Securing Multicast Groups in Ad Hoc Networks

Hartono Kurnio[1], Huaxiong Wang[1], Josef Pieprzyk[1], and Kris Gaj[2]

[1] Centre for Advanced Computing – Algorithms and Cryptography
Department of Computing
Macquarie University
Sydney, NSW 2109, AUSTRALIA
{hkurnio,hwang,josef}@ics.mq.edu.au

[2] Electrical and Computer Engineering
George Mason University
4400 University Drive, Fairfax, VA 22030
kgaj@gmu.edu

Abstract

Group communication in multicast environment has been a growing importance in ad hoc networks, and providing secure communication in the group is essential in many ad hoc applications. A common way of securing multicast group is to establish a cryptographic key known only to the group members. To maintain security, the group key must be updated whenever the group membership changes. A group key distribution scheme provides algorithms to establish and maintain the group key.

In this paper, we propose a reliable and ubiquitous group key distribution scheme that is suitable for ad hoc networks. The scheme has *self-initialisation* and *self-securing* features. The former feature allows a cooperation of an arbitrary number of nodes to initialise the system, and it also allows node admission to be performed in a decentralised fashion. The latter feature allows a group member to determine the group key remotely while maintaining the system security.

We also consider a decentralised solution of establishing secure point-to-point communication. The solution allows a new node to establish a secure channel with every existing node if it has pre-existing secure channels with a threshold number of the existing nodes.

1 Introduction

An ad hoc network allows a collection of wireless devices or nodes to communicate each others without relying on any fixed infrastructure such as base stations. A node can establish direct communication with other nodes that are within its transmission range. The networking functions depend on the cooperation of the wireless nodes, where packet transmissions between two distant nodes are forwarded by a chain of intermediate nodes that is reachable by both. A multicast

group consists of nodes communicating over a multicast channel. Multicasting is a popular media transmission for the group communication for its efficient mechanism of delivering packets from a source to a group of recipients. The sender only sends a copy of the message which will be replicated within the network and delivered to multiple recipients. The efficient bandwith requirement of multicast group is paramount in ad hoc networks that have limited resources. Using unicasting for group communication consumes a large amount of bandwith as the sender has to send an individual copy of the message to each user in the group. Note that several multicast routing mechanisms over ad hoc networks have been proposed in literature, see [16] for instance.

Wireless communications are usually implemented using broadcasting which is wide open to public access, and so multicast groups in ad hoc networks are prone to security attacks ranging from passive eavesdropping to active interfering. Securing group communication in ad hoc environments is important especially in military operations and rescue of hostages where communication among troops must be kept secret and authentic. Also, instantaneous conferences and classrooms need to ensure that only registered members can access the content.

A common way of securing multicast groups is to establish a cryptographic key known only to the group members. The group key will be used to provide secure and authentic communication within the group.[1] A multicasts group is dynamic and at any time, some nodes may leave or some new nodes may join the group; the compromised nodes must be evicted from the group. To maintain security, the group key must be updated whenever the group membership changes. A group key distribution scheme provides algorithms to establish and maintain the group key. Each group member is required to hold an individual set of secret keys. When updating a group key, a rekeying message is multicast to the system and members of the new group use this information and their individual secret key sets to determine the new common key, while nodes outside the new group cannot do so. Consequently, when a new node joins the group, it needs to obtain an individual key set to be able to participate in the group operation. Updating the group key for the same group is also neccessary to refresh the common key shared by the group members. Nevertheless, this operation seems to be irrelevant in ad hoc networks wherein membership constantly changes in a short period of time.

Securing group communication in ad hoc networks is a challenging task since it must take into account the characteristics of the environments. An ideal group key distribution scheme for ad hoc networks will have the following basic properties:

(1) **Secure.** Wireless network is open to eavesdropping attack and wireless devices typically have low physical security (easy to be either stolen or compromised). A group key distribution system has to consider these vulnerabilities in the security assessments. That is, the designer of the system has to assume that the adversaries obtain access to all communications transmitted through the networks, including multicast messages for rekeying, and they also have the knowledge of some portion of individual secret key sets in the system. The system must satisfy the requirement of at any time, only group members, and no other nodes, knowing the common key of the group. The

[1] Authenticity in this context is to ensure that the message is originated from a member of the group. Note that the authorship of the message may not be known to the other members.

security goal normally includes three main security goals required in ad hoc group communication. They are:

- **Session secrecy:** Ensures that a collusion of nodes that leave a group cannot discover the common key of the new group. We may view the leaving nodes as revoked nodes of the particular session. The revocation is on temporary basis where the revoked nodes might be members of any future groups. This security goal is required in numerous scenarios, for example, delegates of the same organisation would like to have private discussion in the middle of an ad hoc meeting by temporarily revoking delegates from other organisations.

- **Forward secrecy:** Ensures that a collusion of nodes that leaves a group cannot discover the common keys of the group for all future communications.[2] The leaving nodes are permanently revoked from the system. Such nodes include compromised/corrupted nodes, crash nodes or nodes that are out of mission in a battlefield scenario and they must not access the communication during the rest of the system lifetime.

