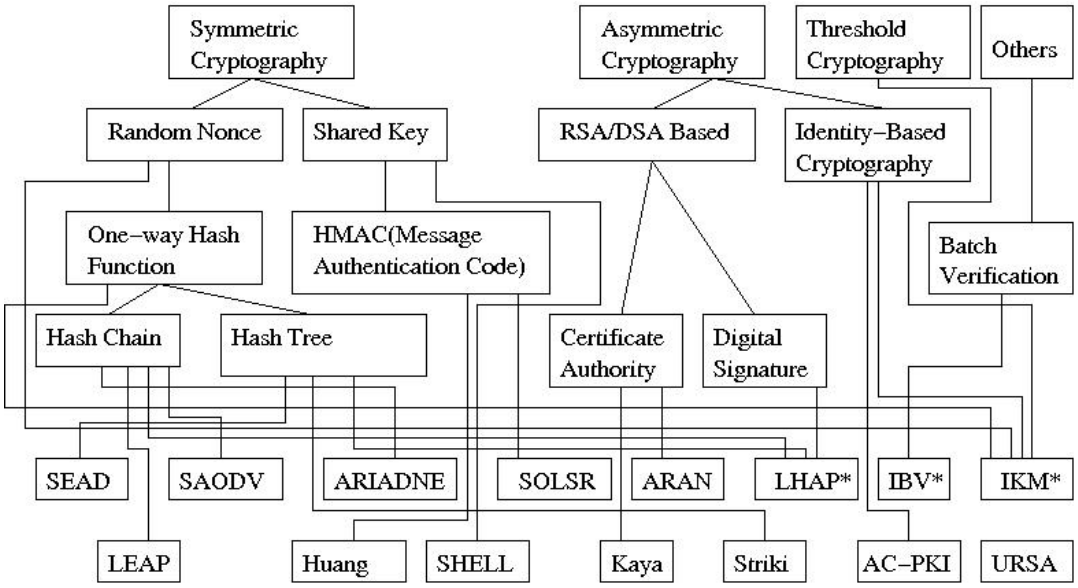
(b) Symmetric cryptography commonly-used techniques and their dependency relationships.



(c) Asymmetric cryptography commonly-used techniques in MANETs/WSNs and their dependency relationships.



(d) Others techniques in MANETs/WSNs.



*(e) Cryptography techniques used in MANETs/WSNs security schemes. Schemes with * are selected as study cases.*

Figure 1: Cryptographic techniques introduction and selected MANET/WSN security schemes applied.

Cryptographic techniques are grouped together and associated with each other to support schemes and protocols in MANET/WSN as shown in Figure 1 (a), (b), (c), (d), (e). Cryptography can be categorized into four parts seen in Figure 1 (a); In detail, symmetric key techniques are shown in Figure 1 (b), in which random nonce, shared keys, one-way hash functions, hash chains, hash trees, and message authentication codes are most-commonly-used in MANET/WSN; and as part of Figure 1(d), those symmetric techniques are used for schemes SEAD (Hu, Johnson, & Perrig, 2002), SAODV (Zapata, 2002), ARIADNE (Hu, Perrig, & Johnson, 2002), SOLSR, LEAP (Zhu, Setia, & Jajodia, 2003), Huang (Huang, Buckingham, & Han, 2005), and SHELL (Younis, Ghumman, & Eltoweissy, 2006). Secondly, asymmetric key techniques are presented in Figure 1 (c), in which public/private keys, RSAs, Digital Signature Algorithms (DSA), ID-based cryptography, certificate servers, and digital signatures are most-commonly-used techniqures in MANET/WSN; and as part of Figure 1 (d), those asymmetric techniques are associated to support schemes such as Kaya (Kaya, 2003), ARAN, LHAP, IKM, AC-PKI (Zhang, Liu, Lou, Fang, & Kwon, 2005), and Striki (Striki & Baras, 2004). Third, threshold cryptography is shown in Figure 1 (d) to support part of the IKM scheme, URSA (Luo & Lu, 2004). Last but not least, batch verification based on ID-based signature is shown in Figure 1 (d) to represent other cryptographic techniques that are not included in the discussion of the survey to support security of MANETs/WSNs. For example, for the IBV scheme, the comment is that there are many other cryptographic techniques that can be applied in MANETs/WSNs. In the following paragraph we show a collection of short reviews of cryptographic techniques and a small discussion about selected MANET/WSN security solutions.

- **Symmetric cryptography:** The encryption key is closely related to the decryption key in that they are identical in most cases. In practice, keys represent a shared secret between two or more parties that can be used to maintain private communication.
  Usually the network can choose a shared secret key to encrypt and decrypt the message once two more parties use a public/private key pair to build trust in the hand-shake stages, which is more feasible and efficient from a computational standpoint than asymmetric key techniques.
- **Random nonce:** In the network, a timestamp or random number (nonce) is used to make packets fresh and prevent a replay attack (Kaufman, Perlman, & Speciner, 2002). The session key is often generated from a random number. In the public key infrastructure, the shared secret key can be generated from a random number, too.
  Cryptographic pseudo random generators typically have a large pool of seed value. Sometimes the design and implementation of cryptographic pseudo random generators can easily become the weakest point of the system.
- **Shared key:** Symmetric key algorithms are used more often than asymmetric algorithms. They are less computationally intense; in practice, asymmetric algorithms are at least hundreds of times slower than symmetric key algorithms. The most commonly-used algorithms include AES, RC4 and IDEA. The disadvantage of shared keys in networks is that there are a total of $n(n-1)/2$ shared keys among *n* nodes in order to have a secure communication between any two nodes.

  In wireless sensor networks, some of the key management protocols are based on symmetric shared key techniques. In detail, instead of a shared key for each pair of nodes called pairwise key in a structure, other choices include one shared secret key for networks, or group key for each group or cluster of networks. Lee (2007) (Lee, Leung, Wong, Cao, & Chan, 2007) shows the

detailed discussion using case studies of five key management protocols: Eschnauer (Eschenauer & Gligor, 2002), Du (Du, 2003), LEAP, SHELL, and Panja (Panja, Madria, & Bhargava, 2006).

- **HMAC message authentication code:** It is a type of message authentication code calculated using a hash function in combination with a secret key. Usually in MANETs/WSNs, the hash functions chosen are mostly MD5 or SHA-1. It can also be used to make sure that the message sent unencrypted retains its original content by calculating the message HMAC using a secret key. For example see SOLSR, Huang (Huang, Buckingham, & Han, 2005).
- **Hash chain:** It is generated by a successive application of a hash function to a string. Lamport (Menezes, Oorschot & Vanstone, 1996) suggested the use of hash chains as a password protection scheme. Due to the one-way property of secure hash functions, it is impossible to reverse the hash function. The hash chain length is set to a limited number, and it is used as a reversed order of generation. For example, SAODV, ARIADNE, and LEAP are three applications in MANETs/WSNs that use one-way key chains.
- **Hash tree:** It was originally invented to make it possible to handle many Lamport one-time signatures. At the top of a hash tree there is a top hash or master hash. Nodes further up in the tree are the hashes of their respective children. An example can be found in the MANET/WSN security scheme SEAD.
- **Asymmetric cryptography:** It is also known as public-key cryptography. In public key cryptography, there is a pair of public/private keys. The private key is kept private, while the public key can be public to others. One of the earliest public-key cryptographic techniques, known as RSA, was developed in the 1970s. Since the 1970s, a large number of encryption, digital signature, key management, and other techniques have been developed in public-key cryptography, such as the ElGamal cryptograph system, DSA, and elliptic curve cryptography.
- **Certificate Authority:** In cryptography, a certificate authority is an entity that issues digital certificates for use by other parties. CA is the most important role in many public key infrastructure schemes.

  In MANETs/WSNs, a popular topic of debate is whether certificate authorities are practical. But it is good to take advantage of the CA role if possible even in MANETs/WSNs. Usually network nodes in MANETs trust the CA in the stage of bootstrap and can verify the CA's signature. Then, nodes can also verify whether a certain public key does indeed belong to another node, as it is identified in the certificate. For example, ARAN and Kaya (Kaya, 2003) are two applications in MANETs/WSNs that use certificate authority.
- **Digital signature based on RSA/DSA:** The ElGamal signature is based on the difficulty in breaking the discrete log problem. DSA is an updated version of the ElGamal digital signature scheme published in 1994 by FIPS, and was chosen as the digital signature standard (DSS) (Mehuron, 94).

  Digital signature using the RSA/DSA algorithm is popular for authentication or confirming the message's integrity. A digital signature scheme typically consists of three algorithms: a key generation algorithm, a signing algorithm, and a signature verifying algorithm.

  In MANETs/WSNs, the digital signature is more expensive to compute compared to the hash function, and it is not scalable with a lot of nodes in MANETs/WSNs. For example, digital signature is only performed once in bootstrapping a TESLA key chain in LHAP scheme.

- **Identity-based cryptography:** It is a type of public-key cryptography, and the first identity-based cryptography was developed by Adi Shamir in 1984, which uses the identity of the user as a public key. Modern schemes include Boneh/Franklin's pairing-based encryption scheme (Boneh & Franklin, 2001). For example, IKM and AC-PKI schemes are applications that use ID-based cryptography.
- **Batch verification with ID-based signature:** Although there are advantages of ID-based cryptography signature schemes based on pairing, the signature verifications are at least ten times slower than that of DSA or RSA; the batch verification (Yoon, Cheon, & Kim, 2004) of many signatures increases the efficiency.

Table 1 lists some of the security schemes that use cryptographic techniques, security objectives, and cryptographic techniques.

Table 1: Overview of cryptographic techniques used in security schemes in MANETs/WSNs.

