On Key Distribution in MANETs

Alexis Bonnecaze  
Université de la méditerranée,  
IML, ERISCS  
Marseille, France  
Email: bonnecaze@univmed.fr

Alban Gabillon  
Université de Polynésie Française  
Tahiti, France  
Email: alban.gabillon@upf.pf

Abstract—We first review and compare private key distributions in MANETs based on secret sharing schemes. Since there exist many kinds of networks with different objectives and constraints, we then determine the suitable private key distribution scheme according to the nature of the network.

We also stress that identity-based cryptography represents a valuable security solution as it provides many interesting features for MANETs.

Keywords—Key Distribution, MANETs, Verifiable Secret Sharing, Identity Based Cryptography

I. INTRODUCTION

MANETs are self-configuring networks of mobile nodes without the presence of static infrastructure. They can also be heterogeneous, which means that all nodes don’t have the same capacity in terms of resources (power consumptions, storage, computation, etc.). A good example is given by military battlefield networks. In that case, mobile devices have different communications capability such as radio range, battery life, data transmission rate, etc. Some nodes are then more vulnerable to attacks and may be considered as less reliable than others. Moreover, MANETs don’t have a fixed size: nodes can join or leave the network dynamically. When joining the network, nodes need public and private keys (we assume the network has the computational ability to allow asymmetric cryptography). In absence of a central administration, key management must be self-organized by the nodes. Nodes obtain their keys with the help of other nodes, called master. Networks generally use a threshold scheme: a node must request at least \( t + 1 \) master nodes out of \( n \) in order to obtain its key.

Key management represents a big concern in MANETs. Identity-based cryptography (IBC) has been considered in many recent papers (for example [6], [10], [11]) in order to avoid the use of a heavy public key infrastructure (PKI). It is well known that IBC can simplify systems that manage a large number of public keys.

In this paper, we explain why identity-based is an appropriate concept for key management in MANETs. Then we analyse and compare private key distribution solutions based on secret sharing schemes. Section II introduces identity-based cryptography schemes and lists the advantages of using IBC in MANETs. Secret sharing schemes are introduced in Section III. In this section, we describe and evaluate different solutions to distribute the trusted third party (TTP).

II. IDENTITY-BASED CRYPTOGRAPHY IN MANETS

A. Description

In 1984 Shamir asked for a public key encryption and signature scheme in which the public key can be an arbitrary string. These schemes are called identity-based cryptography schemes (IBC). Shamir [18] easily constructed an identity-based signature (IBS) scheme using the RSA function, but he was unable to construct an identity-based encryption (IBE) scheme, which became a long-lasting open problem. In 2001, Boneh and Franklin used Weil pairing to introduce the first ID-based encryption scheme [1].

In an IB scheme, the public key of a user is directly derived from his identity. It can be an email address, and in the case of a MANET, the MAC address of the device or any identity bounded to the hardware of the device. Encrypting a message only requires to have the correct public key and some public parameters of a trusted party called Private Key Generator (PKG). There is no need to obtain public key certificate and it is possible to encrypt a message before the private key is computed. A user, who would like to decrypt a message, obtains his private key from the PKG. More precisely, an IB-Encryption scheme consists of four algorithms: (1) Setup generates global system parameters and a master-key, (2) Extract uses the master-key to generate the private key corresponding to an arbitrary public key string ID, (3) Encrypt encrypts messages using the public key ID, and (4) Decrypt decrypts messages using the corresponding private key. Algorithm Extract is achieved by the PKG and IBE systems provide key escrow since the PKG knows all private keys. In a MANET which has no central administration, the PKG must be distributed so that there is no single node that knows the secret master key. Secret sharing schemes implement such distribution.