- **Backward secrecy:** Ensures that a collusion of new nodes that join a group cannot discover the keys used by the group in the past. Joining the group allows the new nodes to participate in any current and future groups. This security goal is required in ad hoc admission where a node joins only for a certain time interval.[3]

(2) **Decentralised.** Ad hoc networks usually involve mobile nodes that are subject to frequent changes of network topology. This dynamic behaviour is also influenced by group membership changes, node failures and channel errors. The group key distribution system should be able to function under these conditions. Obviously, relying on a single entity (trusted authority) to perform security operations is not a wise solution since it has drawbacks such as (i) a single point of failure where the entity could be unavailable (because of congestion or DoS attacks such as *Smurf* at network layer and *SYN flooding* at transport layer) or unreachable by other nodes, and (ii) a single point of attack where compromising the entity means revealing all the system secrets. A decentralised model alleviates these problems by distributing the trust to all nodes in the network, and contributions from a number of nodes are required to perform security operations. This model allows the system to provide continuous and ubiquitous security services for mobile nodes despite the network condition. Also, the system is more resistant to attack where a single compromised node will not bring the entire system down.

(3) **Efficient.** A group key distribution system in ad hoc networks must require low amount of communication, computation and secure storage to perform security operations. These requirements are to accommodate the limited bandwidth in ad hoc networks and the limited energy and secure memory of mobile devices. A more efficient system implies a smaller response time, that is,

[2]The leaving nodes might know all previous group keys.

[3]Some scenarios might allow the new joined users to discover all previous communication, so the backward secrecy does not apply to them.

the time elapsed between starting and completing a security operation. A smaller response time is important in ad hoc group applications since membership changes frequently and in short period of time, requiring fast key updates.

(4) **Scalable.** Another desired property of group key distribution systems is scalability for large groups *while preserving efficiency of the systems*. For example, in a battlefield scenario the system must work well for a small number of troop as well as for large numbers with hundreds or thousands of army troops.

In this paper we focus on the communication protocols for dynamic multicast groups in ad hoc networks, and assume the underlying wireless networks support multicast channel. We propose a group key distribution scheme that satisfies all of the basic properties described above. Our construction is based on Key Distribution Patters (KDP) [14], and it has some desirable features such as self-initialisation and self-securing in which no trusted authority is required in system initialisation and key update, respectively.[4] The former feature allows a cooperation of an arbitrary number of nodes to initialise the system, and new nodes can join the initial group where their individual secret key sets are given in a distributed fashion. The latter feature allows a group member to determine the group key remotely while maintaining the system security. This results in a reliable and ubiquitous key update service.

We also consider a decentralised solution of establishing secure point-to-point communication. The solution allows a new node to establish a secure channel with every existing node if it has pre-existing secure channels with a threshold number of the existing nodes. We use the idea of threshold cryptography and employ Blom key distribution scheme [3] in the construction.

Organisation of this paper. We first discuss several existing group key distribution systems and show their limitations in ad hoc environments in Section 2. Then we describe our proposed system that is suitable for ad hoc networks in Section 3, and give the construction in Section 4. We conclude this paper in Section 5.

2 Related Work and Drawbacks

Group key distribution schemes have been a popular research area and many schemes have been proposed, see for instance [23, 24, 5, 12, 4, 19, 20, 2]. However, most of them consider traditional model of networking requiring wired and fixed infrastructure that is not suitable for ad hoc networks.

The schemes in [23, 24, 5, 12, 17] employ key hierarchy to implement an efficient key establishment for dynamic groups in multicast environments. In these schemes, each node in the tree corresponds to a key and each leaf corresponds to a user. A user's individual secret key set comprises of all keys along the path from his corresponding leaf to the root, and the individual secret key set has to be updated every time the group changes. Each scheme proposes different key generation and rekeying methods with the aim of reducing communication and storage costs. The authors

[4] The terms *self-initialisation* and *self-securing* are also used in [11] and [13], respectively, but in different contexts.

of [15, 7] proposed revocation schemes in the context of broadcast encryption. They utilise binary tree structure to define a collection of user subsets and use a pseudo-random sequence generator (whose output length is three times the length of the input) to assign keys to the subsets. A user knows the keys corresponding to the subsets that contain him, but he only needs to store some fraction of the keys. In the revocation schemes, a sub-collection of disjoint subsets that partition all authorised users, but not contain any of the revoked users, needs to be found and the new group key is encrypted using the keys corresponding to the subsets. (Our proposed scheme will follow this basic idea but using a different construction to have decentralised property.) Variants of the schemes can be found in [1] where a higher computation cost is required. Although these solutions are secure, efficient and scalable for large groups, they are not suitable for ad hoc networks because of the requirement for a fixed trusted authority to perform all group operations.

The schemes in [8, 4, 19, 20] extend the two party Diffie-Hellman key exchange for multiple parties. The schemes are decentralised and require all parties to contribute for the initial group key establishment. The schemes in [8, 4] do not consider dynamic group membership and the update of group key is essentially equivalent to re-setup the system. The schemes in [19, 20] provide efficient protocols to update the group key. The work in [6] attempt to adapt the schemes in ad hoc networks. Nevertheless, they still involve excessive communication cost (high number of transmissions) and computation cost (high number of exponentiations) in the system initialisation that are unbearable in most ad hoc applications.

