Decentralized key generation scheme for cellular-based heterogeneous wireless ad hoc networks

Ananya Gupta, Anindo Mukherjee, Bin Xie, Dharma P. Agrawal *

Department of Computer Science, OBR Center of Distributed and Mobile Computing, University of Cincinnati, Cincinnati, OH 45221-0030, USA

Received 14 February 2006; received in revised form 15 May 2007; accepted 22 May 2007

Abstract

With the support of cellular system a cellular-based mobile ad hoc network (MANET) offers promising communication scenarios while entails secure data exchange as other wireless systems. In this paper, we propose a novel decentralized key generation mechanism using shared symmetric polynomials in which the base stations (BSs) carry out an initial key generation by a symmetric polynomial in a distributed manner and then pass on the key material to mobile stations (MSs). Thereafter, our proposed key generation scheme enables each pair of MSs to establish a pairwise key without any intervention from the BS, thus reducing the management cost for the BS. The shared key between two MSs is computed without any interaction between them. In addition, the trust among MSs is derived from the cellular infrastructure, thus enjoying an equal security level as provided in the underlying cellular network. Simulations are done to observe the system performance and the results are very encouraging.

© 2007 Elsevier Inc. All rights reserved.

Keywords: Ad hoc networks; Base station; Cellular networks; Distributed algorithm; Heterogeneous networks; Multi-interface mobile station; Pairwise keys; Polynomial; Symmetric key

1. Introduction

Group and peer-to-peer communication in mobile cellular-based ad hoc networks (MANET) has been of enormous interest in recent years [5]. A dual-mode mobile station (MS), such as the HP iPAQ h6315 [11], is equipped with two different interfaces so that it allows to simultaneously communicate with a variety of mediums (e.g., Bluetooth, Infrared, Wi-Fi) and infrastructure-based (cellular, access point) networks. In other words, such an MS has the ability to locally initiate an MANET with other MSs for multi-hop communication by employing the high-speed interface. Meanwhile, it can simultaneously access the cellular network [2] through the cellular base station (BS). In such a scenario, a group can be defined as a set of mobile nodes in the MANET that wish to communicate with each other over a secure channel. The MANET with infrastructure-support is achieved by a self-configuring networking protocol and provides many benefits [5]. For example, it allows flexible peer-to-peer communication between two MSs by utilizing a high-speed interface without passing through the BS, and thus releases the traffic load in cellular wireless systems. The localized MANET data transfer conserves the energy in data transmission if the transmission range can be reduced.

With aforementioned scenarios in mind, key generation has become a fundamental issue in supporting a secure MANET communication. It is well known that the traditional MANET (i.e., pure MANET) endures a serious problem for key generation as there are no prior trust relationships among ad hoc nodes due to absence of any centralized authority. In a mobile environment, it is difficult to identify an MS consistently with a unique identifier so that it is easy for an MS to change its identity, renew its key, and attack the MANET again. Therefore, it is hard to establish a trust relationship by key creation among MSs in a MANET. A malicious MS can use a forged MS identity to make feigned trust relations with other MSs, and then attack the MANET internally. On the contrary, integration of MANET with cellular network enables...
availability of a trustable infrastructure (i.e., BS) so that validation of MS's identity is feasible before any actual key generation. In order to establish trust relationship between any two MSs in the cellular-based MANET, it is important to take advantage of cellular infrastructure so as to enable a trustable and secure key generation before communication.

There are two types of key-based encryption techniques [1,12] that are widely used—asymmetric encryption using public–private keys and symmetric encryption with shared symmetric key. Although the former provides a high level of security and scalable key management, it results in high computation overhead, and even unrealistic to support bulk data transmission as each MS has limited computational capability. On the other hand, using a single key, shared among all ad hoc members for encrypting the group session key, may be unsecured since a single key can be easily disclosed or compromised. Alternatively, if the cellular network generates and maintains a shared key for each pair of MSs or other keys needed in the MANET, it poses heavy load on the key management in the BS, and causes frequent communication between the MS and the BS. As a result, the above key generating alternatives are not applicable for the cellular-based MANET communication.

It may be noted that a primary issue for cellular-based MANET scenarios is to establish a trust relation within the group. However, it involves several challenging steps. The support for MANET from a cellular network may involve multiple BSs because the MS may be associated with several BSs, which requires appropriate collaboration among BSs. The second issue is to maintain a secured channel between any pair of MSs in the MANET with minimal intervention of the BSs. To initiate a trust relation between two MSs, a secured channel for key distribution scheme spans through the wireless links as well as cellular infrastructure. Thirdly, scalability of key generation and distribution is involved in ad hoc group security. The bandwidth and energy required in key setup must also be reduced due to system constraints. For a group security, it is imperative to maintain appropriate security association between key generators, distributors, and receivers ensuring a certain level of trust between them. This problem has been commonly addressed in a way that logically segregates the key management/distribution entities and group memberships. In addition, another challenge facing the group key management infrastructure is that MANET members may join or leave at any time. In either case, the key has to be appropriately refreshed, as it would otherwise enable an old member to be able to access the group data, even when it is not a part of the group.