B. Pairings

Let \( G \) be a cyclic (additive) group of a prime order \( q \) and \( G' \) be a multiplicative group of the same order \( q \). The group \( G \) is the group of points of an appropriate elliptic curve. A
map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}'$ is called a cryptographic bilinear map if it satisfies the following properties:

1) bilinearity: for all $P, Q \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, $e(aP, bQ) = e(P, Q)^{ab}$

2) non-degeneracy: $e(P, P)$ is a generator of $\mathbb{G}'$ and therefore $e(P, P) \neq 1$

3) computable: there exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}$.

The security of ID-based schemes is based on the assumed hardness of the computational Diffie-Hellman (CDH) problem in $G$.

Let $s \in_R \mathbb{Z}_q^*$ be the master key of the system. The corresponding public key is $P_{pub} = sP$.

Security of pairing-based schemes is based on problems which are considered more difficult than DLP over $\mathbb{Z}_q$. This is because known algorithms to solve these problems are exponential (or sub-exponential in some special cases). Hence, for the same security, these schemes use smaller keys. For example, RSA signature using a key of 1024 bits can be compared in term of security to BLS signature [2] using a key of length 160 bits. Even though the computational cost of a pairing is still high, computations are done in a smaller group and there exist now efficient implementations. Tate pairing, which has better computational performances than Weil pairing, has already been implemented on smartcards. In [16], it is demonstrated that, on smartcards, pairings can be calculated as efficiently as classic cryptographic primitives. It is then realistic to use IBC in MANETS. Moreover, pairing based cryptography provides us with many interesting primitives. For example, in the domain of authentication, there are many kind of short specific signatures, like group signatures, blind signatures, multi-signatures, or ring-signatures (see [4], [12]).

Note that a library (written in C), called Pairing-Based Cryptography (see [3]), is maintained on line and propose efficient implementation of pairings.

C. Features of IBC schemes in MANETS

In this section, we present some features of IBC schemes that can be implemented in MANETS. Some of them can improve the management of a MANET. Indeed, with such schemes there is no need for a certificate authority, the revocation process is simplified and there exists a secret key between each pair of nodes which allows the nodes to communicate without using asymmetric cryptography:

1) Self-authenticating public key: users don’t need certificates to authenticate public keys. Moreover, there is no need to exchange public key since there are known in advance. Communication overhead and memory space are then reduced. The public key $Q_i$ is predetermined in the following way. Let $H$ be a map-to-point hash function: $H : \{0, 1\}^* \rightarrow \mathbb{G}$ and $Id_i$ be the identity of node $N_i$, the public key is $Q_i = H(Id_i) \in \mathbb{G}$. The corresponding private key is $D_i = sQ_i$.

2) Key renewal: it is also possible to limit the validity period of the public key by concatenating an expiry date. Then we have $Q_i = H(Id_i || \text{'date'})$. The validity period has to be carefully chosen to minimize both the cost of key renewal and the probability of key compromise. When the key is compromised before the end of expiry date, the following format can be adopted (for more detail, see [10]): $Q_i = H_i(Id_i || \text{'date'} || \text{'version#'})$ and the version number must be broadcasted to all nodes.

3) Symmetric keys: when the network is fully operational, each node knows its own private key and the public key of any other node. Nodes could then communicate using asymmetric cryptography. However symmetric cryptography is more suitable in MANETs since the computational cost is lower. It is interesting to note that every pair of nodes $(N_i, N_j)$ can compute a secret key $K_{ij}$ without interacting. Node $N_i$ computes $e(D_i, Q_j)$ and node $N_j$ computes $e(D_j, Q_i)$. These two values are equal since $e(D_i, Q_j) = e(sQ_i, Q_j) = e(Q_i, sQ_j) = e(Q_i, D_j) = K_{ij}$ (here the pairing is symmetric).

4) Delegation of duties: this is another interesting feature that can be implemented in MANETs. Assume a node $N$ has several node assistants each being responsible for a given particular task. Node $N$ gives one private key to each of his assistants corresponding to the assistant’s responsibility. Each assistant can then decrypt messages whose subject line falls within its responsibility, but it cannot decrypt messages intended for other assistants. Here, messages are encrypted with the same public key and a key-word corresponding to a given responsibility. The message can only be read by the assistant responsible for that subject.