The work in [25] and its extensions [9, 10, 11] consider security in ad hoc networks. Their approach is based on public key infrastructure (PKI) whereby any two entities might establish a secure and authentic channel using certificates carried by each of them. The schemes employ a t-out-of-n threshold secret sharing mechanism to distribute the certification authority function to n entities, in which each entity holds a secret share, and any t entities can collaboratively sign a public key. The scheme in [25] considers distribution of the function to n *special nodes* but it has a drawback that the special nodes may be multi-hop away or may move, so a reliable certification function is not guaranteed. The schemes in [10, 11] extend [25] to provide effective and ubiquitous certification service by enabling any t *local nodes* (one-hop away) to collaboratively perform the function. The schemes periodically update the secret shares of all nodes to resist intrusions for a long term by using the idea of proactive secret sharing. Observe that the PKI approach is suitable for secure point-to-point communication, but not for secure group communication (one-to-many, many-to-many, many-to-one). If the approach is used for group application, the sender needs to send a copy of the message to each node encrypted using the node's public key. This clearly results in high communication and computation costs. Therefore, the schemes, although have secure and decentralised solutions, they are only efficient for small groups.

3 Our Proposed System

We propose a group key distribution system using Key Distribution Patterns (KDP). We consider a system of wireless ad hoc network with an upper bounded number of nodes, say n. Each node

in the system has limited resource and it communicates with others via the bandwidth-constraint, error-prone, and insecure multicast channel. They may freely roam in the networks. Nodes in the system may form a secure group where all communication within the group is encrypted (symmetric encryption) using a group key. The number of nodes in the group may change over time because nodes temporarily or permanently leave, or new nodes join. A new group will have a new common key. Our security model tolerates a collusion of up to t adversaries to maintain session secrecy, forward secrecy or backward secrecy.

We assume each node P in the system has a globally unique identifier i, denoted by P_i. We also assume each node has a channel with other nodes for secure point-to-point communication. The secure channel can be realised by using a public-key based cryptosystem that is suitable for ad hoc networks such as [9], or as an alternative, private-key encryption in Section 4.3 can be employed for the purpose. We emphasize that secure point-to-point communication in our system is only for secure delivery of key information. Some desirable features of our system are as follows.

- *Self-initialisation* – It does not require a trusted authority to set up the system. Instead, a cooperation of ℓ, $\ell \leq n$, nodes can initilise the system and form a multicast group. First, they decide on mutually acceptable parameters of the underlying construction, i.e., key distribution patterns, that include the values of n and t.[5] Next, they cooperatively generate all system secrets and assist new nodes to join. When a new node joins the group, it obtains an individual secret key set from a subset of the existing nodes in the group, called sponsors. The new joined node may be sponsor for other new nodes. Note that this feature is desired in ad hoc networks where a group is usually formed in an ad hoc manner and node admission happens on the fly.

- *Self-securing* – Members of a new group can determine the common key by finding an appropriate combination of their secret keys, and no rekeying message is required in the key update. This implies a reliable and ubiquitous key update service since there is no transmission through unreliable wireless networks and all group members can compute the group key anywhere, anytime regarless the network topology. The key update method is also fault-tolerant as several faulty group members (node failures) will not affect the group key establishment at all (other group members still can determine the group key as if there is no faulty group members). The key update method is simple and very efficient since it requires (i) zero communication cost, and (ii) a group member to compute some elementary operations and to store a reasonable number of secret keys.

Distributed Establishment of Secure Channels

In ad hoc networks, the assumption that every pair of entities in the system can establish a secure channel using public key cryptosystem is not realistic, since it requires every entity in the system to hold a public-key certificate which is verifiable by all other entities. It is more realistic to assume that some, but not all, pre-existing point-to-point secure channels do exist.

[5] We do not consider the negotiation process in any detail and assume the initial nodes can reach a valid agreement.

We consider a scenario where any $t+1$ nodes can collaboratively establish secure channels between a new node with other nodes provided that the new node has pre-existing secure channels with the $t+1$ nodes. The channel between two nodes is secure against any collusion of at most t other nodes (this assumption forces an adversary to compromise at most t nodes before she can listen to the private communication of the group).

The secure channels are based on private key setting where a symmetric encryption is used for secure communication, and the system is an efficient alternative to the system with public-key encryption. This is because private-key encryption requires less computation power compared to the public-key one. Also in a public-key encryption, allowing a node to have secure channels with all other nodes requires the node to store all public keys in the system, which consumes a large volume of memory.

The threshold t defines a tradeoff between service availability and security. The new node needs to have at least $t+1$ pre-existing secure channels, which may not be fulfilled because of dynamic topology in ad hoc network. Smaller value t requires smaller number of collaborations to provide the service, but it has lower intrusion tolerance.

4 Construction

In this section we describe our architecture for the proposed group key distribution system. First we give some cryptographic primitives that will be employed in the construction. Later we will completely describe the protocols for the system.

4.1 Background

4.1.1 Key Distribution Patterns (KDPs)

Key distribution patterns (KDP) [14] are finite incidence structures that were originally designed to distribute keys between pairs of participants in a network in the absence of an online key distribution centre. A KDP is used to allocate a collection of subkeys to users in a system in such a way that any pair of users can compute a common key by finding an appropriate combinations of their subkeys. In general, a KDP can be defined as follows.

Definition 4.1 *Let $K = \{k_1, \ldots, k_v\}$ be a v-set and $\mathcal{B} = \{B_1, \ldots, B_n\}$ be a family of subsets of K. We call the set system $(K, \mathcal{B})$ a t-resilient (v, n, r) key distribution pattern (KDP) if the following condition holds*

$$\bigcap_{i \in \Delta} B_i \not\subseteq \bigcup_{j \in \Lambda} B_j \ ,$$

where Δ and Λ are any disjoint subsets of $\{1, \ldots, n\}$ such that $|\Delta| = r$ and $|\Lambda| = t$.