To address the aforementioned challenges, a decentralized key generation scheme is proposed in this paper, with the target of utilizing a cellular network for key management with less computational and communication overhead. The proposed scheme enables every pair of MSs, with the ability to calculate a shared symmetric key as required by using secure symmetric polynomial. In the proposed key generation scheme, the BS only distributes a piece of keying material (i.e., a polynomial) to each MS so that every pair of MSs can compute the shared key between them, rather than directly managing the key with an intensive interaction. The main contributions of the proposed key generation scheme are as follows:

- Instead of an asymmetric key generation scheme, the proposed key generation takes advantage of symmetric polynomials by which a shared symmetric key can be generated with minimal intervention of the BS.
- By using a cellular backbone for initial key setup and distribution, we employ the inherent security association and trust between MSs by using the cellular network. The trustworthiness of the third party (i.e., BS authority) is not achievable in a pure MANET.
- We are the first to employ a symmetric polynomial key generating scheme in a hierarchical and distributed manner for communication in a MANET.

This paper is organized as follows. In the next section, we provide a summary of related work and introduce the problem, and the background of polynomial in Section 3. In Section 4, we describe the main scheme of our distributed key generation, and present a method to reduce the communication overhead. In Section 5, we perform the security analysis and performance evaluation. Finally, Section 6 concludes the paper.

### 2. Related work

Key exchange using symmetric polynomials has always been an active area of interest over the last decade. Several schemes [3,4,9] have been suggested to exploit the symmetry in various kinds of polynomial. It was first applied by Blom [3] for a symmetric key distribution based on bivariate polynomials in a conference scenario. The bivariate polynomial is maintained by an authority and a partially evaluated polynomial is securely distributed to each conferencing user. Then, every conferencing user is able to compute a shared key with a peer user by using the peer identifier. Theoretical analysis with respect to its security robustness is provided by Blundo et al. [4,9]. The benefit of using a symmetric polynomial is that each user only maintains a single polynomial with which it allows to compute a common key for every other user non-interactively.

Recently, this idea has been used for pure MANETs and wireless sensor networks [6,8,13,14]. Kong et al. [3] introduce a polynomial scheme in which a distributed certificate is generated for each node in such a manner that a few nodes initialized with the key generation material send shares of the certificate to the requesting node. If the number of obtained shares is larger than a particular threshold, the requesting node is able to calculate the certificate. Deng et al. [8] enhanced the polynomial-based approach and presents an ID-based key generation scheme for a MANET. In this approach, a centralized server in a MANET distributes the shares for a master key, based on which the individual key share for a node can be calculated in a distributed manner by using its identifier.

In a wireless sensor network, the system (i.e., authority) initially distributes a polynomial share (a partially evaluated polynomial) to each sensor node before deployment of the sensor. The polynomial enables sensors to generate a shared key in a deployed network without any intervention again from the...
authority. Liu et al. [14] consider the fact that not all pairs of sensors have to generate shared keys in a wireless sensor network. Sensors are static and shared keys are only required for neighboring sensors. By using the shared key between two sensors, Liu et al. [15] further enable group key creation in a wireless sensor network. The security scalability [6] by using polynomial has been further improved based on the environment of a large-scale wireless sensor network. The purpose of Cheng et al. [6] is to reduce the storage requirement imposed by memory constraint of sensors.

These schemes are not applicable for the cellular-based MANET since all of them are not entirely distributed as they require the presence of a centralized server to initially distribute key generation material to nodes. In a pure MANET considered by Kong et al. [3] and Deng et al. [8], the infrastructure is not present and a mobile node acts as the authority by distributing polynomial for each pair of mobile nodes, which may be unreliable in terms of trustworthiness. On the other hand, in the wireless sensor network, the centralized authority pre-defines the polynomial for each sensor before the deployment of sensors. As discussed in Section 1, the cellular-based MANET, mobile users may be associated with different cellular BSs that may be charged by different authorities (i.e., service providers). Therefore, a key generation scheme needs to take into account the security collaboration between them. At the same time, the key generation in cellular-based MANET should consider the hierarchical architecture that the BS is on the infrastructure level and mobile users on the MANET level. The security association in the MANET level again should be inherited from the security association at the infrastructure level.

3. Problem formulation and polynomial

3.1. Cellular-based MANET and key requirement

Fig. 1 illustrates the cellular-based MANET with which the MS initiates MANET groups with the support of cellular BS (e.g., BS i and j as shown in Fig. 1), which is connected to the cellular backbone. The authenticated server on the cellular backbone maintains the information about the MS’s identifier (i.e., IP address), billing, account, and security credentials. This is equally applicable if BSs are replaced by Access Points (APs), served by the same Internet service provider (ISP), maintaining the credential of each participating MS. Each AP performs the same functionality as a BS so far as the group is concerned. Like a regular cellular network, each MS is able to securely communicate with its registered BS. The associated key between MS i and the BS is denoted by $S_{i-BS}$.

In Fig. 1, the owners of PDA3 and Smartphone2 wish to exchange confidential product drawings during a conference using ad hoc Wi-Fi technology. In such a scenario, one problem immediately apparent is that their transactions would be open to any uninvited prying MSs within the vicinity. Thus, a key generation scheme becomes necessary to enable secure peer-to-peer data transmission between two MSs in the same ad hoc group, which imposes the following basic requirements:

- **Forward secrecy**: A node joining the network should not be able to compute the peer or group keys that were used prior to its join.
- **Backward secrecy**: A node leaving the group should not be able to compute new keys that would be used in the network.
- **Key independence**: A node that is not a member of the group should not be able to derive any information about the group key from the knowledge of other group keys.
- **Group key secrecy**: It is computationally infeasible for an adversary to derive the group key.