MANET/WSN Security Scheme	Security Objectives	Cryptographic Techniques
ARAN	Authentication, integrity, and non-repudiation of signaling packets, based on AODV (Perkins, 2001), designed to substitute reactive routing protocols.	Certificate authority, timestamp.
ARIADNE	Authentication and integrity of signal packets, based on the basic operations of DSR (Perkins, 2001).	Symmetric cryptography primitives, hash function and timestamp.
SAODV	Authentication and integrity of signaling packets, a security extension for AODV.	Digital signature and hash chain.
SEAD	Authentication and integrity of signaling packets, based on DSDV (Perkins, 2001), applied to other distance vector protocols.	Hash chain and sequence number.
Huang	A secure level key infrastructure for multicast to protect data confidentiality via hop-by-hop reencryption and mitigate DoS-based flooding attacks through an intrusion detection and deletion mechanism. The multicast protocol divides a group routing tree into levels and branches in a clustered manner.	MACs and one way sequence number, cluster-based tree as key management.
Kaya	A dynamic multicast group management protocol is proposed which aims to equally distribute the workload of securing communication to all participating members through MANETs.	Certificate authority and ad hoc group shared key.
LEAP	Source and message one way key chain based authentication and cluster-based shared key in key management to countermeasure wormhole, sinkhole, Sybil, DoS, replay, insider attacks.	Hash chain and cluster-based shared key.
SLSP (Papadimitratos & Haas, 2003)	Authentication, integrity, and non-repudiation of signal packets, extends an intrazone protocol for ZRP (Perkins, 2001).	Certificate authority.

SPAAR (Carter & Yasinsac, 2002)	Authentication, integrity, non-repudiation, and confidentiality, secure position aided ad hoc routing protocol.	Certificate authority and timestamp.
SOLSR	Authentication and integrity of signaling packets.	MACs and timestamp.
SHELL	A cluster-based key management scheme. Each cluster has its own distributed key management entity residing in a-cluster-head node. Therefore, the operational responsibility and key management responsibility are separated, which offers better resiliency against node capture.	Group shared key.
LHAP	A hop-by-hop authentication protocol for ad hoc networks.	Digital signature and hash chain.
IKM	Key management to secure mobile ad hoc network, efficient network-wide key update via a single broadcast message.	ID-based cryptography and threshold cryptography.
Striki	User authentication and Merkle tree-based data authentication in MANETs.	Hash function and hash tree.
IBV	An efficient batch signature verification scheme for vehicular sensor networks.	Batch verification of ID-based signature.

In general, most surveys have been done on security routing and other specific areas such as key management. The difference of our approach is that we take the approach from the technique used, outside of the network area. Right now we prefer to choose cases that include the latest research in the area, putting different cryptographic techniques under review. The following discussion will focus on cryptographic techniques. Using Figure 1 as the outline, we will go through the discussion from symmetric key techniques to asymmetric key techniques, and from RSA/DSA-based schemes to ID-based cryptography, with some discussion about threshold cryptography. Most of the discussion focuses on three cases in MANET/WSN security research; the LHAP scheme, the IKM scheme, and the IBV scheme.

Symmetric key techniques applied in MANETs/WSNs

From Figure 1, symmetric key techniques are used in most security MANETs/WSNs schemes, detail seen in Figure 1 (e); those techniques are random nonce, shared key, one way hash function, message authentication code, hash chain, and hash tree, seen Figure 1 (b). They are used so often that we should pay more attention to what the network factors are before they are used in our design.

One way hash chain and TESLA key (Perrig, Canetti, Tygar, & Song, 2000) are faster than the traditional PKI private key calculation; they are used in the design of several security protocols including SAODV, ARIADNE, and LHAP as shown in Figure 1. We will now discuss one-way hash chains. Hash functions can be very easy to compute compared to public key distribution, which typically requires central authentication. Thus in the network field, in order to achieve the best performance, we sometimes use hash functions instead of PKI public keys.

Lamport used one-way hash chains for password authentication. In this way, a one-way hash chain repeatedly applies a one-way hash function starting from a random number. The user picks up the secret key, which is usually a random number. Supposing that the chain length is N, the user runs the hash function N times on the random number. Actually, each hash function value is the key on the chain. In the list of keys, the original random number is the most important key, because all other secret keys can be calculated via hash function from the key. To describe using mathematical formulas, if a node wants to generate a key chain of size N, it first needs to choose a key, denoted as *seedKey*, which will be the last one used to do the encryption. The one-way hash chains are generated as follows: $K(0) = seedKey, K(1) = h(K(0),..., K(N) = h(K(N-1))$, in which h is the one-way hash function. It is infeasible to compute inversely from a one-way hash function. In various standards and applications, the two most-commonly used hash functions are MD5 and SHA-1.

Two commonly-used cryptographic techniques that are used in WSN broadcast authentication are *μTESLA* (Perrig, Szewczyk, Wen, Culler, & Tygar, 2001) and digital signatures. *μTESLA* is considered to be a symmetric cryptography technique; *μTESLA* and its variations implement broadcast authentication through delayed disclosure of authentication keys. *μTESLA* keys are based on a symmetric cryptographic hash function, and the operations cost is more efficient even though the network has to be loosely time synchronized and suffers from authentication delays. If digital signatures such as ECDSA (IEEE, 2006) are used directly for broadcast authentication, which is easily attacked by broadcasting forged packets, the receiving nodes would be forced to perform a large amount of unnecessary signature verification.

To countermeasure DoS attacks against digital signature verification in the case where the digital signature is directly used for broadcast authentication without further protection in WSNs, hop-by-hop pre-authentication filters can be used to remove bogus messages before verifying the actual digital signatures. In particular, two filtering techniques, a group-based filter and a key chain-based filter (Dong, Liu, & Ning, 2008), are based on a symmetric cryptographic hash function, hash chain, shared pairwise key, and MAC. With a group key among a sender and its neighbor nodes, an adversary cannot forge messages without compromising the group key. However, a compromised sensor leaks the group key. Alternatively, a sensor node can add a MAC to a broadcast message for each of its neighbor nodes. However, this incurs large communication overhead. Based on the above two simple methods to filter out forged messages, the group-based filter technique has to trade-off communication efficiency with security. Specifically, the group-based filter organizes the neighbor nodes of a sender into multiple groups, which are protected by different keys in a tree structure. In the second filter technique, the key chain-based filter is designed to apply a two-layer filter to deal with the DoS attacks on the verification of signatures and chained keys. On the other hand, one-way key chains feature a simple pre-authentication filter used by LHAP which cannot countermeasure the DoS attack in which an adversary may claim a key close to the end of key chain and cause a large amount of unnecessary hash operations. In terms of two layers in the two-layer filter, the first layer employs a one-way key chain to filter out fake signatures, and the second layer uses existing pairwise keys to prevent a node from conducting unnecessary hash operations.

Key management is a challenging issue in WSNs due to the sensor node's resources constraints. Various key management schemes in WSNs still are based on symmetric key techniques. To different degrees in sharing keys, the key distribution scheme models are generally network keying, pairwise keying, and group keying. For example, in Lee (Lee, Leung, Wong, Cao, & Chan, 2007) the security and operational requirements of WSNs are examined and five key management protocols - Eschenauer, Du,

LEAP, SHELL, and Panja- are reviewed. The shared keying models for WSNs are used to compare the different relationships between the security and operation requirements for WSNs: accessibility, flexibility, and scalability.

Like security, key management in WSNs is comprised of a cross-layered design, which can go from the link layer to the application layer. As an applicable link layer standard in a WSN, IEEE 802.15.4 considers key usage for secure data transmission, but it does not specify how to securely exchange keys. This opens the door to the key management problem that has been the focus of recent research. We like to sum up the benefits and problems for three models - network keying, pairwise keying and group keying. In the first place, a network model is chosen to allow the entire network to use one shared secret key; therefore, the benefits are simple to implement and allow data aggregation and fusion, are easy to scale, able to self-organize, and are flexible and accessible, whereas compromising one node compromises the entire network, which shows lack of robustness. In the second place, a pairwise model is chosen to allow each specific pair of nodes to share a different key; hence, the pairwise model has benefits of best robustness and each node is authenticated, but the pairwise model suffers from scalability problems in storage, energy and computation areas. In addition, the pairwise model is unable to self-organize, and is not flexible for addition or removal of nodes. Last but not least, the group model is designed to let each group use a different shared key; therefore, the benefits are to allow multicast and group collaboration, better robustness than network-wide keying, and adjustable scalability with the ability to self-organize within the cluster. On the other hand, the group model lacks efficient storage methods for group keying in *IEEE 802.15.4*, is difficult to securely set up, and cluster formation information is application-dependent.

So far, we had some discussion about one way hash chains, *μTESLA* key, pre-authentication filters on broadcast authentication in MANETs/WSNs, and shared key models in WSNs. Indeed, using symmetric cryptography in networks is a state-of-art advancement. Therefore, we like to use the case study of the LHAP protocol to enhance the symmetric cryptography discussion section.

Case study 1: LHAP protocol

In Figure 1, the three cryptographic techniques that are used in the LHAP protocol are shown as hash chain, hash tree, and digital signature. Taking LHAP as our first case study, we like to show the advantage of using symmetric cryptographic techniques to handle special network situations in security.

To countermeasure resource consumption attacks, one of the mechanisms employs authentication and ensures that only authorized nodes can inject traffic into MANETs. As a hop-by-hop authentication protocol for MANETs, LHAP resides in between the network layer and the data link layer, hence providing a layer of protection that can counter many attacks, including outsider attacks and insider impersonation attacks.

Multiple security schemes take advantage of the better aspect of hash chains. To illustrate, in Table 2, we present the one way hash chain techniques used in different MANET schemes. To incur small

performance overhead and a tradeoff between security and performance, various cryptographic techniques are customized for different network services in the LHAP scheme, as illustrated in Table 3.

Table 2: One way hash chain techniques in variety MANETs schemes

Secure Routing Protocol	**Cryptography Techniques**	**Network Service Provided**
SEAD	One way hash chain	Used on a hop-by-hop basis due to the basic operation of DSV
ARIANDE	TESLA key	Applied to secure on-demand routing protocols in source-to-destination nature.
LHAP	One way hash chain	Used for traffic packet authentication.
	Merkle hash tree chain	Used to achieve fast hash verification.

Table 3: LHAP scheme variety cryptographic techniques customized for different network service

LHAP Cryptographic Techniques	**Network Rationale**
1024-bit RSA digital signature	The most expensive operation in LHAP, but it is only performed once in bootstrapping a TESLA key chain. Therefore the cost is negligible when amortized over the entire packet.
TESLA key	Used to reduce the number of public key operations for maintaining trust between nodes, and also for maintaining trust between nodes.
One way hash chain (It is more efficient compared with HMAC over the message.)	Used to authenticate traffic packets for mainly two reasons: 1: One hash time cost is small compared to the overall end-to-end transmission latency of a packet. 2: Limit network memory used for buffering the received packets, and only authenticate traffic packets to its immediate neighbors to prevent an attacker. to launch replay attacks.
Merkle hash tree	Used to support fast hash verification; the maximum number of verifications a receiver has to perform is $O(\log(N))$, where N is the length of a TESLA key chain. The verification process only works for TESLA key chains.