The KDP guarantees that for any r subsets, $\{B_{i_1}, \ldots, B_{i_r}\}$ and any t subsets, $\{B_{j_1}, \ldots, B_{j_t}\}$, where $\{B_{i_1}, \ldots, B_{i_r}\} \cap \{B_{j_1}, \ldots, B_{j_t}\} = \emptyset$, there is exist at least an element k that belongs to the r

subsets, but does not belong to the t subsets. This means, for a given r subsets or less, an arbitrary union of at most t other subsets cannot cover elements in the r subsets.

Observe that KDP with $|\Delta| = 2$ is considered in [14]. KDP with more general size of Δ have been studied by Stinson et al (see for example [21, 22]). We give an example of 1-resilient $(9, 12, 2)$-KDP.

Example 4.1 *Let* $K = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ *and* $\mathcal{B} = \{B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}, B_{11}, B_{12}\}$ *defined as follows.*

$$\begin{array}{ll} B_1 = \{4, 5, 6, 7, 8, 9\} & B_2 = \{2, 3, 5, 6, 8, 9\} \\ B_3 = \{2, 3, 4, 6, 7, 8\} & B_4 = \{2, 3, 4, 5, 7, 9\} \\ B_5 = \{1, 2, 3, 7, 8, 9\} & B_6 = \{1, 3, 4, 6, 7, 9\} \\ B_7 = \{1, 3, 4, 5, 8, 9\} & B_8 = \{1, 3, 5, 6, 7, 8\} \\ B_9 = \{1, 2, 3, 4, 5, 6\} & B_{10} = \{1, 2, 4, 5, 7, 8\} \\ B_{11} = \{1, 2, 5, 6, 7, 9\} & B_{12} = \{1, 2, 4, 6, 8, 9\}. \end{array}$$

Then $(K, \mathcal{B})$ *is a* 1*-resilient* $(9, 12, 2)$*-KDP.*

Rationale. A KDP can be used to provide secure group communication among nodes. We may consider K is a set of keys and $\mathcal{B}$ is a collection of subsets of the keys. The KDP structure is public while keys of the KDP are secret. Therefore, given a t-resilient (v, n, r)-KDP, if we can distribute keys $K = \{k_1, \ldots, k_v\}$ to n nodes in such a way that each node P_i has a subset of keys B_i, then any r-subset $G = \{P_{i_1}, \ldots, P_{i_r}\}$ can come up with a common key without any interaction, that is

$$k_G = \bigoplus_{k \in \cap_{j=1}^{r} B_{i_j}} k \ ,$$

where $\oplus$ denotes the exclusive-or operation (assuming that all the keys are strings of the same length). The common key k_G can be used to provide secure communication among nodes in G. For example, a node can send the ciphertext $E_{k_G}(m)$ of the plaintext m that can only be decrypted by other nodes in G. Any collusion of at most t nodes outside G cannot decrypt the plaintext m.

Observe that v corresponds to the number of the keys in the whole system and naturally bounds the number of keys for each node P_i (the required secure memory to store keys in B_i). The trivial construction is to assign a key k to each d-subset of the n nodes, for all $d \leq r$, and every node obtains keys of subsets to which the node belongs. The construction is resilient to any number of colluders, however, observe that it has $v = \sum_{d=0}^{r} \binom{n}{d} = O(2^n)$ and every node has to store a huge number of keys. This suggests a tradeoff between the collusion size and the number of system keys. Note that the restriction on the collusion size is reasonable since in many ad hoc group applications, it is unlikely that all nodes outside G will be the colluders.

An efficient KDP would be the one in which for given t, r, n the value v is as small as possible. An efficient KDP can achieve $v = O(\log n)$ that is shown in [22] as follows.

$$v \geq 2c \frac{\binom{r+t}{r}}{\log(r+t)} \log n \ ,$$

where the constant c is shown to be approximately $\frac{1}{8}$ in [18]. For example, given $t = 10, r = 5, n = 100$, and $c = \frac{1}{8}$, it gives $v \geq 1277$ keys. This number is very small compared to the trivial construction that gives $v = 79375495$ keys.

A lot of research has been done for efficient KDP construction with bounds on various parameters. An explicit construction that is asymptotically optimal is given in [21]. It uses perfect hash families or separating hash families to construct a t-resilient (v, n, r)-KDP with $v = O(\log n)$ while the parameters r and t are fixed. In practice, a KDP must be designed with respect to the nature of group applications. For example, military applications require high collusion resistance in which case the underlying KDP will have a large t value.

The KDP structure can be represented as a $n \times v$ table $T = [e_{i,j}]$, where row i is indexed by $B_i \in \mathcal{B}$ and column j is indexed by $k_j \in K$, and where each entry $e_{i,j}$ is either 0 or 1 such that $e_{ij} = 1$ if and only if $k_j \in B_i$. The entries can be represented as bits (there are $n \times v$ bits) and they are stored by each node.

4.1.2 Blom's Key Distribution Scheme

Blom's scheme [3] allows any pair of users in the network to compute a secret key without any interaction. The scheme is said to be t unconditional secure if, given a specified pair of users, any collusion of t or fewer users (disjoint from the two) cannot compute the secret key shared by the two. The scheme uses a symmetric polynomial $F(x, y)$ over a finite field $GF(q)$ in the construction,

$$F(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{t} h_{i,j} x^i y^j \bmod p \ ,$$

where $h_{i,j} \in GF(q)(0 \leq i \leq t, 0 \leq j \leq t)$, and $h_{i,j} = h_{j,i}$ for all i, j. The polynomial $F(x, y)$ has a symmetry property, that is, $F(a, b) = F(b, a)$ for any $a, b \in GF(q)$. We give an example of symmetric polymomial with 1 unconditional secure.