As stated in Section 1, the asymmetric key (public and private key) involves high computation overhead in both encryption and decryption. Due to this, the need is to devise a symmetric key generation scheme which could inherit the trust of the cellular infrastructure. By using the key generation, any pair of MSs is able to compute a shared secret key between them before communication, and the group key can be further initiated in the MANET group. Trust relationships are such that no MS from another MANET group should be able to decipher any MS’s conversation in which it does not participate in. In the cellular-based MANET scenario considered in this paper, a MANET group corresponds to several MSs that wish to form a group for local multi-hop communication. These MSs may or may not be under the same BS or under the same cellular service provider. Further, the computation overhead needed for key generation should be minimized for the MSs.

3.2. Polynomial-based conference key

Polynomial-based dynamic conference key distribution [3] is an approach that an authorized server distributes a small piece of information to a set of conference users in such a way that each user can compute a shared key with every other user to use for secure communication. The theoretical analysis and its robustness of such an approach are further provided in [4] and [9]. In this section, we present this approach and its limitations in the scenario of cellular-based MANET.
Let us consider a symmetric key distribution scheme based on bivariate polynomials employed by a central server. At first, the server selects a polynomial $f(x, y)$ and keeps it secretly. The function $f$ satisfies the property $f(x, y) = f(y, x)$. The polynomial can be evaluated by $x = i$, where $i$ is the identity of the node. Instead of the original polynomial $f(x, y)$, the authorized server securely transmits the $f(i, y)$ to the corresponding node $i$ so that node $i$ has no knowledge of the original polynomial. Thus, whenever two nodes wish to communicate, they evaluate their individual polynomials with the corresponding IDs (i.e., IP address) of the other to get the symmetric key. For example, node $i$ having $f(i, y)$ and node $j$ having $f(j, y)$ would calculate $f(i, j)$ and $f(j, i)$, respectively. Due to symmetric property of the polynomial $f(x, y)$, these two quantities would be equal (i.e., $f(i, j) = f(j, i)$) and would serve as the symmetric key between these two nodes.

To illustrate the above scheme, we consider three conference users, having the identities of $1, 2,$ and $3$, respectively. In the beginning, the server selects a polynomial such as $f(x, y) = 1 + 2(x + y) + 3xy$, which satisfies $f(x, y) = f(y, x)$. The polynomial is kept secretly by the server. Then, the server calculates three polynomials by using the user IDs $1, 2,$ and $3$ as follows:

$$
f(1, y) = 1 + 2(1 + y) + 3 \cdot 1 \cdot y = 3 + 5y; \\
f(2, y) = 1 + 2(2 + y) + 3 \cdot 2 \cdot y = 5 + 8y; \\
f(3, y) = 1 + 2(3 + y) + 3 \cdot 3 \cdot y = 7 + 11y.
$$

The three polynomials (e.g., $f(1, y) = 3 + 5y$) are securely distributed to the corresponding user separately. When two of the three users initiate the communication, each node just evaluates its stored polynomial by using the ID of the other node to establish a pairwise key between them. For example, user $1$ computes $f(1, 2) = 3 + 5 \cdot 2 = 13$ and user $2$ calculates $f(2, 1) = 5 + 8 \cdot 1 = 13$. As can be seen, a shared key (i.e., $13$) is calculated by users $1$ and $2$ without any interaction between them. In the same fashion, users $1$ and $3$ can establish a shared key (i.e., $18$) between them. The third node cannot derive any information regarding the key between the other two users (e.g., $1$ and $2$) without the original polynomial. Since users calculate the shared key separately without the peer polynomial, then any user is unable to compute the shared key between other users. In other words, it satisfies the group key secrecy. The theoretical analysis in [4,9] shows that the generated keys have the property of key independence. For example, the keys $13$ and $18$ in the above example have no dependence between them. The robustness against brute-force attack increases with the increase of polynomial degree.

However, the above scheme is totally centralized without the consideration of the security requirements in a cellular-based MANET. In a cellular-based MANET, MANET members may be under the coverage of different BSs and an MS cannot securely communicate with the BS that is not attached. The MS can only securely communicate with the BS that it has registered with. On the other hand, it may be several local MANETs in the region of neighboring BSs. For example, there are three local MANETs around two BSs shown in Fig. 1. Therefore, the group key generation needs to be a decentralized process, which involves all BSs that MSs in a MANET are attached, rather than a single BS. The decentralized key generation scheme has the ability for the communicating MSs with appropriate key generating polynomials through the BSs of their respective cellular ISPs. In using a symmetric key, another underlying problem that has to be addressed is distributing the key and updating it securely and efficiently. Instead of depending on a centralized server or a single BS, multiple BSs have to contribute in generating or renewing keys.

For clarification, we first define the following terms that are used in developing our scheme.

- A node group (NG) is the group of MSs in a local MANET with the same polynomial distributors and derives its keying material from these leaders.
- An ad hoc node (AHN) is an MS that belongs to an NG.
- Polynomial distributor (PD) is a BS that acts as a polynomial supplier to an NG. A PD is a founder-PD if it is involved in the initial key generation process.

In the proposed key generation, the PD acts as a trusted authority for MSs and they collaboratively make decision on the key (i.e., the polynomial of each MS) and pass it on to their respective AHNs. As explained in the next section, our proposed key setup allows establishment of a pairwise key using only one polynomial between any pair of MSs through the cellular interface. In addition, the communication using this pairwise key can now independently take place securely over ad hoc Wi-Fi for securely distributing requisite group session encryption keys. The detail of the proposed scheme is given in the following section.

4. Decentralized key generation scheme

The principal objective of our scheme is to enable each MS to be able to securely communicate with any other MS. This should be possible without any prior communication between the MSs. In this section, we present the proposed decentralized key generation scheme.