In the LHAP protocol, in order to counter the resource consumption attack, the protocol is designed to use authentication of traffic packets to avoid bogus packets. Based on the wireless ad hoc network analysis, in cases such as network deployment, when nodes join the network, when a node gains trust from other nodes in the network, trust management may be used based on one way hash traffic key chains and a TESLA key chain. To minimize the overhead for use, the node uses an RSA digital signature only for gaining trust while traffic packet authentication is used when keys are generated from the one-way hash chain, which is generated by a hash function. Also in order to do fast hash verification, LHAP uses a tree-based authentication scheme, the Merkle hash tree.

Through the short case study of LHAP, we show that symmetric cryptography can be used creatively in special cases, that it is scalable, and also that it can be used for similarity comparisons among different

schemes that also use symmetric cryptography. Therefore, it leads us to the fact that cryptography technique studies really do help us to organize security design schemes better.

Asymmetric cryptographic techniques applied in MANET/WSN security

From Figure 1, we show that asymmetric cryptography is popularly used in the security of MANET/WSN schemes, detail seen in Figure 1 (e). Most public key infrastructure schemes are either based on RSA/DSA or ID-based cryptography, seen in Figure 1 (c); for example, the most popular scheme, ARAN, has been discussed in many surveys (Lou, 2003; Wu, Cardei, & Wu, 2008; Xiao, 2007).

Public key infrastructure in MANETs is a very popular choice when it comes to securing the network. Schemes (Luo & Lu, 2004; Yi, Naldurg, & Kravets, 2002) use a public-key infrastructure to associate public keys with the node's identity. One of PKI's approaches is to pre-load each node with all other nodes's public key certificates prior to network deployment. The above approach has two problems: scalability with network size and public key update if needed. Another approach is to use on-demand certificate retrieval, but it is not good choice considering communication latency and overhead. Secure routing protocols, such as ARAN, ARIADNE, SEAD, and SPINS (Perrig, Szewczyk, Wen, Culler, & Tygar, 2001), all are based on the assumption that there is pre-existence and pre-sharing of secret and/or public keys for all the nodes in the network. This leaves ad hoc key management and key distribution as an open problem that must be dealt with.

By deploying identity-based cryptography (IBC) and threshold secret sharing and then taking away the assumption of a pre-fixed trust relationship between nodes, several IBC-based certificate-less public-key management schemes for MANETs have been developed, such as (Deng, Mukherjee, & Agrawal, 2004; Khalili, Katz, & Arbaugh, 2003; Saxena, Tsudik, & Yi, 2004; Zhang, Liu, Lou & Fang, 2006; Zhang, Liu, Lou, Fang, & Kwon, 2005). The basic idea is to let some or all network nodes share a network master-key; some of them (Saxena, Tsudik, & Yi, 2004; Zhang, Liu, Lou, Fang, & Kwon, 2005) use threshold cryptography and collaboratively issue ID-based private keys. The PKI digital signature scheme is widely recognized as the most effective approach for Vehicular Sensor Networks (VSNs) to achieve authentication, integrity, and validity. In order to avoid scalability problems, the efficient identity-based batch verification scheme (Yoon, Cheon, & Kim, 2004) is proposed, which uses another cryptography technique - *batch verification* - based on IBC (Camenisch, Hohenberger, & Pedersen, 2007). The scheme uses IBC to generate private keys for pseudo identities, so PKI certificates are not needed and thus transmission overhead is significantly reduced.

Identity-based cryptography introduction

In 1984, Shamir proposed the idea behind identity-based encryption. However, there was no workable method to solve the problem known at the time until 2001, when Boneh (2001) invented a practical scheme based on elliptic curves and a mathematical construct called the Weil Pairing.

A bi-linear map is special mathematical function that makes IBE work. A bi-linear map is a pairing that has the property: $Pair(a*X, b*Y) = Pair(b*X, a*Y)$.

For identity-based encryption, the operator "*" is used for multiplication of integers with points on elliptic curves. The multiplications, for example $a*X$, are easy to calculate, but the inverse operations, such as finding parameter *a* from *X* and value of $a*X$, are practically impossible. The function is one way and practically non-invertible. The concept is actually the same as one-way hash functions; the bi-linear map can be a Weil Pairing.

The pairing technique can be outlined as follows with more detail by using a concrete example.

Let *p*, *q* be two large primes and E/F_p indicate an elliptic curve $y^2 = x^3 + ax + b$ over the finite field F_p. G_1 is a *q-order* subgroup of the additive group of points of E/F_p, and G_2 is a *q-order* subgroup of the multiplicative group of the finite field $F_{p^2}^*$. The discrete logarithm problem is required to be hard in both G_1 and G_2, which means that it is computationally infeasible to extract the integer *x*, given $p, q \in G_1$ such that $q = xp$. For example, a pairing is a map $\psi : G_1 \times G_1 \to G_2$ with the following properties:

- Bilinear property:

For $\forall P, Q, R, S \in G_1$, $\psi(\mathrm{P}+\mathrm{Q}, \mathrm{R}+\mathrm{S}) = \psi(\mathrm{P}, \mathrm{R})\psi(\mathrm{P}, \mathrm{S})\psi(\mathrm{Q}, \mathrm{R})\psi(\mathrm{Q}, \mathrm{S})$. And also, for $\forall a, b \in Z_q^* = \{a | 1 \le a \le q-1\}$, there is $\psi(aP, bQ) = \psi(aP, Q)^b = \psi(P, bQ)^a = \psi(P, Q)^{ab}$, etc.

- Non-degenerate property: If *P* is a generator of G_1, then $\psi(P, P) \in F_{p^2}^*$ is a generator of G_2.
- Computable property: There is an efficient algorithm to compute $\psi(P, Q)$ for all $P, Q \in G_1$.

A more comprehensive description of how these pairing techniques work can be found in papers (Barreto, Kim, Bynn, & Scott, 2002; Boneh & Franklin, 2001; Boneh, Franklin, 2003).

In our case study, we choose a hybrid cryptography scheme combining threshold cryptography with ID-based cryptography as a certificateless key scheme IKM.

Case study 2: ID-based key management scheme – IKM

As seen in Figure 1(e), several cryptographic techniques (including random nonce, one way hash function, threshold cryptography and ID-based cryptography) are shown to be used in IKM scheme. We spent some time studying the detail of network initialization in the IKM scheme, and did some reverse engineering, breaking down the initialization design using cryptographic techniques and presenting the design using a comprehensive tree-structure. Figure 2 makes it easy for readers to understand the complicated design of the network initialization, based on prototype of most commonly-used case, one random nonce, one node specific identity, and one hash function to apply node's identity, IKM scheme are designed with extended

feature, two random nonce, two set of identities, two hash functions, and many (up to maximum M) phases. The approach taken here is also a trial to use the most recent scheme in IBC, through which we like to encourage readers to take the "cryptographic techniques used" exercise, ask a series of questions regarding how many cryptographic techniques are used, when to use them, and why to use them. This break-down approach will also be a training exercise to encourage us to design a security scheme using the cryptographic techniques used in practical approaches in MANETs/WSNs.

Comparing several IBC-based certificate-less public-key management schemes (Deng, Mukherjee, & Agrawal, 2004; Khalili, Katz, & Arbaugh, 2003; Saxena, Tsudik, & Yi, 2004; Zhang, Liu, Lou, Fang, & Kwon, 2005), IKM solved several issues related to the previous IBC-based key management scheme:

- The security of the whole network is compromised when a threshold number of network nodes who share the network's master key are compromised.
- Significant communication overhead in a large-scale MANET occurs while updating ID-based public/private keys because each node has to contact a threshold number of nodes who share the network master key one by one.
- There is no quantitative argument to prove the advantage of IBC-based public key management schemes over certificate-based cryptography.

IKM contributes to several areas, one of which is to provide a novel construction method of ID-based public/private keys. In IKM, each node's public key and private key includes two parts: one is a node specific ID-based element, and the other one is a network-wide common element. The node specific ID-based elements are designed to ensure that the compromises of an arbitrary number of nodes don't affect the secrecy of non-compromised nodes' private keys. With network-wide common key elements, a single broadcast message can update the network-wide public/private keys.

In IKM, each node has an authentic ID-based public/private key pair and uses the key pair as proof of its group membership. Those key pairs help to implement the mutual authentication, key management, public-key encryption, and digital signatures. As a key management scheme, IKM consists of three phases: key pre-distribution, revocation, and update.

In first phase, key pre-distribution is done once the network has initialized where a Private Key Generator (PKG), acting as a trust authority, prepares a set of system parameters and pre-loads every node with certain key contents. After that, the PKG distributes its functionality to n distributed authorities which are selected from the overall number of nodes N to enable secure key revocation and update during network operation. The n distributed authorities in IKM are called D-PKGs for convenience.

If a node is compromised, its public key may be explicitly revoked. During network operation, if a node suspects a peer, say A, the node can send a signed accusation against A to some D-PKGs. When the number of accusations against node A reaches a predefined revocation threshold, denoted by γ, in a certain window, the node A is diagnosed as compromised. The D-PKGs can jointly issue a key revocation against A.

As a common practice, public/private keys of mobile nodes are updated at intervals for various reasons, such as preventing cryptanalysis. IKM also takes the approach and the non-revoked node can update its public key autonomously and its private key via a single broadcast message.

IKM is designed to make the distributed authorities D-PKG's indistinguishable from common nodes via anonymous routing (Zhang, Liu, Lou, Fang, & Kwon, 2005). Because of the shared wireless medium, D-PKGs IDs leak in routing and data packets, these make D-PKGs vulnerable to pinpoint attacks.

We will focus on the basics of IKM related to network initialization, key revocation, key update, and its security analysis, and go through the discussion of threshold cryptography and see how the IKM scheme can benefit from previous threshold cryptography analysis.

Network Initialization

In Table 5, the cryptography techniques used in IKM scheme are reviewed and put together to show the detail IKM scheme functionality of cryptography techniques. And also in Figure 2, in the stage of IKM scheme network initialization, the main ideas are using threshold cryptography update of network key each network phase based on hash function, and node specific public/private key based on ID-based cryptography.