III. SECRET SHARING SCHEMES WITHOUT TRUSTED PARTY

A. Secure secret sharing schemes

MANETs generally belong to a unique entity like a company or a military administration. In that configuration, an external server TTP plays the role of a PKG and can distribute keys $D_i$ to all nodes before joining the network. Then, nodes can easily communicate using asymmetric or symmetric cryptography, as indicated before. But sometimes, the network must be fully self-organized. There is no trusted server to distribute keys to nodes and the master key of the PKG must not be known by any single entity. The idea is to construct a system in which each node of a fixed subset of nodes knows a part of the secret. This subset of nodes is the distributed representation of the PKG.

Secret sharing schemes are used to distribute a secret, which, in our case, is the master key $s$ of the PKG. Every node gets a share $s_i$ of the secret and the private key $D_i$ of a node can be computed by the collaboration of $t+1$ nodes. In its seminal paper [17], Shamir’s idea is based on the fact that the knowledge of $t+1$ points of a given polynomial $f(z) = \sum_{k=0}^t a_k z^k \in R, F_q[z]$ ($q$ being a large prime number) of
degree $t$ allows us to compute the coefficient $a_0 = f(0)$. The simplified protocol has the following steps:

1) TTP picks at random a polynomial $f$ of degree $t$
2) TTP sends to each user $i$ a point $(i, f(i))$ as a share
3) the secret key $a_0$ can be recovered by collusion of any group of $t + 1$ users using Lagrange interpolation.

As we consider fully self-organized network, the TTP must be distributed. In that case, $n$ nodes play the role of the TTP. These nodes are called master nodes. Each master node picks at random a polynomial $f_i (i = 1, \ldots, n)$ of degree $t$ and the sum of all of these polynomials play the role of the polynomial $f: f = \sum f_i$. The polynomial $f$ is indeed never constructed by any entity. Let $h$ be a hash function: $h : \{0, 1\}^* \rightarrow \mathbb{Z}_q$. We set $h_i = h(Id_i)$. The simplified protocol has the following steps:

1) node $N_i$ picks at random a polynomial $f_i$ of degree $t$,
2) node $N_i$ sends $f_i(h_i)$ to node $N_j$, for $j = 1, \ldots, n$.
3) the share of node $N_i$ is $s_j = \sum_i f_i(h_j) = f(h_j)$.

If at the end of this phase, a node $N_i$ (master or not) needs its private key. It sends to $t + 1$ master nodes its identity $Id_j \in \mathbb{G}$, and receives $f(h_i)Q_j$, $i$ being the index of the set of $t + 1$ nodes. Node $N_j$ computes then its private key: $D_j = \sum_i \gamma_j f(h_i)Q_j$, where $\gamma_j$ denotes appropriate Lagrange interpolation coefficients.

Shares must be periodically refreshed. To do it, each master node $N_i$ picks at random a new polynomial $g_i$ such that $g_i(0) = 0$ and sends the value $g_i(h_j)$ to node $N_j$. The new share of node $N_i$ is $f(h_i) + g(h_j) = (f + g)(h_j)$, and the secret is left unchanged.

In 1991, Pedersen [14] proposed a verifiable secret sharing protocol without third party, using Feldman’s protocol [7]. It acts as a distributed key generation in discrete-log based system. Let $g$ be an element of order $q$ in $\mathbb{Z}_p^*$, where $q | (p - 1)$. This protocol makes use of $n$ polynomials $f_i(z) = \sum_{k=0}^{q-1} a_{ik} z^k \in \mathbb{F}_q[z]$, $n$ being the number of users. More precisely, node $N_i$:

1) picks at random a polynomial $f_i$ of degree $t$,
2) broadcasts $g^{a_{ik}}$ for $k = 0, \ldots, t$,
3) sends $f_i(h_j)$ to user $N_j$ for $j = 1, \ldots, n$. The share of node $N_i$ is $s_j = \sum_i f_i(h_j)$.