In the beginning, a shared secret key $S_{i-BS}$ can be established between the MS $i$ and its corresponding BS by a standardized MS registration process [1]. Then, each PD selects a polynomial and securely sends to every other PD, after evaluating its ID using the polynomial. The original polynomial is kept secretly. Upon receiving the polynomial from every other PDs, the PD combines all polynomials to generate a new group-based polynomial. In this manner, every PD has its contribution on the group-based polynomial. Every PD has no knowledge of others. Then, each PD again evaluates the group-based polynomial by using the MS identifier and securely transmits to the corresponding MS. The MS has no knowledge of the group-based polynomial for the BS. When a pair of MSs evaluates the polynomial again by using its peer MS ID, both MSs obtain the shared secret key. Once a secure pairwise channel is setup between any two MSs, the group formation can be initiated without any further intervention by the BSs.
The distributed algorithmic for the above key generation includes four procedures.

- **Group-based polynomial selection:** At first, each MS registers with the PD and a set of multiple registered MSs using respective PDs to form a MANET with a set of NGs. After determining members in a MANET, each PD independently selects its symmetric polynomial (i.e., \( f_i(w, x, y, z) \)) and are exchanged between all the participating PDs so that a group-based polynomial is created at each PD. Such group-based polynomials are different at each PD.

- **Polynomial for MS:** Each PD again evaluates the group-based polynomial by using the identifiers of the MSs. Pairwise keying material (coefficients of the polynomial) at the PD is distributed to respective member MSs. Each MS obtains a unique polynomial.

- **Pairwise key generation:** Using this polynomial, each member MS can now independently generate a pairwise key by appropriately substituting remaining two variables for every other member belonging to an NG.

- **Group key establishment:** Note that the above generated keys are not the group session keys and are merely pairwise keys, which are then used to encrypt messages, including sending group session encryption keys, exchanged during the group setup process.

In the following subsections, we illustrate the above procedure in details.

### 4.1. Group-based polynomial

In the first stage of key generation, we employ a group-based polynomial distribution among the PDs.

As stated earlier, a localized MANET is supported by multiple BSs. Each of these BSs is a PD of an NG and each NG has a set of AHNs and a PD. This means that a localized MANET consists of MSs associated with multiple PDs that are involved in group-based polynomial distribution. We denote a participating group of MSs (AHNs) by \( NG_i \) and the polynomial distributor of \( NG_i \) by \( PD_i \) (\( 1 \leq i \leq n \)), where \( n \) is the number of PDs. The \( j \)th AHN belonging to a group \( i \) is denoted by \( AHN_{ji} \). The size of a group \( NG_i \) is denoted by \( |NG_i| \).

At the outset of this stage, each \( PD_i \) chooses a function \( f_i \) in four variables \( w, x, y, z \), such that:

\[
f_i(w, x, y, z) = f_i(x, w, y, z),
\]

and

\[
f_i(w, x, y, z) = f_i(w, x, z, y).
\]

The variables \( w \) and \( x \) represent the MSs and \( y \) and \( z \) denote the variables associated with PDs. The maximum degree of this polynomial is \( t \) in each variable. Each \( PD_i \) independently generates a \( t \)-degree symmetric polynomial as

\[
f_i(w, x, y, z) = \sum_{i,j,m,n=0}^{t} a_{i,j,m,n} w^j x^j y^m z^n,
\]

where each coefficient \( a_{i,j,m,n} \) \((0 \leq i, j, m, n \leq t)\) is randomly selected from a finite field \( GF(q) \), and \( q \) is a large prime number. The polynomial in Eq. (3) has four variables with \( t \)-degree while the example polynomials in Section 3.2 are bivariate (\( x \) and \( y \)) with 1-degree symmetric polynomial. The extension on the variable is for the purpose of representing the hierarchical architecture of the cellular-based MANET. The random selection of \( a_{i,j,m,n} \) ensures coefficients to be independent, without any correlation between them. It means that every number in the finite field \( GF(q) \) has the same chance to be selected. At the same time, the choice of the polynomial is entirely dependent on the PD. The robustness of the polynomial lies on the size of the coefficients and the degree of the polynomial. We note from (1) and (2) that the polynomials have to be symmetric in \( w, x \) and also in \( y, z \), which is ensured by Eq. (3). Thus, the coefficient of \( w^m x^n \) should be the same as that of \( x^m w^n \). In addition, the coefficient of \( y^m z^n \) should be the same as that of \( z^m y^n \).

After having chosen the polynomial \( f_i \), \( PD_i \) sends \( f_i(w, x, y, j) \) to \( PD_j \), which could be a part of another cellular service provider, as shown in Fig. 2. To prevent the polynomial from being compromised, this communication takes place over secured, pre-authenticated, backend cellular channels. Each \( PD_i \) now obtains the polynomial \( P_i \) as follows:

\[
P_i(w, x, y) = \sum_{k=1}^{n} f_k(w, x, y, i).
\]

In the above polynomial distribution, each PD provides its contribution for the final polynomial. Rather than the original polynomial \( f_i(w, x, y, z) \), the \( PD_i \) sends \( PD_k \) with polynomial \( f_i(w, x, y, k) \) after the evaluation by using the peer PD identifier \( k \). This ensures that each PD has no knowledge about the original polynomial and the polynomial associated with others. We can illustrate this by using a simple example as given in Section 3.2. After the evaluation of \( f(x, y) \) with \( x = 1, f(1, y) = 3 + 5y \) has no information about the original polynomial of \( f(x, y) = 1 + 2(x + y) + 3xy \). Meanwhile, \( f(1, y) = 3 + 5y \) has no information of the polynomial of \( f(2, y) = 5 + 8y \). Therefore, the generated keys on the BS will be resilient to the attack assuming that a different BS has been compromised. We will further analyze it in the next section.
4.2. Polynomial for MS