Table 5. The cryptography techniques and their functionalities in IKM scheme

Cryptography techniques	**Design consideration – IKM scheme functionality.**
Random nonce	Used as network master secret keys K_{p_1}, K_{p_2}, in which one constructs a node private key, another constructs a series of network phase private keys.
Hash function	One hash function is used to make a series of network phases. In detail, $salt_i = h(salt_{i-1} + 1)(1 < i \le M)$, *h* is a hash function, such as SHA-1.
	Hash function is chosen for ID-based identity application, H_1, which maps arbitrary string to non-zero element in the subgroup G_1.
ID-based cryptography	Node specific element is related to nodes which can join network anytime, and its major concern is to define its public key and private key. For example, node A with identity ID_A has the keys: $< \Gamma_A, \Gamma_A^{-1} >= (H_1(ID_A), K_{p_1} H_1(ID_A))$.
	Network phase specific element is related to phases in different time period, and its public/private key pair is $(H_1(salt_i), K_{p_2} H_1(salt_i))$.
Threshold cryptography	It is used to apply relatively frequent key update to enhance the security. The IKM is composed of a number of continuous, nonoverlapping *key update phases*, denoted by p_i for $(1 < i \le M)$, where *M* is the maximum possible phase index.

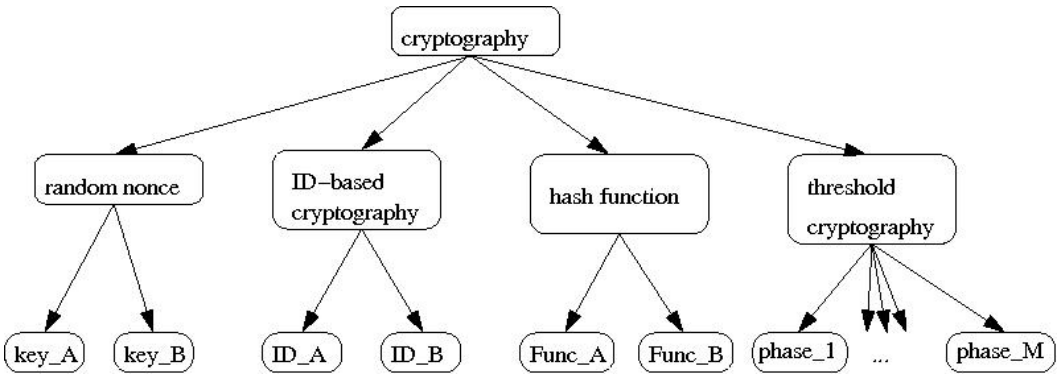


Figure 2: IKM scheme design network initialization demystified – A threshold cryptography and identity-based cryptography composite design tree structure illustration of parameters. Key_A is network master key for all nodes in network. Key_B is network master key for all network phases. ID_A is node specific identity. ID_B is network phase identity. Func_A is a hash function applied in node's identity in network. Func_B is a hash function applied in phases to generate salts. Phase_1 is network phase salt in first phase in the process of relatively frequent key update. Phase_M is network phase salt in Mth which is maximum phase index.

The PKG does the following three steps to bootstrap the network. First, generate the pairing parameters (p,q,ψ). Select an arbitrary generator W of G_1. Secondly, choose a hash function H_1 that maps arbitrary binary strings to nonzero elements in G_1. Thirdly, choose two distinct random number $K_{p_1}, K_{p_2} \in Z_q^*$ as network master-secrets. Set $W_{p_1} = K_{p_1} W$ and $W_{p_2} = K_{p_2} W$ respectively.

Preload parameters $(p,q,\psi,H_1,W,W_{p_1},W_{p_2})$ to each node; those parameters are public, while network master keys K_{p_1}, K_{p_2} should never be disclosed to any node.

In IKM secret sharing design, only knowledge of K_{p_2} is introduced into the network, and the PKG performs a *(t, n)-threshold* secret sharing of K_{p_2} to avoid the single point of compromise and failure. The random polynomial of threshold cryptograph, $g(x) = K_{p_2} + \sum_{i=1}^{t-1} g_i x^i \pmod q$. Distributed authorities D-PKGs are randomly selected from a subset of size *n* of nodes $(t \le n < N)$. After that, the PKG assigns to each node in D-PKG a secret share computed as $K_{p_2}^V = g(ID_v)$. The design is based on Lagrange interpolation. The PKG's master secret key K_{p_2} can get from value $g(0)$. However, any subset with size $(t-1)$ or smaller cannot reconstruct $g(0)$. The PKG also calculates a set of values to enable verifiable secret sharing, the values $\{W_{p_2}^V = K_{p_2}^V W | V \in \Omega\}$ preloaded to each D-PKG, in which Ω is the D-PKG set. Due to the difficulty of solving the DLP in G_1, none of the other D-PKGs can know the secret share $K_{p_2}^V$ of D-PGK *V* from $W_{p_2}^V$. To make key revocation and update feasible, the IDs of all the D-PKGs are public to each node.

IKM is designed to construct ID-based public/private keys for each node *A*. The IKM contains a number of continuous, non-overlapping key update phases, in which i^{th} key update period is denoted by p_i for $(1 < i \le M)$, where M is the maximum possible phase index. Each phase p_i is associated with a unique binary string called a phase salt, denoted by $salt_i$. The PKG issues a random number $salt_1$ to each node before the deployment of network, and with a hash function *h* such as SHA-1, a series of $salt_i$ is generated using $salt_i = h(salt_{i-1} + 1)(1 < i \le M)$.

There are both node-specific and phase-specific public/private key pairs, and node *A*'s key pair is valid only during phase p_i is denoted by $< \Gamma_{A,p_i}, \Gamma_{A,p_i}^{-1} >$. Each public key Γ_{A,p_i} and private key Γ_{A,p_i}^{-1} is compromised of a node-specific element and a phase specific element common to all the nodes, both in G_1. $\Gamma_{A,p_i} := (\Gamma_A, \Gamma_{p_i}) = (H_1(ID_A), H_1(salt_i))$, $\Gamma_{A,p_i}^{-1} := (\Gamma_A^{-1}, \Gamma_{p_i}^{-1}) = (K_{p_1} H_1(ID_A), K_{p_2} H_1(salt_i))$.

At the beginning, the PKG issues $< \Gamma_{A,p_1}, \Gamma_{A,p_1}^{-1} >$ to node A from which $< \Gamma_{A,p_i}, \Gamma_{A,p_i}^{-1} > (1 < i \le M)$ is originated from the D-PKGs during network operation. $< \Gamma_{p_i}, \Gamma_{p_i}^{-1} >$ is common public-key and private-key element of phase p_i, and $< \Gamma_A, \Gamma_A^{-1} >$ as node-specific public-key and private-key elements of node A. The phase p_i public/private key pair changes across key-update phases, while node A public/private key pairs remain the same during network lifetime and should not be released to node A itself.

Because it is difficult to solve the discrete logarithm problem in the subgroup G_1, it is not possible to figure out the network master secret K_{p_1} and K_{p_2} from an arbitrary number of public/private key pairs. Therefore, IKM has a property which allows it to keep the confidentiality of the node's private key if the node is compromised regardless of how many key pairs adversaries are able to acquire from compromised nodes. The IKM scheme has more resilience to the compromise of D-PKGs than the conventional key construction method (Saxena, Tsudik, & Yi, 2004; Zhang, Liu, Lou, Fang, & Kwon, 2005).

The IKM scheme allows dynamic node joins at any time. Suppose a new node X joins the network at phase p_i, the PKG just needs to pre-equip X with public system parameters and $< \Gamma_{A,p_i}, \Gamma_{A,p_i}^{-1} >$. Based on the support of a node joining the network at any time, the IKM scheme network size can grow without limitation, therefore high network scalability is achieved.

Key Revocation

The IKM scheme includes key revocation design has three subprocesses: misbehavior notification, revocation generation, and revocation verification. For our case study, we only show the misbehavior notification; the other two parts refer to the original paper (Zhang, Liu, Lou & Fang, 2006).

Suppose node *B* detects node A's misbehavior. Node B can generate signed accusation $[ID_A, s_B]_{\Gamma_{B,p_i}^{-1}}$ against A, where s_B is a timestamp to countermeasure message replay attacks. If node B likes to send the revocation message to the D-PKGs, several things have to be considered. It is not wise

for node *B* to naively flood the accusation because it is not secure, considering that node A may temporarily behave normally. Node A attempts to lower the number of accusations against it down to the level that is below the predefined revocation threshold γ. Therefore, the IKM scheme takes the approach to let node B unicast the accusation secretly to one of the D-PKGs instead.

During network initialization, the PKG provides each node with a function η that maps each node ID to the IDs of β distinct D-PKGs. For any node A in the network node set, denoted by Λ, $\eta(ID_A) = \{ID_{X_j} \mid 1 \le j \le \beta, X_j \in \Omega, X_j \ne A\}$. The approach IKM used to construct the function divides the node set Λ into n disjoint node sets, each associated with β distributed authorities D-PKGs.

The β is the part to determine the tradeoff between resilience to D-PKG compromise and communication overhead. Smaller β leads to a lower related communication overhead, and also a less resilient network to the compromise of D-PKGs.

Key Update

It is common practice to update keys to countermeasure the cryptanalysis and limit any potential damage from compromised keys. Previous research in MANETs and WSNs show the related work to update keys using threshold cryptography, for example, (Zhou & Haas, 1999; Luo & Lu, 2004). It is in our interest to show the threshold cryptography applied in different cases, determine the main factors to be evaluated, and how to become a pro when it is challenged to be put into the design. So in our case study, we like to show the detail of the key update of IKM and get more details about the network analysis.

A new key update phase p_{i+1} starts either because the previous phase p_i times out comparing with a predetermined time threshold, or because the number of nodes revoked in p_i is not less than the prescribed threshold. In the IKM scheme, each node can update its public/private key autonomously. For example, node *B* uses the following formula: public key case, $\Gamma_{B,p_{i+1}} := (H_1(ID_B), H_1(salt_{i+1}))$, where $salt_i = h(salt_{i-1} + 1)(1 < i \le M)$. From the computation overhead standpoint, there are only two hash operations for node *B* to operate in order to update its public key; private key case, $\Gamma^{-1}_{p_{i+1}} = K_{p_2} H_1(salt_{i+1})$, private key update needs the work from *t* D-PKGs in Ω. In the IKM scheme, the simple way taken is to assume that $Z \in \Omega$ initiates phase p_{i+1}, but the D-PKGs should take turns to balance their resource usage. *Z* randomly selects $(t-1)$ other non-revoked distributed authorities D-PKGs from Ω and sends a request to each one.