The private key (the secret) is $s = \sum_{i=1}^{t} a_{i0}$ and the corresponding public key is $y = \prod_i y_i$, where $y_i = g^{a_{i0}}$. We note that $s$ can (only) be computed using any set of $t + 1$ correct shares, as $s = \sum_{i=1}^{t} a_{i0} = \sum_{i=1}^{t} (\sum_j \gamma_j f_i(h_j)) = \sum_j \gamma_j s_j$. Where $\gamma_j$ denotes appropriate Lagrange interpolation coefficients and $j$ belongs to any set of $t + 1$ correct shares.

In 1999, Gennaro et.al. [8] show that Pedersen protocol is not secure as an adversary can influence the distribution of keys to a non-uniform distribution. They show how an active attacker controlling a small number of parties can bias the values of generated keys. They propose a new protocol, called DKG, which they prove secure. They introduce in the beginning of the protocol an initial commitment phase where each node commits to its initial choice $a_{i0}$ in a way that prevents the attacker from later biasing the output distribution of the protocol. This commitment phase is called Pedersen Verifiable Secret Sharing or Pedersen-VSS [15]. The distributed protocol DKG performed by $n$ nodes generates $n$ private outputs called the shares, and a public output. It satisfies the following requirements of correctness:

1) Any subset of $t + 1$ shares provided by honest nodes define the same unique secret key $s$.
2) All honest nodes have the same value of public key.
3) The secret $s$ is uniformly distributed in $\mathbb{Z}_q$.

Moreover, no information on $s$ can be learnt by an adversary (except for what is implied by the public value). In the following, we describe the protocol of Gennaro et.al. in the context of Elliptic Curve Discrete Logarithm Problem (ECDLP). We call it ECDKG. This key distribution solution is adopted in [11] for ad-hoc networks. The value $s$ is the secret (the master key) and $P_{pub}$ is the corresponding public key of the distributed PKG.

Let $P \in \mathbb{G}$ and $P' \in (P)$ where $(P)$ denotes the subgroup generated by $P$. We assume that ECDLP is hard in $\mathbb{G}$.

B. ECDKG protocol

Generating $s$

1) Each node $N_i$ performs a Pedersen-VSS of a random value $z_i$:

a) node $N_i$ chooses two random polynomials $f_i(z), f'_i(z) \in \mathbb{F}_q[z]$ of degree $t$:

$$f_i(z) = \sum_{k=0}^{t} a_{ik} z^k, \quad f'_i(z) = \sum_{k=0}^{t} a'_{ik} z^k$$

Let $z_i = a_{i0}$, node $N_i$ broadcasts $C_{ik} = a_{ik} P + a'_{ik} P'$

where $k = 0, \ldots, t$. Remark that $C_{ik} \in \mathbb{G}$. Node $N_i$ computes the shares

$$f_{ij} = \sum_{k=0}^{t} a_{ik} j^k \quad \text{and} \quad f'_{ij} = \sum_{k=0}^{t} a'_{ik} j^k$$

for $j = 1, \ldots, n$ and sends $f_{ij}$ and $f'_{ij}$ to node $N_j$.

b) Each node $N_j$ verifies the shares he received from the other nodes by checking the equality

$$f_{ij} P + f'_{ij} P' = \sum_{k=0}^{t} (C_{ik}) j^k.$$
c) Each node $N_i$ who received a complaint from node $N_j$ broadcasts the values $f_{ij}$ and $f'_{ij}$ that satisfy the equality.

d) Each node marks as disqualified any node that either received more than $t$ complaints in step 1b or badly answered to a complaint in step 1c.

2) Let $Q$ be the set of qualified nodes. This set is built by all the nodes (it is shown in [8] that all honest nodes find the same set).