Before participating in an NG, each MS (i.e., an AHN) should have securely registered with the BS by following a standard registration process [1]. In the process of registration, the MS and BS mutually authenticate each other, e.g., the credential of the MS has been verified by the BS and the BS is trustable to the MS. For example, the server on the cellular backbone as shown in Fig. 1 can perform the mutual authentication between the BS and the MS. If the MS is roaming, the mutual authentication may involve the MS’s home server. For detail, please refer to the book [1]. As one of registration registration results, the BS generates a shared key such as \(S_{i-BS}\) for MS node \(i\). This shared key allows exchange of secure information between the BS and the MS. At the same time, once polynomials are selected by a group of PDs, the PD can provide pairwise keying material to the member AHN of corresponding NGs as follows. When a PD \(P_i\) obtains \(P_i\) as shown in Eq. (4), it further evaluates it at the ID of each of its group member AHN\(_{ki}\) as

\[
S_{ki}(x, y) = P_i(ID(AHN_{ki}), x, y, i).
\]

This quantity is now sent by \(P_i\) to AHN\(_{ki}\) as represented in Fig. 3. The transmitted keying material is the polynomial coefficients after the evaluation by using the AHN identifier. For example, instead of \(P_i(w, x, y)\), PD\(_i\) transmits the keying material of MS \(a\), which is the \(P_i(a, x, y)\). Therefore, each MS (i.e., AHN\(_{ki}\)) does not know the polynomial in the PD. From Eq. (5), it is known that the polynomial transmitted to AHN is bivariate since the variables \(w\) and \(z\) have been substituted by the identifier of the MS and its PD. Due to uniqueness of the MS identifier, every MS does not have the knowledge of the polynomial of other MSs. If an MS is compromised, the PD and other MSs are not affected due to the secret of their polynomials. The transmission between the PD and the AHN\(_{ki}\) is secured by using their shared key \(S_{i-BS}\). The PD sends every MS the ID-related polynomial so that each MS can compute the pairwise key with any other MS in the localized MANET.

4.3. Pairwise key generation

In order to calculate a pairwise symmetric key with any other nodes in the localized MANET, the node AHN\(_{ki}\) simply substitutes the ID of the other node for \(x\) and the ID of the other node’s PD for \(y\). We demonstrate the effectiveness of key establishment process by considering the following example. Let there be two PDs with IDs \(i\) and \(j\) and two MSs associated with these BSs with IDs \(a\) and \(b\). Thus, AHN \(a\) receives the following polynomial coefficients and constructs the polynomial:

\[
S_{ai}(x, y) = \sum_{k=1}^{n} f_k(a, x, y, i).
\]

Similarly, AHN \(b\) has the polynomial:

\[
S_{bj}(x, y) = \sum_{k=1}^{n} f_k(b, x, y, j).
\]

As illustrated in Fig. 4, if AHNs \(a\) and \(b\) wish to communicate with each other, node \(b\) calculates the key by using AHN \(a’\) ID and the corresponding PD (i.e., PD\(_i\)):

\[
S_{bj}(a, i) = \sum_{k=1}^{n} f_k(b, a, i, j).
\]

Similarly, AHN \(a\) calculates the key by using AHN \(b’\) ID and the corresponding PD (i.e., PD\(_j\)):

\[
S_{ai}(b, j) = \sum_{k=1}^{n} f_k(a, b, j, i).
\]

Using (1) and (2), we see that it has \(f_k(a, b, i, j) = f_k(b, a, j, i)\) for every \(k\), \(1 \leq k \leq n\). Thus, \(S_{bj}(a, i)\) in Eq. (7a) equals to \(S_{ai}(b, j)\) in Eq. (7b). It means that AHNs \(a\) and \(b\) would have the same symmetric key to communicate with each other. We can also see that the calculation of Eqs. (7a) and (7b) are achieved without any message exchanges between \(a\) and \(b\). We also see that the shared keys are independently computed among any pair of AHN in the MANET, without any further intervention from the PD after the distribution of key material. In addition, only one polynomial is maintained by an AHN, and uses it to calculate the key for every peer MS by substituting identifiers of the peer node and its related PD.

If a new AHN joins the MANET group, its PD can just distribute a polynomial to the AHN without the need of the group-based polynomial procedure as shown in Section 4.1. The new AHN can compute the key by following Figs. 3 and 4 while the other AHNs remain their previous polynomials. Such flexibility is provided by our hierarchical key generation.
architecture. The only exception is when the PD of the new MS is not included in the previous PD group. If an AHN leaves from the ad hoc group, the remaining AHNs are not affected. On the other hand, if the polynomial of the leaving AHN is regarded as a compromised one, the system can totally re-calculate new polynomials and redistribute by following the above steps, depending on the security requirement on the robustness (see Section 5).

4.4. Group key establishment

Let us consider two types of communication in the localized MANET. The peer-to-peer communication is between two MSs. In this communication type, each MS can securely communicate with any other MS using the pairwise keys established by the above steps. The symmetric key enables effectively data encryption/decryption, even for bulk data transmission by employing a high-speed interface. The other type is group communication in which the group key is further required. The group leader is defined as the initiator of the localized cellular-based MANET group. It can further generate the session encryption key, $K_{S0}$, for a defined duration. Once the MSs receive their individual keying materials, the group leader securely sends $K_{S0}$ over a high-bandwidth MANET to the other group members, instead of the cellular network. $K_{S0}$ is then used for session encryption to form a secure group channel between the members over any high-bandwidth local ad hoc interface.