The key update method of the IKM design involves the consideration of the network's self healing capabilities, for the scenario in that any non-revoked node can recover $\Gamma^{-1}_{p_j}$ for any phase $p_j (j > i)$ if the node did not receive the key-update broadcast message due to MANET mobility, channel errors, and temporary network partitions.

Security Analysis On Threshold Cryptography Used

The IKM scheme is designed and provides more security than other MANET security schemes using certificate key management CKM (Yi & Kravets, 2003; Zhou & Haas, 1999;) and previous identity based cryptograph IBC-based schemes (Saxena, Tsudik, & Yi, 2004; Zhang, Liu, Lou, Fang, Kwon, 2005 (referred to as o-IKM)).

All these approaches are *(t,n)-threshold* schemes, so they have the same level of security as long as the *t-limited* assumption holds. The difference is the worst-case scenario. Table 6 shows the detail, IKM part of threshold scheme is as secure as conventional certificate based key management CKM's, but it outperforms o-IKM in the worst-case scenario.

Table 6: Threshold cryptography worst case comparison: compromised nodes reaches threshold.

(t,n)-threshold scheme distributed CAs are compromised	**CKM**	**IKM**	**o-IKM**
Can adversaries construct a secret key? If yes, what key is it?	Yes CA's private key.	Yes One of PKG master secret key.	Yes Same as IKM.
Can adversaries deduce the private key of any non-compromised node?	No	No	Yes
Is overall system security lost?	No	No	Yes

After the case study, we will now cover a discussion on threshold cryptography (Shamir, 1979; Desmedt & Frankel, 1989) applied in MANETs and WSNs.

Threshold cryptography applied in MANET/WSN security

From Figure 1, the threshold cryptography is shown as a technique used by IKM, URSA schemes in MANET/WSN. Actually the cryptography is widely used in a variety of schemes (Capkun, Buttyan, & Hubaux, 2003; Gouda & Jung, 2004; Kong, Zerfos, Luo, Lu & Zhang, 2001; Luo & Lu, 2004; Saxena, Tsudik, & Yi, 2004; Yi & Kravets, 2003; Zhou & Haas, 1999; Zhang, Liu, Lou & Fang, 2006). We like to compare those schemes and discuss the most frequently asked questions when the threshold cryptography is applied in MANETs/WSNs.

For detailed background knowledge of threshold cryptography, please refer to the paper (Shamir, 1979; Desmedt & Frankel, 1989); due to limited space, we don't present the threshold cryptography primitives here. There are several features mostly discussed in research literature, namely verifiable secret sharing (Chor, Goldwasser, Micali, & Awerbuch, 1985; Gennaro, Jarecki, Krawczyk, & Rabin, 1996) and periodical updates on the participants' secret sharing called proactive secret sharing (Frankel, Gemmell, MacKenzie, & Yung, 1997; Herzberg, Jarecki, Krawczyk, & Yung, 1995). Applied in MANETs and WSNs, some of them require a trusted centralized authority to bootstrap the secret sharing procedure, while others provide joint secret sharing and do not require any trusted authorities.

Zhou & Haas (1999) used certificate based cryptography (CBC) and *(t,n)-threshold* cryptography in MANET. Let *N* be overall number of nodes and *t, n* be the two integers of threshold parameters, and

$t \leq n > N$. Prior to network deployment, the certificate authority CA's public key is furnished to each node, while each node's private key is divided into *n* shares, each uniquely assigned to one of *n* chosen nodes - let us denote as D-CAs. During network operations, any *t* D-CAs can work together to perform certificate generation and revocation using their secret share, while any less than t D-CAs cannot restore the secret key. Yi and Kravets (Yi & Kravets, 2003) proposed that it is better to select computationally more powerful and physically more secure nodes as D-CAs; this is consideration from the network and noticing the difference among nodes in the MANET, it is more practical. Both schemes can tolerate the compromise of up to $(t-1)$ D-CAs and the failure of up to $(n-t)$ D-CAs according to *(t,n)-threshold* cryptography.

Another approach to apply threshold cryptography in MANETs is URSA (Kong, Zerfos, Luo, Lu & Zhang, 2001; Luo & Lu, 2004), which is a (t, N) threshold scheme where *N* is the overall number of nodes. The advantage of URSA is network benefit; it increases the service availability because a certificate can be generated by any *t* nearby nodes or revoked by any *t* nearby nodes. The pitfall of the design is that the compromise of any *t* out of *N* nodes would break the secret key which is certificate authorities CA's private key, it leads to loss of overall system security. From the network attacks analysis, several security problems have been studied (Douceur, 2002; Jarecki, Saxena, & Yi, 2004; Narasimha, Tsudik, & Yi, 2003). One major problem is the Sybil (Douceur, 2002) attack, in which an attacker can take as many identities as necessary to collect shares and once it reaches the threshold it may then reconstruct the CA's private key.

Another approach to use threshold cryptography CBC schemes for MANET is to let each node act as a CA to issue certificates to other nodes (Capkun, Buttyan, & Hubaux, 2003; Gouda & Jung, 2004). This approach is less suitable in MANETs, but it is good for authority-free civilian networks. The IBC-based certificate-less public-key management schemes for MANETs (Deng, Mukherjee, & Agrawal, 2004; Khalili, Katz, & Arbaugh, 2003; Saxena, Tsudik, & Yi, 2004; Zhang, Liu, Lou, Fang, & Kwon, 2005) were also developed and some of them used threshold cryptography. Table 7 demonstrates selection criteria applied threshold cryptography in MANET/WSN. Table 8 is an illustration of the various schemes, main features, pros and cons.

From a cryptography technique study viewpoint of threshold cryptography, there are some thoughts and ideas that are very helpful to consider and evaluate before it is applied in MANET and WSN security design regardless of whether threshold cryptography is used alone or with other cryptography techniques, applied in key management, secure routing, or intrusion detection. We collect some of the questions and answers in Table 9 seen in papers (Jarecki, Saxena, & Yi, 2004; Narasimha, Tsudik, & Yi, 2003; Zhang, Liu, Lou, & Fang, 2006); and display them as a collection of design knowledge related to threshold cryptography.

Table 7: Threshold cryptography a variety of criteria applied in MANET/WSN schemes.

Certificate authority	**Quantities of CA**	**Asymmetric cryptography**	**Private key**
Selective on network node, not any node in network.	Selective on quantities of CA, not one unique CA.	Selective on RSA/DSA based asymmetric cryptography or IBC based one.	Selective on network-wide element's private key, not node specific element.

Table 8: Threshold cryptography a variety of usage in MANET.

Scheme information	**Main features**	**Pros/Cons**
Zhou & Hass (1999) CBC scheme	Choose *n* nodes to be D-CAs, each secret key to give n shares, threshold is t, $t \le n > N$.	Traditional approach
Yi & Kravets (2003) CBC scheme	Certificate authorities selected based on network factors: physical security, computation power, etc.	**Pros**: Consider network factors, thus smart choice from network viewpoint.
URSA (Kong, Zerfos, Luo, Lu & Zhang, 2001; Luo & Lu, 2004) CBC scheme	Each of *N* nodes is a D-CA, where *N* is the overall number of nodes.	**Pros:** Increase service availability, any *t* nearby nodes can provide service. **Cons**: Overall security is decreased, attacks such as Sybil.
Multiple CA (Capkun, Buttyan, & Hubaux, 2003; Gouda & Jung, 2004)	Each node acts as a CA.	**Cons**: Less authority available in network.
ID-GAC (Saxena, Tsudik, & Yi, 2004) IBC based	IBC-based access control scheme.	**Pros**: Apply in MANET service availability same as URSA does. **Cons**: Overall security is decreased.
IKM, IBC based scheme	There are two parts of public/private keys which are node-specific keys and network-wide common keys; nonetheless, the threshold cryptography only applies to network-wide common element.	**Cons**: Node-specific key elements ensure the secrecy of noncomprised nodes's private key; common key elements enable efficient key updates via a single broadcast message.

Table 9: Design questions related to threshold cryptography technique applied in MANET/WSN.

Ideas of threshold cryptography	**MANET/WSN network advantage/pitfall analysis**
1. Threshold cryptography distributes the ability to	MANET/WSN network has better fault tolerance than non-threshold cryptography, better security.

decryption or signing etc. service, what is the advantage to use threshold cryptography?	
2. Is secret share verifiable?	If not, it cannot be used in a setting where malicious insiders can exist. It requires a trusted third party to initialize the group during bootstrapping.
3. What is a common problem if MANET has one single trusted authority, one certificate authority CA?	One single trusted authority introduces a single point of failure attack, limited scalability.
4. What are some concerns to configure MANET using group authority instead of single one?	Although the group authority can be replicated for better availability, the scalability cannot be addressed by replication alone. Furthermore, unpredicted network faults and partitions complicate placement of group authority "replicas" in the network.
5. Is a fixed threshold policy applicable?	Sometimes it is necessary to reduce the threshold t to motivate the group to operate. A large group of nodes leave the network, resulting in a new smaller group size.
6. How to determine dynamic group size if using dynamic threshold cryptography?	MANET and WSN have distributed, asynchronous and decentralized dynamic group setting. Therefore, every member can send periodically a heart-beat message to the trusted authority to maintain the group size.
7. What cryptograph library is commonly used in MANETs?	MIRACL (Chor, Goldwasser, Micali, & Awerbuch, 1985), a standard cryptographic library which is used in IKM.

Other cryptographic techniques applied in security of MANET/WSN

From Figure 1 (d), we can tell that the batch verification cryptographic technique is a representative of many other cryptographic techniques which can be applied in MANET/WSN security. By looking up new research results of applied cryptography, applying into MANET/WSN security is not as far from reality as we have seen before. It takes time and effort to digest the cryptographic techniques, put together the security analysis, and make up an innovative design.