Example 4.2 *Given* $F(x, y) = 8 + 7(x + y) + 2xy \bmod 17$*, observe that* $F(a, b) = F(b, a)$ *for any* $a, b \in GF(17)$*.*

Blom's scheme requires a trusted authority to distribute the keys to the users. In this paper, we slighly modify the scheme to support decentralised setting.

Rationale. The scheme can be employed to provide secure communication for a pair of nodes. Given a symmetric polynomial $F(x, y)$, if we can distribute the polynomial $F(x, i)$ to every node P_i in the system, then any two nodes P_{i_1}, P_{i_2} can establish a secret key $F(i_1, i_2) = F(i_2, i_1)$, which can be used for secure communication. For example, P_{i_1} can send a ciphertext $E_{F(i_1,i_2)}(m)$ that can only be decrypted by P_{i_2}. Any collusion of at most t other nodes cannot learn the plaintext m.

The threshold t gives the balance between storage requirement and collusion resistance. An extreme case is to allow an arbitrary number of colluder that requires each pair of nodes to be independently assigned a secret key. This requires a total of $\binom{n}{2} \approx \frac{1}{2}n^2$ secret keys and each node

must store $n-1$ secret keys, which is unbearable if n is large. By using Blom's scheme, a node needs to store a polynomial of degree at most t, whereas at least $t+1$ nodes must collude (an adversary must break in at least $t+1$ nodes) to discover the system secret $F(x,y)$. (In fact, t corrupted nodes will not reveal the secret key between any pair of other nodes.)

4.2 The Group Key Distribution Scheme

We assume each node has a secure channel with every other nodes in the system. If PKI is used to establish the secure channels, each node has to store certificates of all other nodes. As an alternative, the solution in Section 4.3 can be applied for the purpose in which each node is required to store a polynomial of degree at most t. A secure channel is for secure transmission of key information. We assume that key information can be successfully delivered to the recipient (even the recipient is multi-hop away from the source). This is reasonable since the recipient can always move to a new location to maintain the connection with the source.

The straightforward scheme to establish a common key for a group of nodes is as follows. Each time a new group is created, a selected node randomly generates a group key and sends it to all other nodes in the group through secure channels. The scheme has minimum secure memory requirement as each node holds just key information for the secure channel, and it is secure against an arbitrary number of colluders. However, the scheme requires large amount of communication as the number of transmissions is equal to the number of nodes in the group. Note that ad hoc group applications require frequent group changes, and consequently, the total communication overhead required to establish all group keys will be prohibitively high.

Our proposed scheme has several advantages over the straightforward scheme: (i) one time system initialisation and occasional key refreshing that requires communication over secure channels to all nodes in the new created group and (ii) zero communication overhead required to establish a group key. The proposed scheme requires each node to store a reasonable number of keys and it is secure against a collusion of at most t adversaries.

4.2.1 Initialisation

Suppose initially there are ℓ nodes, $\{P_{i_1}, \ldots, P_{i_\ell}\}$, that form the ad hoc network. They agree on values for parameters (t, v, n, r) of a KDP satisfying $n \geq \ell$, and establish the $(K, \mathcal{B})$-KDP using some efficient construction. Every P_i corresponds to a block $B_i \in \mathcal{B}$. Next, the question is: how to distribute keys $K = \{k_1, \ldots, k_v\}$ to the nodes with respect to the KDP structure? It is easy if there is a single trusted server that will generate the v keys and distribute them to each node according to the underlying KDP. However, as we discussed before, the centralised setting is not suitable for ad hoc network. We would like to have a decentralised fashion for key distribution. That is, generation and distribution of the v keys are cooperatively performed by the ℓ nodes.

We assume that the union of all blocks $\{B_{i_1}, \ldots, B_{i_\ell}\}$ corresponding to the ℓ nodes will cover K, that is, $B_{i_1} \cup \ldots \cup B_{i_\ell} = K$ (this can be easily achieved when designing the underlying KDP). The generation and distribution of the v keys is shown in Table 1. Recall that the KDP structure

is public.

Table 1: Generation and distribution of system keys

$K' = K$, $z = \ell$
while $K' \neq \emptyset$ {
    P_{i_z} randomly generates keys for $B_{i_z} \cap K'$
    for $1 \leq u \leq z-1$ {
        P_{i_z} sends to P_{i_u} the keys for $B_{i_z} \cap B_{i_u}$ over a secure channel
    }
    $K' = K' \setminus B_{i_z}$
    $z = z - 1$
}

4.2.2 Group Key Update

We consider three events that require key update: temporarily leaving nodes, permanently leaving nodes, and new joining nodes.

Temporarily leaving nodes
Suppose nodes $\{P_{j_1}, \ldots, P_{j_{t'}}\}, t' \leq t$, temporarily leave the group, and a node P_i of the remaining nodes $\{P_{i_1}, \ldots, P_{i_{r'}}\}$ wishes to securely send a message m to the the rest of nodes in the subgroup. Group key establishment is as follows, referring to the KDP description in Section 4.1.1.

1. If $r' \leq r$, clearly nodes in $\{P_{i_1}, \ldots, P_{i_{r'}}\}$ can establish the group key k_G that is known only by them. The node P_i uses the group key to encrypt the message m and multicasts the result to the subgroup.