In the event of a member leaving the secure group, the member generates a new session encryption key, $K_{S1}$, and sends it to each remaining member over the high-bandwidth localized domain. Alternatively, it can request complete polynomial re-keying from the PD for the entire group as discussed in Section 4.3. Likewise, a group join of an MS first involves the generation of key for the new member by its PD, following Eqs. (4)–(7b). Hereafter, the group creates a new session encrypting key and distributing it to every group participating member, including the newer joining entities.

Additionally, it is important to consider the security threat of a potential member authenticated by an impersonating BS, $B_S'$. Firstly, the forged $B_S'$ cannot be validated by the MS in the process of securely MS registration [1]. On the other hand, since such a forged $B_S'$ is outside the cellular infrastructure, it cannot participate in the group-based polynomial exchange as illustrated in Section 4.1. Therefore, it is unable to provide the necessary keying materials that would enable an MS to directly contact another BS until its secure channel could be established.

4.5. Efficiency improvement at PDs

We now consider how to select the polynomial by the PDs so that the communication overhead among PDs can be reduced. In most applications, the cellular-based MANET is a localized multi-hop network. Thus, the number of involving PDs should not be large. If there are $n$ PDs involved in the MANET, every PD sends a polynomial to every other PD, resulting in $(n - 1)$ polynomial transmission. It totally has $n * (n - 1)$ polynomials that should be exchanged among PDs. The exchange of polynomial is also localized because PDs are neighbors in a local region of the MANET. Furthermore, the exchange of polynomial passes through the hardwired cellular backbone, and thus the bandwidth should not be a major issue.

On the other hand, we can reduce the communication overhead by selecting sparse polynomial if the number of PD increases. Let $A_i$ be the coefficient matrix of $f_i(w, x, y, z)$ with dimensions $(t^2, r^2)$, such that the $(x, y)$th element of the coefficient matrix $A_i$ denotes the coefficient of $w^x x^y z^w$, where $l = x/t$, $n = y/t$. We further represent the coefficient matrix $A_i$ by using two matrices, $R$ and $Q$, and each has $t^2$ items.

Let $R$ be the matrix:

$$R = \begin{bmatrix} 1, w, w^2, \ldots, w^{t-1}, x, xw, \ldots, xw^{t-1}, \ldots, \\
x^{t-1}, x^{t-1}w, \ldots, x^{t-1}w^{t-1} \end{bmatrix}.$$  

And, let $Q$ be the matrix:

$$Q = \begin{bmatrix} 1, y, y^2, \ldots, y^{t-1}, z, zy, \ldots, zy^{t-1}, \ldots, \\
z^{t-1}, z^{t-1}y, \ldots, z^{t-1}y^{t-1} \end{bmatrix}.$$  

We first clarify the motivation of choosing $f$ to be a sparse polynomial. Instead of transmitting the entire coefficient matrix, now only an indexed array of the non-zero coefficients may be included in the transmission. Since this size is much smaller than the size of the entire coefficient matrix, the size of the message can be drastically reduced.

On the downside, employing sparse matrices can make an attacker (in most cases, a compromised MS) gain relative advantage by subjecting the system to a brute-force attack. However, we observe that with a relative large number of PDs, the final polynomial of the coefficient matrix by Eq. (4) would have a much-reduced sparseness. It is because each PD is independently and randomly selects its polynomial, and thus the degree of sparseness in the summarized polynomial is significantly decreased.

To verify this, let $s$ denote the probability of the element $(x, y)$ of the matrix $A_i$ to be non-zero. Thus, $1 - s$ is the corresponding probability of the element $(x, y)$ of the matrix, $A_i$, being zero. The probability $s'$ for the element $(x, y)$ of $\sum A_i$ $(1 \leq i \leq n)$, being non-zero is $s' = (1 - (1-s)^n)$. Here, $n$ is the number of PDs participating in polynomial exchange. This probability is plotted in Fig. 5. As can be seen, with an increase in the value of $n$, this $s'$ approaches 1. This result motivates us to use the polynomial selection strategy. If the number of PDs is small, it selects less sparse polynomial. On the other hand, if the number of PDs is large enough, the polynomials can be chosen to be sparse.
We present a technique to choose the coefficients in a manner, which is both secure and efficient. In order to reduce the sparseness of the matrix $A$, we do the following:

(i) In the matrix $Q$, we randomly choose exactly one term out of the $t$ terms $z^i (0 \leq i \leq t - 1)$, corresponding to each term of $y^j (0 \leq j \leq t - 1)$ with probability $\lambda_1$. Thus, we have a maximum of $t$ terms chosen out of $Q$.

(ii) In the matrix $R$, we randomly choose exactly one term out of the $t$ terms $w^i (0 \leq i \leq t - 1)$, corresponding to each term of $x^j (0 \leq j \leq t - 1)$ with probability $\lambda_2$. Thus we have a maximum of $t$ terms chosen out of $R$.

(iii) In the coefficient matrix $A$, we allow a term $a_{x, y}$ to be non-zero only if $x$ corresponds to a row chosen in Prop. (i) and $y$ corresponds to a column chosen in Prop. (ii).

By following the above steps, the polynomials are evaluated for each individual AHN and the polynomial $S$ is obtained by Eq. (5) would have all terms as non-zero if $\lambda_1$ and $\lambda_2$ are 1.

5. Security and performance analysis

5.1. Security analysis

As the security of the group formation mechanism depends solely on the pairwise key used to encrypt the group formation messages, we focus our analysis on the security of the pairwise keys. The security robustness of the pairwise keys primarily depends upon the inherent security of our key distribution scheme.