As seen in Figure 1 (a) and (d), there are four categories of cryptography, in which "*others"* is the topic we like to discuss. A special cryptography technique called batch verification with ID-based signature (Yoon, Cheon, & Kim, 2004), and its application in emerging area vehicular delay tolerant networks, is the third case study in this chapter.

Vehicular sensor networks (VSNs) have been envisioned to be useful in many commercial applications and in road safety systems. It is common practice to apply a digital signature scheme to countermeasure the attacks and resource abuse for VSNs. Consider the fact that a roadside unit cannot handle a large number of signatures received within the short interval according to the dedicated short range communication broadcast protocol (DSRC). A cryptography technique called the batch signature verification scheme based on ID-based cryptography (Fiat, 1989) is used and applied to communication between vehicles and roadside units. A roadside unit can verify multiple received signatures at the same

time to reduce the total verification time dramatically. In VSNs, the scheme is designed to employ identity-based cryptography to generate private keys for pseudo identities, achieve conditional privacy preservation, and reduce transmission overhead.

There are so many cryptographic techniques that can be applied in the security of MANETs/WSNs. The latest research on cryptography can be advanced so quickly that a new scheme applied in MANET/WSN can dramatically change the performance of that network, for example, the following case study will focus on the vehicular sensor network and improving the batch verification of signatures from linear time to constant time, which actually is algorithm optimization problem in applied cryptography.

Case study 3: an identity-based batch verification scheme IBV

The design of a new security scheme can be very complicate, but for simplicity, we like to go through a simple algorithm run time analysis case, IBV scheme, which demonstrates the design based on applied cryptography. The node identity is used in the IKM scheme, whereas pseudo identity is used with network context in the IBV scheme. Multiple batch verification schemes are updated in applied cryptography; therefore, IBV scheme is designed to adapt the work in MANETs/WSNs.

The batch verification scheme is designed to handle all the signatures received in a time window with less time compared to verify each signature one after the other. There are several batch cryptographic techniques.

Fiat (1989) introduced batch cryptography in 1989, and several other batch schemes (Cha, & Cheon, 2003; Naccache, M'Raihi, Vaudenay, & Raphaeli, 1994; Yoon, Cheon, & Kim, 2004; Zhang, & Kim, 2003; Zhang, Safavi-Naini, & Susilo, 2003) were proposed later. The batch verification scheme (Camenisch, Hohenberger, & Pedersen, 2007) is based on the CL signature scheme, and the scheme is a batch verification with high computation efficiency because it doesn't use random oracles. The batch verification scheme takes constant time instead of linear time; for example, verifying n signatures takes *3* pairing operation instead of $3n$ pairing operations. So the batch verification can be applied to vehicular sensor network to achieve good scalability.

The Identity-based Batch Verification (IBV) scheme for vehicular traffic related message transmission includes four phases: the key generation and pre-distribution phase, the pseudo identity and private key generation phase, the message signing phase, and the batch verification phase.

Key generation and pre-distribution

The network is based on several assumptions when it starts. Each vehicle is equipped with a tamper-proof device, and there is trusted authority (TA) which is designed to check the vehicle's identity, and generate and pre-distribute the private master keys of the vehicles. Before the network deploys, the trusted authority sets up the system parameters for each road side unit and on board unit.

Let G be a cyclic additive group generated by *P*, and G_T be a cyclic multiplicative group. G and G_T have the same order *q* which is big prime number. Let $\psi : G_1 \times G_1 \to G_T$ be a bilinear map.

The trusted authority generates two master keys by randomly choosing $s_1, s_2 \in Z_q^* = \{a | 1 \le a \le q-1\}$, and computes $P_{pub1} = s_1 P$, $P_{pub2} = s_2 P$ as its public keys.

The tamper-proof device of each vehicle is preloaded with the parameters (s_1, s_2). Each road side unit and vehicle are preloaded with the public parameters $\{G, G_T, q, P, P_{pub1}, P_{pub2}\}$.

Each vehicle is assigned with a real identity, denoted as $RID \in G$, and a password, denoted as PWD, where RID uniquely identifies the vehicle, and the PWD is used for the tamper-proof device to do authentication.

Pseudo Identity Generation

The tamper-proof device is designed to generate random pseudo identities and corresponding private keys based on identity-based cryptography. The tamper-proof device is designed in the IBV scheme to be composed of three secure modules: an authentication module, a pseudo identity generation module, and a private key generation module.

The authentication module is used to protect the tamper-proof device even if it is physically held by the adversary. It adds the authentication to use the service of the device. In the IBV scheme, the RID is the vehicle's unique real identity and the password PWD can be generated in different ways. The PWD is chosen to be generated by a trusted authority TA as the signature of RID.

The pseudo identity generation module is designed to generate a list of random pseudo identities from the authentication RID. Each pseudo identity ID is composed of ID_1 and ID_2. The formula to generate ID_1 and ID_2 is: $ID_1 = rP$, and $ID_2 = RID \oplus H(rP_{pub1})$ where *r* is random nonce and *r* is changed each time so that ID_1 and ID_2 are different for each pseudo ID. $\oplus$ is an Exclusive-OR(XOR) operation. *P* and P_{pub1} are the public parameters preloaded by the trusted authority TA. ID_1 and ID_2 are used by the third module: the private key generation module.

Private key generation module uses identity based cryptography. There are two corresponding parts private keys to the pseudo identity two parts, denoted as SK_1 and SK_2. And $SK_1 = s_1 ID_1$ and $SK_2 = s_2 H(ID_1 \| ID_2)$, in which $\|$ is the message concatenation operation.

A vehicle can go through the tamper-proof device using PWD and RID and get a list of pseudo identities $ID = (ID_1, ID_2)$ and the associated private keys $SK = (SK_1, SK_2)$. One comment is that the pseudo identities and the private keys can be generated offline by the tamper-proof device.

Message Signing

Vehicles can sign a message and send it to the roadside unit. In the IBV scheme, the message signing phase is designed as follows.

Suppose the traffic message, denoted by M_i, is generated by a vehicle, denoted by V_i. V_i goes through the tamper-proof device and get a pseudo identity $ID^i = (ID_1^i, ID_2^i)$ and the corresponding private key $SK^i = (SK_1^i, SK_2^i)$. The vehicle V_i can compute the signature σ_i of the message M_i, where $\sigma_i = SK_1^i + h(M_i)SK_2^i$. Subsequently, the vehicle V_i sends the final message $\langle ID^i, M_i, \sigma_i \rangle$ to its neighboring roadside unit. These steps are done once every *100-300ms* according to the current dedicated short range communication broadcast protocol (DSRC).

The signature of the IBV scheme has no need for any signature certificate to be sent along with the message because the identity-based cryptography is used. Only a pseudo identity is sent, which has a length of *42* bytes, the sum of length of ID_1^i and ID_2^i. Compared with the ECDSA signature scheme of *IEEE 609.2*, a certificate in the message is *125* bytes long.

Secondly, the signature of the IBV scheme does not release any real identity information of the vehicle because of pseudo identity is used in the scheme.

Batch Verification

When a road side unit (RSU) receives a traffic related message from a vehicle in the IBV scheme, the RSU has to verify the signature of the message for two reasons: one is to make sure the corresponding vehicle doesn't impersonate any other legitimate vehicle, another is to prevent the vehicle from disseminating bogus messages. The verification process of the IBV scheme is illustrated in the following single signature verification and batch verification in detail.

Given the system public parameters $\{G, G_T, q, P, P_{pub1}, P_{pub2}\}$ preloaded on each RSU and vehicle in the network in IBV scheme assigned by the trusted authority TA and the message $\langle ID^i, M_i, \sigma_i \rangle$ sent by the vehicle V_i, the signature σ_i can be validated by testing if $\Gamma(\sigma_i, P) = \Gamma(ID_1^i, P_{pub1})\Gamma(h(M_i)H(ID_1^i, ID_2^i), P_{pub2})$, as verified below using bi-linear maps bi-linear feature. Therefore, the computation cost for the RSU to verify a single signature is mainly one MapToPoint hash (Boneh, Lynn, & Shacham, 2001), one multiplication, and three pairing operations.

Given *n* distinct messages denoted as $\langle ID^1, M_1, \sigma_1 \rangle$, $\langle ID^2, M_2, \sigma_2 \rangle$,…,$\langle ID^n, M_n, \sigma_n \rangle$, respectively, which are sent by *n* distinct vehicles denoted as V_1, V_2, …, V_n, all signatures, denoted as $\sigma_1, \sigma_2, \ldots, \sigma_n$ are valid if $\Gamma(\sum_{i=1}^{n} \sigma_i, P) = \Gamma(\sum_{i=1}^{n} ID_1^i, P_{pub1})\Gamma(\sum_{i=1}^{n} h(M_i)HID^i, P_{pub2})$, in which HID^i denotes $H(ID_1^i \| ID_2^i)$. Detail verification can be found in the IBV scheme paper (Zhang, Lu, Ho & Shen, 2008).

Batch verification in the IBV scheme reduces the verification delay, and the computation cost that the RSU verifies *n* signature can be deduced to *n* MapToPoint hash, *n* multiplication, *3n* addition, *n* one-way hash, and *3* pairing operations. Because the computation cost of a pairing operation is much higher than the cost of a MapToPoint hash and a multiplication cost, the verification time for multiple signatures is constant instead of linear with the size of the batch.

Security Analysis

The IBV scheme design is based on ID-based batch verification which can improve efficiency when a lot of signatures have to be verified. With the rising interest on pair-based cryptography, much research on identity-based signatures and also on enhancing performance of verifying identity-based signatures by a batch verification has been proposed. Here we focus on the IBV scheme security analysis - the basics of the cryptography foundation that supports the IBV scheme. The following three aspects of security analysis will be presented: the message authentication, the user identity privacy preservation, and the traceability by the trusted authority.