2. If $r' > r$, the node P_i partitions $\{P_{i_1}, \ldots, P_{i_{r'}}\} \setminus \{P_i\}$ into subsets, each is of size at most $r-1$, and adds itself as a member to each subset. Subsequently, the node P_i establishes the group key k_G with each of the subsets and encrypts the message m using the group key. The node P_i multicasts the ω encrypted messages to the subgroup, where ω is the number of partitions.

It is straightforward to see that the protocol satisfies session secrecy.

Permanently leaving nodes
Suppose nodes $\{P_{j_1}, \ldots, P_{j_{t'}}\}, t' \leq t$, belonging to the group $\{P_1, \ldots, P_\ell\}$ permanently leave the system, and the remaining nodes are $\{P_{i_1}, \ldots, P_{i_{r'}}\}$. Secure communication within the new group can be done in the same way as temporarily leaving nodes. The permanently leaving nodes will not exist in all future groups, and so they naturally can be viewed as the nodes that will also leave all future groups. This requires the permanently leaving nodes to be consistently included in the

set of temporarily leaving nodes in all future groups. Obviously, the permanently leaving nodes cannot find either the new group key or all future group keys. Nevertheless, this trivial method is not scalable since the total number of temporarily leaving nodes will grow over time, and they must not exceed t nodes.

We give a scalable method that requires some system keys to be replaced by fresh keys in order to maintain forward secrecy. That is, the keys in blocks $\{B_{j_1}, \ldots, B_{j_{t'}}\}$ that belong to nodes $\{P_{j_1}, \ldots, P_{j_{t'}}\}$ have to be refreshed in such a way that the permanently leaving nodes (even they collude) cannot learn the fresh keys. The fresh keys are collaboratively generated by nodes $\{P_{i_1}, \ldots, P_{i_{r'}}\}$ whose key blocks intersect with keys $B_{j_1} \cup \ldots \cup B_{j_{t'}}$, in the following way (see Table 2).

Table 2: Refreshing system keys

$K' = (B_{i_1} \cup \ldots \cup B_{i_{r'}}) \cap (B_{j_1} \cup \ldots \cup B_{j_{t'}})$, $z = r'$
while $K' \neq \emptyset$ {
    P_{i_z} randomly generates fresh keys for $B_{i_z} \cap K'$
    for $1 \leq u \leq z-1$ {
        P_{i_z} sends to P_{i_u} the fresh keys for $B_{i_z} \cap B_{i_u}$ over a secure channel
    }
    $K' = K' \setminus B_{i_z}$
    $z = z - 1$
}
for $1 \leq u \leq r'$ {
    P_{i_u} replaces the keys in $B_{i_u} \cap (B_{j_1} \cup \ldots \cup B_{j_{t'}})$ with the fresh keys
}

Observe that this method allows an arbitrary number of nodes to permanently leave the system. After key refreshment, the nodes $\{P_{i_1}, \ldots, P_{i_{r'}}\}$ can have secure communication by following the protocol of temporarily leaving nodes with none leaves.

None of the fresh keys is sent to the permanently leaving nodes so a collusion of them cannot learn the fresh keys. It is not necessary to include them in the set of temporarily leaving nodes in all future groups. Forward secrecy is satisfied because they hold outdated key information which is useless to find the new group key and all future group keys.

This method can be used in the conjunction with the trivial method to reduce the number of key refreshing operations. That is, the trivial method is normally used to update the group key, and the system keys are only refreshed each time the total number of temporarily leaving nodes reaches t nodes. Note that in a key refreshing operation, the permanently leaving nodes are those nodes that have permanently left the group since the previous key refreshment operation.

New joining nodes

Suppose the group contains nodes $\{P_{i_1}, \ldots, P_{i_{n'}}\}, n' \leq n$, that hold key blocks $\{B_{i_1}, \ldots, B_{i_{n'}}\}$,

respectively. Observe that the system allows $n - n'$ new nodes to join during the rest of the system lifetime. If a new node $P_{i_{n'+1}}$ is going to join the group, it needs to obtain the key block $B_{i_{n'+1}}, B_{i_{n'+1}} \in \mathcal{B} \setminus \{B_{i_1}, \ldots, B_{i_{n'}}\}$. The new node can acquire the key block from some existing nodes, called sponsors, provided that the key blocks of the sponsors cover the key block of the new node (such sponsors can be found in a trivial way).

Node admission require the group to do the following. First, the nodes $\{P_{i_1}, \ldots, P_{i_{n'}}\}$ establish a common key k_C by following the protocol of temporarily leaving nodes in which case no node leaves. [6] Each node P_i replaces every key $k \in B_i$ by a fresh key $k' = k + k_C$ to provide backward secrecy. Assuming the sponsors are $\{P_{g_1}, \ldots, P_{g_s}\} \subseteq \{P_{i_1}, \ldots, P_{i_{n'}}\}$, next, key distribution for the new node is as follows (see Table 3).

Table 3: Distributing system keys

$K' = B_{i_{n'+1}}$, $z = s$
while $K' \neq \emptyset$ {
    P_{g_z} sends to $P_{i_{n'+1}}$ the fresh keys for $B_{g_z} \cap B_{i_{n'+1}}$ over a secure channel
    $K' = K' \setminus B_{g_z}$
    $z = z - 1$
}

If multiple new nodes simultaneously join the group, the key replacement step above is carried out once only and each new node will obtain its key block by following the steps depicted in Table 3. Observe that if the keys are not refreshed, the new nodes may recover previous group keys. The key replacement step guarantees backward secrecy as it ensures the new nodes, that are unable to find k_C, hold a new version of keys which is useless to find the previous group keys.