In the process of group-based polynomial exchanges among PDs, we employ a “$t$-secure” polynomial for each PD. In the other word, each PD generates a $t$-degree polynomial and the polynomials in Eqs. (4) and (5) are also $t$-degree since only linear operations are performed in each step. The “$t$-secure” means if more than $t$ polynomials are compromised, the adversary can calculate all the coefficients of the polynomial given in Eq. (3). Otherwise, the adversary cannot compute the key by substituting the identifiers of the MS and PD to obtain the key used by the corresponding MS. The reason is that the adversary does not know the single coefficients of $a_{i,j,m,n}$ in Eq. (3) since the PD has evaluated the polynomial by using the identifier of the PD or the MS as shown in Figs. 2 and 3. The adversary, who compromises an MS, only knows the linear combination of the polynomial. In fact, the exchanged polynomials essentially form a linear system consisting of linear equations like:

$$\sum_{i,j,m,n=0}^{t} a_{i,j,m,n} x^{i} y^{j} z^{m} = \text{key} \quad (0 \leq i, j, m, n \leq t). \quad (9)$$

When every coefficient $a_{i,j,m,n}$ is uniformly selected from $GP(q)$ while $q$ is a large prime number, the linear system has a unique solution over $GP(q)$. Indeed, the coefficient of the symmetric polynomial is any possible way to choose elements over $GP(q)$, and thus the key distributed to the MS is randomized. Therefore, the adversary has to know $(t + 1)$ polynomial $f_i$ in Eq. (3) to resolve the linear system for the purpose of determining the coefficient $a_{i,j,m,n}$. The knowledge of $t$ or less polynomials $f_i$ does not convey any information on another $f_j$.

In particular, such a polynomial system is further referred to as $t$ threshold system [3], and our scheme is a hierarchical $t$ threshold system. The adversary can combine the knowledge from different levels to resolving the linear system given by Eq. (9). In this hierarchical threshold system, the number of colluding nodes differs at each level. This is primarily because of the unevenness in the distribution of each tier. In what follows, we look into the number of nodes required at each level to carry out a successful collusion attack. We define compromise of the system as the situation where a node (AHN or PD) becomes aware of a polynomial $(f, P$ or $S$) which it should not know.

Lemma 1. Our scheme is secure to the collusion of a maximum of $(t - 1)$ nodes of any kind.

Proof. Eq. (3) has four variables totally with $t^4$ coefficients and will finally result in $t^4$ equations in the MS level. A $PD_i$ has substituted one variable (i.e., $z$) in the process of group-based polynomial distribution so that the peer $PD_j$ knows $t^3$ equations, which indicates it is $t$-secure in the PD level. Similarly, each AHN knows $t^2$ equations for the coefficients matrix $\sum A_i (0 \leq i \leq t - 1)$. Furthermore, if AHNs and PDs collude, the AHNs belonging to the same NG as a PD would contribute nothing to the attack.

We now consider collusion between PDs. Since each PD is aware of $t^3$ equations, a PD would require $t - 1$ other PDs to collude so as to get all the coefficients of the polynomial $f(w, x, y, z)$.

Next, we consider an attack by an AHN node. Since the polynomial $S(x, y)$ provided to an AHN has been evaluated at two points, collusion between AHNs would require $t^2$ nodes to find out the polynomial $f(w, x, y, z)$. If the AHNs of the same group wish to attack another node of the same group or
compromise their PD, collusion between at least $t$ nodes would be required.

Finally, we come to the case where $a$ PDs and $b$ AHNs collude. Since a PD contributes $t^3$ equations and an AHN contributes $t^2$ equations to solve for the coefficient matrix $\sum A_i$, this combination can compromise the polynomial $f(w, x, y, z)$ if $at + b > t^2$. Solving for the inequality shows that $a + b$ is always greater than or equal to $t$.

We therefore infer that under no conditions can collusion help if the number of colluding members is less than $t$. □

Table 1 illustrates various combinations of nodes required to launch a successful attack by collusion. It can be seen from the Table 1 that the increase of $t$ could improve the robustness of the security key. The maximal $t$ can be the number of PDs. If the $t$ is larger than the number of PDs, it is meaningless to increase the security robustness when all $r$ BSs are compromised. Meanwhile, the augment of $t$ also increases of the communication overhead since more coefficients should be transmitted.

Next, we consider a brute force attack. We see that there are $t^2$ unknowns in the polynomial $S(x, y)$. Thus, to carry out a brute-force attack, all these values have to be guessed. Assuming a field of size $\kappa$ for the polynomial coefficients, the attack complexity becomes $\kappa t^2$. For $\kappa = 2^8$, this value is beyond the reach of all modern computers, even for a small value of $t$.

If sparse matrices are chosen as mentioned in the last section, the attacker would have no relative advantage, as all possibilities would still be equally probable.

### 5.2. Performance analysis

The proposed decentralized key generation has two key benefits in terms of efficiency: (i) All the computation for the MS is linear combination without complicate calculation such as multiplicative inverse; (ii) it has no interaction between the MS in the key generation. The purpose of this analysis is to evaluate the scheme for the message and latency overhead. To keep the analysis generic and to incorporate scenarios involving WLAN APs as well as BSs, we have simulated the network over an ad hoc network running a routing protocol.