- **Message authentication in the IBV scheme:** Let us review the IBV signature, $\sigma_i = SK_1^i + h(M_i)SK_2^i$ is a one-time identity-based signature. It is impossible to forge an IBV signature without knowing the private key SK_1 and SK_2. The NP-hard computational complexity of the Diffie-Hellman problem in G makes the private key SK_1 and SK_2 infeasible to be derived from ID_1, P_{pub1}, P and $H(ID_1 \| ID_2)$. The Diphantine equation is used to construct the IBV signature σ_i, it is infeasible to compute the private key SK_1 and SK_2 by the knowledge of σ_i and $h(M_i)$.
- **Identity privacy preserving:** In the design of IBV scheme, the identity privacy preserving is implemented using the ElGamal-type ciphertext construction, the real identity RID of a vehicle is used to construct two random pseudo identities ID_1 and ID_2, where $ID_1 = rP$ and $ID_2 = RID \oplus H(rP_{pub1})$, in which *r* is random number, *P* and P_{pub1} are public parameters that are preloaded on each roadside unit and vehicle. The master-key (s_1, s_2) is preloaded with each vehicle tamper-proof device. So without the master key (s_1, s_2), it is impossible to get the real identity from the pseudo identity pair. Also, because the pseudo identities (ID_1, ID_2), in each signature are distinct, it is not helpful to compound the series of signatures to get real identity. In other words, there is no linkability.
- **Traceability by the trusted authority:** In proposed IBV scheme, the trust authority (TA) has the way to verify if the signature is authentic or not by using the master key (s_1, s_2) which is preloaded to each tamper-proof device on each vehicle. Because computing $ID_2 \oplus H(s_1 ID_1)$ can get value of *RID* from the following steps: $ID_2 \oplus H(s_1 ID_1) = RID \oplus H(rP_{pub1}) \oplus H(s_1 rP)$, $P_{pub1} = s_1 P$. From the above two equations, we get $ID_2 \oplus H(s_1 ID_1) = RID \oplus H(rs_1 P) \oplus H(s_1 rP)$, in which $H(rs_1 P) \oplus H(s_1 rP) = 1$. So above all, we conclude that $ID_2 \oplus H(s_1 ID_1) = RID$.

OPEN CHALLENGES AND FUTURE DIRECTIONS

Security of MANETs/WSNs will continuously be under research and development in academia and industry as advanced technology updates are made. The ad hoc network is updated with new applications in emerging areas such as vehicular sensor network research, with popularity of Global Positioning System (GPS) or other facilities set up much more easily and faster such as wireless device technology, small screens in cellular phone technologies, new challenge will be come out for the research work of MANET/WSN security.

More research on secure MANETs/WSNs will focus on long term effects instead of most expeditious results. To continually focus on achieving concrete results, some researchers contribute to MANETs/WSNs security by dealing with real specific problems. Our previous work (Wu & Chen, 2008) is the investigation of attacks and countermeasures in MANETs according to different network layers. More to drive for results, Kannhavong (Kannhavong, Nakayama, Nemoto, & Kato, 2007) did a survey on routing attacks and countermeasures against those attacks in MANETs after our work. Certainly to balance the drive for expeditious results and long term effect, our current research is based on the foundational knowledge of security research of MANETs/WSNs, in which we fully investigate the cryptographic primitives by our roughly categorized techniques, and cover a variety of topics ranging from security routing protocol to broadcast communication, group key management, composite key management scheme and single cryptography techniques such as batch verification. Compared to previous works (Kannhavong, Nakayama, Nemoto, & Kato, 2007; Wu & Chen, 2008), we look for ways to improve the long-term research results of security of MANETs/WSNs. As an adventurous trial, we effectively study applied cryptography to overcome difficult obstacles in understanding complicated security designs in this survey chapter.

The previous rapid advancement of cryptography showed the result of reducing the computational costs of outstanding cryptographic primitive operations in algorithms. In addition, the computation can be accelerated by using dedicated cryptographic hardware with cheaper hardware in the future. To support the above fact, in the IKM scheme's performance evaluation, IKM's computation cost as an IBC scheme is not only compared to RSA operations; however, Zhang (2006) also prompted us that the Barreto approach can expedite the Tate pairing to be up to *10* times faster than previous methods although the implementation is still underway. As a practical approach, researchers in the security of MANETs/WSNs may pay attention to applied cryptography research results in time to fully take advantage of performance gains through improved algorithms in applied cryptography.

As always, key management is a fundamental and challenging issue, and with rapid advancement in cryptography research, it brings more topics to the research field. With wireless network security technology advancing more quickly in our daily business life, it is much easier to form a MANET/WSN. The cryptographic techniques always play a major role in the design of each stage of the key management. The art of the design can be better evaluated from the conceptual level to the implementation of the simulation study. The security of design will be dissected more by the research community and the new design will come out quickly and easily reusable as popular “design patterns” using cryptography terminologies. More variations in selection chosen by the designer will still heavily depend on the knowledge and skills level of cryptographic techniques, such as hiding the real identity of a vehicle in the IBV scheme, there are several alternative designs beside the ElGamal type ciphertext.

The privacy issues and all other non-cryptography based security solutions can also be under the research community work mainly seen from data mining and machine learning area; for example, privacy model and algorithms, attacks using background knowledge and patterns (Aggarwal, & Yu, 2008). It is not required that we have "cryptographic techniques" in each security scheme design, while association rules hiding and many other techniques from statistics, combinatorics, data mining can be helpful in privacy of ad hoc network area if they are applied to ad hoc network area. Much research has been done related to MANET/WSN location privacy.

From a network research perspective, the MANET/WSN specific area needs to be looked into when security routing or key management issues are required. Because there are a variety of cases involving MANETs/WSNs, the network scalability, computer cost, and resource constraints vary and may have to be considered case by case, such as vehicular sensor network, it is more relaxation power and processing constraints than MANETs, and the vehicle has temporary infrastructure access via road-side units as seen in the IBV scheme and public hot-spots. The symmetric cryptography and asymmetric cryptography, and their customized usage according to different network stages, will always be a challenge to cover the wide range of network layers in MANETs/WSNs. The current cryptography library and available MANET simulator or self-developed simulation study also will be advanced by the talent of the research community in different areas.

MANET/WSN research will progress more quickly if emphasis is placed on the cryptographic techniques and a study of each cryptographic technique is unique contribution to MANET/WSN security by case study and discussion. The security of MANET/WSN research will attract more talent with various experiences and should be normalized by more outstanding work with simplicity, abstraction level, and popularity to move forward along with all other innovative technology we have.

ACKNOWLEDGEMENT

This work was supported in part by NSF grants ANI 0073736, EIA 0130806, CCR 0329741, CNS 0422762, CNS 0434533, CNS 0531410, and CNS 0626240.

REFERENCES

Aggarwal, C., & Yu, P. (2008). Privacy-Preserving Data Mining Models and Algorithms. Springer Science+Business Media, LLC.

Barreto, P., Kim, H., Bynn, B., & Scott, M. (2002). Efficient Algorithms for Pairing-Based Cryptosystems. *Proc. CRYPTO* (pp. 354-368).

Boneh, D., & Franklin, M. (2001). Identify-Based Encryption from the Weil Pairing. *Proc. CRYPTO'01* (pp. 213-229).

Boneh, D. & Franklin, M. (2003, March). Identify-Based Encryption from the Weil Pairing. *SIAM J. Computing vol. 32, no. 3* (pp. 586-615).

Boneh, D., Lynn, B., & Shacham, H. (2001). Short signatures from the weil pairing. In *Proceedings of Asiacrypt, Vol.2248* (pp.514-532).

Camenisch, J., Hohenberger, S., & Pedersen, M. (2007). Batch verification of short signatures. In *Proceedings of EUROCRYPT, LNCS, Vol. 4514* (pp. 246-263).

Camenisch, J., & Lysyanskaya, A. (2004). Signature schemes and anonymous credentials from bilinear maps. In *Proceedings of Crypto, LNCS, Vol. 3152* (pp. 56-72).

Capkun, S., Buttyan, L., & Hubaux, J.P. (Jan. - Mar. 2003). Self-Organized Public Key Management for Mobile Ad Hoc Networks. *IEEE Trans. Mobile Computing, vol.2, no.1* (pp.52-64).

Carter, S., & Yasinsac, A. (2002, November). Secure Position Aided Ad hoc Routing Protocol. In *Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN02).*

Cha, J.C., & Cheon, J.H. (2003). An identity-based signature from gap Diffie- Hellman groups. In *Proceedings of Public Key Cryptography* (pp. 18-30).

Chor, B., Goldwasser, S., Micali, S., & Awerbuch, B. (1985). Verifiable secret sharing and achieving simultaneity in the presence of faults. In *FOCS*.

Clausen, T., Adjih, C., Jacquet, P., Laouiti, A., Muhlethaler, A., & Raffo, D. (2003, June). Securing the OLSR Protocol. In *Proceeding of IFIP Med-Hoc-Net*.

Dedicated Short Range Communications (DSRC), [Online]. Available: http://grouper.ieee.org/groups/scc32/dsrc/index.html

Deng, H., Mukherjee, A., & Agrawal, D. (2004, April). Threshold and Identity-Based Key Management and Authentication for Wireless Ad Hoc Networks. *Proc. Int'l Conf. Information Technology: Coding and Computing (ITCC '04)*.

Desmedt, Y., & Frankel, Y. (1989, August). Threshold Cryptosystems. *Proc. CRYPTO '89* (pp. 307-315).

Dong, Q., Liu, D., & Ning, P. (2008). Pre-Authentication Filters: Providing DoS Resistance for Signature-Based Broadcast Authentication in Wireless Sensor Networks. In *Proceedings of ACM Conference on Wireless Network Security* (WiSec).

Douceur, J.R. (Mar. 2002). The Sybil Attack. *Proc. First Int'l Workshop Peer-to-Peer Systems (IPTPS '02)* (pp. 251-260).

Du, W., et al. (2003). A Pairwise Key Predistribution Scheme for Wireless Sensor Networks. In *Proc. 10th ACM Conf. Comp. Commun. Sec.* (pp. 42-51).

Eschenauer, L., & Gligor, V.D. (2002). A Key-Management Scheme for Distributed Sensor Networks. In *Proc. 9th ACM Conf. Comp. and Commun. Sec.* (pp. 41-47).

Evans, J., Wang, W., & Ewy, B. (2006). Wireless networking security: open issues in trust, management, interoperation, and measurement. *International Journal of Security and Networks (IJSN), vol.1, no. ½* (pp. 84-94).

Fiat, A. (1989). Batch RSA. In *Proceedings of Crypto* (pp. 175-185).

Frankel, Y., Gemmell, P., MacKenzie, P.D., & Yung, M. (1997). Optimal resilience proactive public-key cryptosystems. In *FOCS*.