After obtaining the key block, the new joined node might sponsor other new nodes. Also, the enlarged group can have secure communication by following the protocol of temporarily leaving nodes in which case no node leaves.

4.3 Distributed Establishment of Secure Channels

The model is as follows. If a new node has secure channels (trust relations) with some $t+1$ existing nodes, then the secure channels (trust relations) can be used as a foundation to establish a secure channel between the new node with every existing node, with the absence of a trusted authority.

Suppose there are n' nodes $\{P_{i_1}, \ldots, P_{i_{n'}}\}$ in the group, and each node P_i has a secret polynomial $F(x, i)$, where $F(x, y)$ is a symmetric polynomial of variables x and y having degree at most t over a finite field $GF(q)$. (We show a protocol whereby $\ell, \ell \leq n'$, nodes collectively generate $F(x, y)$ and distribute $F(x, i)$ to P_i – see Appendix A) Also, suppose a new node $P_{i_{n'+1}}$ has secure channels

[6] If $n' \leq r$, $k_C = k_G$. If $n' > r$, a node takes the initiative to randomly generate k_C and securely communicate k_C to the group through a multicast channel.

(trust relations) with some $t+1$ nodes $\{P_{h_1}, \ldots, P_{h_t}\} \subset \{P_{i_1}, \ldots, P_{i_{n'}}\}$, but not with others. The new node $P_{i_{n'+1}}$ will be able to establish secure channels with all other nodes as follows.

1. For each j, $1 \leq j \leq t+1$, the node P_{h_j} whose secret polynomial is $F(x, h_j)$, sends $F(i_{n'+1}, h_j)$ to the new node $P_{i_{n'+1}}$ through the existing secure channel.

2. The new node $P_{i_{n'+1}}$ collects $t+1$ points $F(i_{n'+1}, h_1), \ldots, F(i_{n'+1}, h_{t+1})$ to compute a polynomial $f(x)$ of degree at most t such that $f(h_j) = F(i_{n'+1}, h_j)$, $1 \leq j \leq t+1$. This will give $f(x) = F(x, i_{n'+1})$ (the computation will be shown in Appendix B).

By referring to the description of Blom's key distribution in Section 4.1.2, the new node $P_{i_{n'+1}}$ shares a secret key $F(i_{n'+1}, i_j) = F(i_j, i_{n'+1})$ with each node P_{i_j}, for $1 \leq j \leq n'$. This shared key can be used for authentication and secure communication between the two nodes. In general, each node can establish a secure channel with every other node in the group using the secret key shared by the two nodes.

The ability for a node to establish secure channels with all other nodes is a desirable property in ad hoc network. For example, a new node wants to acquire key information from its sponsors that requires secure channels between the new node and its sponsors to be preserved (for secure delivery of the key information). If they do not exist, the required secure channels can be created by some $t+1$ other nodes provided that secure point-to-point communication is possible between the new node and the $t+1$ nodes. In this case, each sponsor P_g encrypts key information using $F(g, i_{n'+1})$ and sends the result to the new node $P_{i_{n'+1}}$ through a public channel. Only the node who claims to be $P_{i_{n'+1}}$ can decrypt the ciphertext to get the key information.

The proposed solution is essential for a system that provides ubiquitous authentication services, such as [25, 9, 11]. Below we show how our solution can be applied to Zhou and Haas's authentication system in [25]. Suppose a new node $P_{i_{n'+1}}$ would like to get a certificate for its public key $PK_{i_{n'+1}}$. Zhou and Haas's system employs ℓ-out-of-ω secret sharing to distribute the certification authority to ω servers, and cooperation of ℓ servers can provide a valid certification function as if the certification authority performs the service. Assume the ℓ servers are $\{S_{p_1}, \ldots S_{p_\ell}\}$, in detail,

1. $P_{i_{n'+1}}$ sends his identity $i_{n'+1}$ and its public key $PK_{i_{n'+1}}$ to the ℓ servers using public channels.

2. For each $1 \leq j \leq \ell$, the server S_{p_j} signs $PK_{i_{n'+1}}$ by using its key to get a partial signature $Sig_{p_j}(PK_{i_{n'+1}})$, then encrypts $Sig_{p_j}(PK_{i_{n'+1}})$ using the key $F(p_j, i_{n'+1})$, and sends the ciphertext to $P_{i_{n'+1}}$ through a public channel.

3. $P_{i_{n'+1}}$ collects all ℓ ciphertexts and decrypts them using the corresponding key $F(i_{n'+1}, p_j)$ to get the ℓ partial signatures and then computes the full signature of $PK_{i_{n'+1}}$.

5 Conclusion

We have proposed a group key distribution scheme that is suitable for ad hoc networks. Neither system setup nor key update in the scheme requires a trusted authority, instead, the group operations are performed by nodes in the system. We have shown that the proposed scheme is secure and efficient, and it provides reliable and ubiquitous security services.

We have also proposed a method to establish secure channels in a distributed way. The method, which is based on private-key encryption, allows each pair of nodes in the system to securely and efficiently form a secure channel.

References

[1] T. Asano. A revocation scheme with minimal storage at receivers, *Advances in Cryptology – ASIACRYPT 2002, LNCS 2501*, pages 433-450, 2002.

[2] G. Ateniese, M. Steiner, and G. Tsudik. Authenticated group key agreement and friends, *Proceedings of ACM CCS '98*, pages 17-26, 1998.