3) The private key is $s = \sum_{i \in Q} z_i$. And each node $N_i$ sets its share as

$$s_i = \sum_{j \in Q} f_{ji} \text{ and } s'_{i} = \sum_{j \in Q} f'_{ij}.$$ 

It is important to note that $s$ is never explicitly computed.

### Extracting the public key $P_{pub} = sP$

4) Each node $i \in Q$ broadcasts $A_{ik} = a_{ik}P$ for $k = 0, \ldots, t$.

5) Each node $N_i$ verifies the correctness of the received values by checking the following equality:

$$f_{ij}P = \sum_{k=0}^{t} (A_{ik})^j.$$ 

If the check fails for an index $i$, node $N_j$ complains against node $N_i$ by broadcasting the values $f_{ij}$ and $f'_{ij}$ that satisfy the first equality but not the second one.

6) For nodes $N_i$ who receive at least one such complaint, the other nodes run the reconstruction phase of Pedersen-VSS to effectively compute $z_i, f_i(z), A_{ik}$ for $k = 0, \ldots, t$. All nodes in $Q$ compute $P_{pub} = \sum_{i \in Q} A_{0i} \in G$.

### Private key construction

1) Node $N_i$ sends $Id_i$ to $t + 1$ nodes

2) each node $N_j$ sends $f(h_j), Q_i$ to node $N_i$

3) node $N_i$ computes $D_i = \sum_j \gamma_i f(h_j), Q_i$, where $\gamma_i$ denotes appropriate Lagrange interpolation coefficients.

The security of this protocol is analysed in [8].

C. Properties and improvements

In this protocol, there exist two types of nodes: master nodes and ordinary nodes. Only master nodes are able to deliver shares. If a node is unable to reach more than $t$ master nodes, it cannot obtain its private key. A node which obtained less than $t + 1$ shares could try to move in order to reach new master nodes and get the missing shares. But this solution is not satisfactory in practice and furthermore, during the evolution of the network, the total number of master nodes may decrease to less than $t$. It is then important to replace a master node which leaves the network. A trivial solution is that before leaving the network, any master node sends to an ordinary node of its choice its polynomial. Hence, the number of master nodes in the network is left unchanged. But this solution only applies when nodes are honest. Moreover, a master node may not be able (for example, it may be out of order) to transfer data to an ordinary node just before leaving the network.

### Protocol using a bivariate polynomial

In [5], Daza et.al. propose a more satisfactory solution. In their scheme, there exist three types of nodes: master nodes, parent nodes and ordinary nodes. Master and parent nodes both have the capacity to provide shares to construct private key. Moreover, master nodes can distribute special shares to an ordinary node so as it becomes a parent node. In other words, ordinary nodes can become parents if they request $t + 1$ master nodes. Their idea is to choose a bivariate polynomial instead of a monovariate polynomial. More precisely, each master node $N_i$ chooses a random symmetric bivariate polynomial $f(x, y) = \sum_{k,j=0}^{t} a_{k,j}x^k y^j$.

As usual, we have $f = \sum f_i$ and the share of node $N_j$ is $s_j = f(0, h_j) = S_j(0)$ (if we denote $S_j(x) = f(x, h_j)$). The second variable is only used to provide the special shares. The protocol is as follows:

1) Node $N_m$ selects a group of $t + 1$ master nodes
2) each master node $N_i$ from the group sends to node $N_m$ the value $f(h_i, h_m) = S_i(h_m) = S_m(h_i)$
3) node $N_m$ has enough shares to be able to construct $S_m(x)$ using Lagrange interpolation.
4) Its share of the secret is $s_m = S_m(0)$.

It is important to note that a parent node obtains a monovariate and not a bivariate polynomial. Hence, it can provide shares but not special shares. This scheme partly solves the problem: it increases the probability for an ordinary node to be connected to a sufficient number of parent or master nodes. On the other hand, if the number of master nodes becomes less than $t$, no new parent nodes can be created.