Gouda, M.G., & Jung, E. (Mar. 2004). Certificate Dispersal in Ad-Hoc Networks. *Proc. 24th IEEE Int'l Conf. Distributed Computing Systems* (ICDCS '04).

Gennaro, G., Jarecki, S., Krawczyk, H., & Rabin, T. (1996). Robust and efficient sharing of rsa functions. In *CRYPTO*.

Herzberg, A., Jarecki, S., Krawczyk, H., & Yung, M. (1995). Proactive secret sharing, or: How to cope with perpetual leakage. In *CRYPTO*.

Hu, Y., Johnson, D., & Perrig, A. (2002). SEAD: Secure Efficient Distance Vector Routing in Mobile Wireless Ad-Hoc Networks. *Proc. of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02)* (pp. 3-13).

Hu, Y., Perrig, A., & Johnson, D. (2002). Ariadne: A Secure On-Demand Routing for Ad Hoc Networks. *Proc. of MobiCom 2002*, Atlanta.

Huang, J., Buckingham, J., & Han, R. (Sep. 2005). A Level Key Infrastructure for Secure and Efficient Group Communication in Wireless Sensor Networks. In *Proc. 1st Int'l. Conf. on Security and Privacy for Emerging Areas in Commun. Net.* (pp. 249-260).

IEEE Standard 1609.2 - IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - *Security Services for Applications and Management Messages*, 2006.

Ilyas, M. (2003). *The Handbook of Ad Hoc Wireless Networks*. CRC Press.

Jarecki, S., Saxena, N., & Yi, J.H. (Oct. 2004). An Attack on the Proactive RSA Signature Scheme in the URSA Ad Hoc Network Access Control Protocol. *Proc. Second ACM Workshop Security of Ad Hoc and Sensor Networks (SASN '04)*.

Kannhavong, B., Nakayama, H., Nemoto, Y., & Kato, N. (October 2007). A Survey of Routing Attacks in Mobile Ad Hoc Networks. In *IEEE Wireless Communications* (pp. 85-91).

Kaufman, C., Perlman, R., & Speciner, M. (2002). *Network Security Private Communication in a Public World*. Prentice Hall PTR, A division of Pearson Education, Inc.

Kaya, T., et al., Secure Multicast Groups on Ad Hoc Networks. (Oct. 2003). In *Proc. ACM SASN '03* (pp. 94-103).

Khalili, A., Katz, J., & Arbaugh, W. (2003, January). Toward Secure Key Distribution in Truly Ad Hoc Networks. *Proc IEEE Workshop Security and Assurance in Ad Hoc Networks*.

Kong, J., Zerfos, P., Luo, H., Lu, S., & Zhang, L. (Nov. 2001). Providing Robust and Ubiquitous Security Support for Mobile Ad Hoc Networks. *Proc. IEEE Int'l Conf. Network Protocols*.

Lee, J., Leung, V., Wong, K., Cao, J., & Chan, H. (2007, October). Key management issues in wireless sensor networks: current proposals and future developments. In *IEEE Wireless Communications* (pp. 76-84).

Liu, A., & Ning, P. (2008, April). TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), SPOTS Track* (pp. 245-256).

Lou, W., & Fang, Y. (2003). A Survey of Wireless Security in Mobile Ad Hoc Networks: Challenges and Available Solutions. In X. Chen, X. Huang and D. Du. Kluwer (Ed.), *Ad Hoc Wireless Networks* (pp. 319-364). Academic Publishers.

Luo, H., & Lu, S. (2004). URSA: Ubiquitous and Robust Access Control for Mobile Ad-Hoc

Networks. *IEEE/ACM Transactions on Networking*. *Vol.12 No.6* (pp. 1049-1063).

Mehuron, W. (1994). Digital Signature Standard (DSS). *U.S. Department of Commerce, National Institute of Standards and Technology (NIST), Information Technology Laboratory (ITL)*. FIPS PEB 186.

Menezes, A., Oorschot, P., & Vanstone, S. (1996). *Handbook of Applied Cryptography*. CRC Press.

Naccache, D., M'Raihi, D., Vaudenay, S., & Raphaeli, D. (1994). Can D.S.A be improved? complexity trade-offs with the digital signature standard. In *Proceedings of EUROCRYPT, LNCS, Vol. 950* (pp. 77-85).

Narasimha, M., Tsudik, G., & Yi, J.H. (Nov. 2003). On the Utility of Distributed Cryptography in P2P and Manets: The Case of Membership Control. *Proc. IEEE Int'l Conf. Network Protocols*.

Panja, B., Madria, S. K., & Bhargava, B. (2006). Energy and Communication Efficient Group Key Management Protocol for Hierarchical Sensor Networks, SUTC '06. In *Proc. IEEE Int'l. Conf. Sensor Networks, Ubiquitous, and Trustworthy Comp.* (pp. 384-393).

P. Papadimitratos & Z. J. Haas. (2003, January) Secure Link State Routing for Mobile Ad Hoc Networks. In *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03)* (pp.379-383) Washington, DC, USA.

Perrig, A., Canetti, R., Tygar, J, & Song, D. (2000). The TESLA Broadcast Authentication Protocol. Internet Draft.

Perrig, A., Szewczyk, R., Wen, V., Culler, D., & Tygar, D. (2001, July). SPINS: Security protocols for sensor networks. In *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks* (MobiCom).

Perkins, C. (2001). *Ad Hoc Networks*. Addison-Wesley.

Pervaiz, M.O., Cardei, M., & Wu, J. (2008). Routing Security in Ad Hoc Wireless Networks. In S. Huang, D. MacCallum, and D. -Z. Du (Ed.), *Network Security*. Springer.

Salomaa, A. (1996). *Public-Key Cryptography*. Springer-Verlag.

Sanzgiri, K., Dahill, B., Levine, B., Shields, C., & Belding-Royer, E. (2002). A Secure Routing Protocol for Ad Hoc Networks. *Proc. of IEEE International Conference on Network Protocols (ICNP)* (pp. 78-87).

Srinivasan, A., Wu, J. (2008). A Survey on Secure Localization in Wireless Sensor Networks. In B. Furht (Ed.), *Encyclopedia of Wireless and Mobile Communications*. CRC Press, Taylor and Francis Group.

Saxena, N., Tsudik, G., & Yi, J. (Dec. 2004). Identity-Based Access Control for Ad Hoc Groups. *Proc. Int'l Conf. Information Security and Cryptology*.

Shamir, A. (1979). How to Share a Secret. *Comm. ACM, vol. 22, no. 11* (pp.612-613).

Shamir, A. (1984). Identity Based Cryptosystems and Signature Schemes. *Proc. CRYPTO'84* (pp. 47-53).

Striki, M., & Baras, J. (2004, June). Towards Integrating Key Distribution with Entity Authentication for Efficient, Scalable and Secure Group Communication in MANETs. *In Proc. IEEE ICC'04, vol.7* (pp. 4377-4381).

Wu, B., Cardei, M., & Wu, J. (2008). A Survey of Key Management in Mobile Ad Hoc Networks. In J. Zheng, Y. Zhang, and M. Ma (Ed.), *Handbook of Research on Wireless Security*. Idea Group Inc..

Wu, B., Chen, J., Wu, J., & Cardei, J. (2008). A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks, In Y. Xiao, X. Shen, and D. -Z. Du (Ed.), *Wireless/Mobile Network Security*. Springer.

Xiao, Y, & Rayi, V. (2007). A survey of key management schemes in wireless sensor networks. *Computer Communications*.

Yi, S., Naldurg, P., & Kravets, R. (2002). *Security Aware Ad hoc Routing for Wireless Networks*. (Tech. Rep. No. UIUCDCS-R-2002-2290). Illinois, United States of America: University of Illinois at Urbana-Champaign.

Yi, S. & Kravets, R. (2003, April). Moca: Mobile Certificate Authority for Wireless Ad Hoc Networks. *Proc. Second Ann. PKI Research Workshop (PKI '03).*

Yoon, H., Cheon, J.H., & Kim, Y. (2004). Batch verification with ID-based signatures. In *Proceedings of Information Security and Cryptology* (pp. 233-248).

Younis, M. F., Ghumman, K., & Eltoweissy, M. (2006). Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks. In *IEEE Trans. Parallel and Distrib. Sys., vol.17*, 2006 (pp. 865-882).

M. Zapata. (2002). Secure Ad Hoc On-Demand Distance Vector (SAODV). Internet draft, draft-guerrero-manet-saodv-01.txt.

Zhang, C., Lu, R., Ho, P., & Shen, X. (2008). An Efficient Identity-based Batch Verification Scheme for Vehicular Sensor Networks. *Proc. Of IEEE INFORCOM*.

Zhang, F., & Kim, K. (2003). Efficient ID-based blind signature and proxy signature from bilinear pairings. In *Proceedings of ACISP, LNCS, Vol. 2727* (pp. 312-323).

Zhang, F., Safavi-Naini, R., & Susilo, W. (2003). Efficient verifiably encrypted signature and partially blind signature from bilinear pairings. In *Proceedings of Indocrypt, LNCS, Vol. 2904* (pp. 191-204).

Zhang, Y., Liu, W., Lou, W., Fang, Y., & Kwon, Y. (2005, May). AC-PKI: Anonymous and Certificateless Public-Key Infrastructure for Mobile Ad Hoc Networks. *Proc. IEEE Int'l Conf. Comm* (pp. 3515-3519).

Zhang, Y., Liu, W., Lou, W., & Fang, Y. (2006, Oct.-Dec.). Securing mobile ad hoc networks with certificateless public keys. *IEEE Transactions on Dependable and Secure Computing* (Vol. 3, No. 4, pp 386-399).

Zhang, Y., Liu, W., & Lou, W. (2005, March). Anonymous Communications in Mobile Ad Hoc Networks. *Proc. IEEE INFOCOM '05* (pp. 1940-1951).

Zhou, L., & Haas, Z. (1999). Securing Ad Hoc Networks. *IEEE Network Magazine, Vol.13 No.6* (pp. 24-30).

Zhu, S., Xu, S., Setia, S., & Jajodia, S. (2003). LHAP: A Lightweight Hop-by-Hop Authentication Protocol for Ad-Hoc Networks. *23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03)*.

Zhu, S., Setia, S., & Jajodia, S. (2003). LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. In *Proc. 10th ACM Conf. Comp. and Commun. Sec.* (pp. 62-72).