We carried out our simulation in ns-2 [10]. The number of nodes is varied from 30 to 70 and the number of groups is varied from 3 to 15. The simulation area was taken to be $1000 \times 1000$ m with a communication radius of each node as 200 m. We fixed the speed of the nodes at 10 m/s. Finally, we used DSR as the underlying routing protocol. Although in a practical scenario, members belonging to the same NG would be located next to each other, we have allowed the AHNs to be distributed over the entire network area irrespective of the NG they belong to. The reason for doing this is to observe the worst-case performance of our scheme.

We first observed the effects of the size of the NGs over the routing overhead. Theoretically, with small NG sizes, the number of routing messages should be high, as more number of PDs would need to communicate with each other. Fig. 6 plots the effects of the NG size to the overall network routing messages during the key exchange process. We notice that with an NG size of 3, the number of routing messages is very high. However, as the sizes increase, the routing overhead falls drastically and finally stabilizes. We attribute the reason for this to the fact that for smaller groups, the number of groups would be higher and thus, the initial $O(n^2)$ communication between the PDs would incur a very large routing overhead.

Next, we consider the effects of the threshold value over the latency in obtaining the key shares. A higher threshold value would lead to a higher number of coefficients in the polynomial and thus the message sizes would increase. As shown in Fig. 7, an increase in the threshold values $(t)$ leads to an increase in the latency values. However, the rise in the latency values is significant for a larger network. This is because in a smaller network, the messages do not need to travel a large number of hops and thus the delays are not significant.
6. Conclusions

In this work, we have proposed a novel method for a fully distributed key management and distribution technique in cellular-based MANET, assuming MSs to be heterogeneous by having a second radio for access to cellular BSs. The implicit trust for all communicating MSs is derived from the trust that an MS enjoys with the backbone cellular network. In our scheme, we distribute key material to each node over this network in such a manner that any two MSs can communicate securely with each other. Once this trust has been established, a group formation can be initiated, by any MS with other MSs, without further intervention from the BSs. Further, these MSs can then communicate with each other over any interface and need not require the cellular interface for communication.

In a practical scenario, MSs having Bluetooth interfaces can form secure localized MANETs and communicate with each other, while enjoying the same trust as it is provided in the underlying cellular network. Security analysis of our scheme shows that it is robust to the collusion of a fixed number of nodes. However, by keeping the threshold values of the chosen polynomials high, collusion attack probabilities can be reduced drastically.

Acknowledgment

This work has been supported in part by the Ohio Board of Regents, Doctoral Enhancements Funds.

References


Ananya Gupta received his M.S. degree in Computer Engineering under the direction of Prof. Dharma P. Agrawal from the University of Cincinnati. His research was primarily in the area of Wireless Security. He has a Bachelor’s of Engineering degree from University of Bombay in India. He is currently working with the Security Services Group in Deloitte & Touche, Detroit, USA.

Anindo Mukherjee obtained his B.E. from the Indian Institute of Technology, Roorkee, India in 2000 in and his Ph.D. from the University of Cincinnati in 2006 in the field of Computer Science under the guidance of Dr. Dharma P. Agrawal. He worked as a software engineer in Cisco Systems and Tejas Networks, India from September 2000 to December 2002. He was a student member of IEEE and has multiple publications in the field of wireless network security and multipath wireless networks analytical modeling. He is currently working as a software engineer in Google Inc., Mountain View, California, where he focuses on algorithmic and machine learning techniques for the elimination of web spam.
Bin Xie is a post-doctoral fellow at the Center for Distributed and Mobile Computing in the Department of CS, University of Cincinnati, OH. He received his B.Sc. degree from Central South University, Changsha, China, M.Sc. and Ph.D. degrees (with honors) in Computer Science and Computer Engineering from the University of Louisville (U of L), Kentucky, USA. His research interests are focused on ad hoc networks, sensor networks, wireless mesh networks, integrated WLAN/MANET/cellular with Internet, in particular, on the fundamental aspects of mobility management, performance evaluation, Internet/wireless infrastructure security, and wireless network capacity. In addition to his academic experience, he has six years of industry experience including ISDN, 3G, and Lucent Excel programmable switching systems.

Dharma P. Agrawal is the Ohio Board of Regents Distinguished Professor of Computer Science and the founding director for the Center for Distributed and Mobile Computing in the Department of CS, University of Cincinnati, OH. He has been a faculty member at the N.C. State University, Raleigh, NC (1982–1998), the Wayne State University, Detroit (1977–1982) and was on sabbatical leave at the ECE department, Carnegie Mellon University during Autumn 2007 and Winter 2007 quarters. His recent research interests include resource allocation and security in mesh networks, efficient query processing and security in sensor networks, and heterogeneous wireless networks. His recent contribution in the form of a co-authored introductory text book on Wireless and Mobile Computing has been widely accepted throughout the world, with economical edition reprinted in both India and China and translated in to Korean and Chinese languages and a second edition has just been published. Recently, he has co-authored another text-book on Ad hoc and Sensor Networks. He is an editor for the Journal of Parallel and Distributed Systems, International Journal on Distributed Sensor Networks, International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), and International Journal of Ad Hoc & Sensor Wireless Networks. He has served as an editor of the IEEE Computer magazine, the IEEE Transactions on Computers, and the International Journal of High Speed Computing. He has been the Program Chair and General Chair for numerous international conferences and meetings. He has received numerous certificates and awards from the IEEE Computer Society. He was awarded a "Third Millennium Medal," by the IEEE for his outstanding contributions. He has also delivered keynote speech for five international conferences. He also has five patents in wireless networking area. He has also been named as an ISI Highly Cited Researcher in Computer Science. He is a Fellow of the IEEE, the ACM, the AAAS, and the WIF.