When evaluating the efficiency of this scheme, we must take into account the complexity of the protocol. During the initial phase, master nodes exchange polynomials instead of values from $Z_q$. Moreover, in their paper, Daza et.al. suppose that master nodes are honest which is not always realistic. In the case where master nodes are not all honest, a Pedersen-VSS commitment protocol should be used during the initialisation phase.

This protocol represents a good solution for a MANET where master nodes have strong communications capability and reliability but limited mobility. In that case, ordinary nodes may sometimes be too far from a master nodes to obtain shares and parent nodes may become essential. The initial phase is done once for all and the distribution of shares is similar to the one of the monovariate polynomial protocol.
IV. HIERARCHICAL THRESHOLD SECRET SHARING

Let us consider a network with two groups of nodes: leader nodes and ordinary nodes. Moreover, suppose that the policy of the network imposes the following rule: in order to obtain its private key, a node must request at least \( k \) other nodes of which \( k_1 < k \) are leader nodes. This configuration can be implemented by extending Shamir interpolation: instead of only considering shares as points of a polynomial \( f \), one may consider shares as points of some derivatives of \( f \). The secret can be recovered using Lagrange-Sylvester interpolation. In our example, the node must obtain \( t \geq k_1 \) points \( (i, f(i)) \) and \( k - l \) points of the derivative \( (i, f'(i)) \), where \( f \) is a polynomial of degree \( k-1 \). This example can be generalized to hierarchical schemes of \( t > 2 \) groups, but it is easy to see that parameters of the scheme (here \( t, k, t \)) must be carefully chosen since interpolation does not always have a solution. Lagrange-Sylvester interpolation is also called Birkhoff interpolation. It has been studied, for example in [19].

The use of hierarchical threshold schemes has been mentioned in [13], for a two level hierarchy ad-hoc network. The initial polynomial is bivariate and shares are monovariate polynomials. The initialisation phase is done off-line by a dealer. Nodes have to carry out heavy computations for verifications. The paper does not study in detail the initialisation phase and its security. In practice, for self-organized networks, the initialisation phase is even more costly in terms of communications and computations. The bivariate polynomial, which is in fact a two dimensional matrix should be obtained in a distributed way. In a given subset of nodes, each node should pick at random a matrix and send a polynomial to all nodes. Moreover, the scheme should take into account compromised nodes and include a commitment phase like Pedersen VSS during the initialisation phase. Therefore, hierarchical threshold schemes don’t seem to represent a realistic solution for MANETS as long as they don’t have strong communicational and computational capabilities.

However, hierarchical threshold schemes can be a suitable solution for other purposes. For example, they may be used to share the secret of the nuclear button. In that case, the secret could be recovered by the collaboration of \( k \) personalities like the President, at least \( k_1 \) generals and some colonels. Hierarchies can also become more complex using multivariate polynomials. These schemes are indeed very interesting but they do not provide a realistic solution for key distribution in MANETs.

V. CONCLUSION

In this paper, we showed that MANETs can be classified into those which have a central authority and those which don’t. Networks of the first group have access to an external trusted party during the initialisation phase. This TTP can send their private keys to all nodes. Networks of the second group are fully self-organized. However, within this group, there are two types of networks:

1) networks where any node has always access to at least \( t + 1 \) master nodes. In that case, Gennaro’s protocol is the most appropriate;
2) networks where some nodes may become isolated from master nodes. In this configuration, Daza’s protocol represents the best solution.

Regarding hierarchical schemes, we stressed that schemes based on Birkhoff interpolation are very costly in the eyes of the nature of MANETS. Indeed, MANETs often have limited computational and communication abilities while hierarchical schemes are very costly during the initialization phase. We believe that these schemes are not suitable to MANETs.

Finally, let us mention that we focused on threshold sharing schemes and we did not mention other solutions like the ones based on PGP (see for example [9]) which are now well known but difficult to compare to the schemes we considered.

REFERENCES