[3] R. Blom. An optimal class of symmetric key generation systemsl, *Advances in Cryptology – EUROCRYPT '84, LNCS 209*, pages 335-338, 1985.

[4] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system, *Advances in Cryptology – EUROCRYPT '94, LNCS 950*, pages 275-286, 1995.

[5] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor and B. Pinkas. Issues in multicast security: a taxonomy and efficient constructions, *Proceedings of INFOCOM '99*, pages 708-716, 1999.

[6] T. C. Chiang and Y. M. Huang. Group keys and the multicast security in ad hoc networks, *First International Workshop on Wireless Security and Privacy (WiSPr'03)*, 2003.

[7] D. Halevy and A. Shamir. The LSD broadcast encryption scheme, *Advances in Cryptology – CRYPTO 2002, LNCS 2442*, pages 47-60, 2002.

[8] I. Ingemarsson, D. Tang and C. Wong. A conference key distribution scheme, *IEEE Transactions on Information Theory, IT-28, 5, September 1982*, pages 714-720.

[9] A. Khalili, J. Katz and W. A. Arbaugh. Toward secure key distribution in truly ad-hoc networks, *Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, pages 342-346, 2003.

[10] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla and S. Lu. Adaptive security for multi-layer ad-hoc networks, *John Wiley InterScience Press Journal : Special Issue of Wireless Communications and Mobile Computing*, 2002.

[11] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks, *IEEE Ninth International Conference on Network Protocols (ICNP'01)*, pages 251-260, 2001.

[12] H. Kurnio, R. Safavi-Naini and H. Wang. Efficient revocation schemes for secure multicast. *Information Security and Cryptology – ICISC 2001, LNCS 2288*, pages 160-177, 2002.

[13] H. Luo, J. Kong, P. Zerfos, S. Lu and L. Zhang. Self-securing ad hoc wireless networks, *IEEE Symposium on Computers and Communications (ISCC'02)*, 2000.

[14] C. J. Mitchell and F. C. Piper. Key storage in secure networks, *Discrete Applied Mathematics* **21** (1988), 215-228.

[15] D. Naor, M. Naor and J. Lotspiech. Revocation and tracing schemes for stateless receivers, *Advances in Cryptology – CRYPTO 2001, LNCS 2139*, pages 41-62, 2001.

[16] K. Obraczka and G. Tsudik. Multicast routing issues in ad hoc networks, *IEEE International Conference on Universal Personal Communication (ICUPC'98)*, 1998.

[17] A. Perrig, D. Song and J.D. Tygar. ELK, a new protocol for efficient large-group key distribution, *Proceedings of IEEE Security and Privacy Symposium 2001*, pages 247-262, 2001.

[18] M. Ruszinkó. On the upper bound of the size of the r-cover-free families, *Journal of Combinatorial Theory A* **66** (1994), 302-310.

[19] M. Steiner, G. Tsudik, and M. Waidner, Diffie-Hellman key distribution extended to group communication, *Proceedings of ACM CCS '96*, pages 31-37, 1996.

[20] M. Steiner, G. Tsudik and M. Waidner. Key agreement in dynamic peer groups. *IEEE Transactions on Parallel and Distributed Systems* **11** no. 8 (2000), 769-780.

[21] D. R. Stinson, T. van Trung and R. Wei, Secure frameproof codes, key distribution patterns, group testing algorithms and related structures, *J. Statist. Plan. Infer.,* **86** (2000), 595-617.

[22] D. S. Stinson, R. Wei and L. Zhu. Some new bounds for cover-free families, *Journal of Combinatorial Theory A* **90** (2000), 224-234.

[23] D. M. Wallner, E. C. Harder, and R. C. Agee. Key management for multicast: issues and architectures, *Internet Draft, ftp://ftp.ietf.org/internet-drafts/draft-wallner-key-arch-01.txt*, 1998.

[24] C. K. Wong, M.Gouda, and S.S. Lam. Secure group communications using key graphs, *Proceedings of SIGCOMM '98*, pages 68-79, 2000.

[25] L. Zhou and Z. Haas. Securing ad hoc networks, *IEEE Network 13* **6** (1999), 24-30.

A Generation of Symmetric Polynomial $F(x, y)$

The group consists of nodes $\{P_{i_1}, \ldots, P_{i_{n'}}\}$. Without loss of generality, we assume the ℓ, $\ell \leq n'$, nodes are $\{P_{i_1}, \ldots P_{i_\ell}\}$ and a secure channel exists between every pair of the nodes. The ℓ nodes collectively generate the symmetric polynomial $F(x, y)$ as follows.

1. For each j, $1 \leq j \leq \ell$, the node P_{i_j} randomly generates a symmetric polynomial $F_{i_j}(x, y)$ of variables x and y having degree at most t over a finite field $GF(q)$, independently sends the polynomial $F_{i_j}(x, i_g)$ to the node P_{i_g}, for $1 \leq g \leq \ell$ and $g \neq j$, through the existing secure channels.

2. For each j, $1 \leq j \leq \ell$, the node P_{i_j}, after receiving the polynomials $F_{i_g}(x, i_j)$, for $1 \leq g \leq \ell, g \neq j$, computes a polynomial $f_{i_j}(x) = \sum_{g=1}^{\ell} F_{i_g}(x, i_j) \bmod q$.

3. For each j, $1 \leq j \leq \ell$, observe that $f_{i_j}(x) = F(x, i_j)$ where $F(x, y) = \sum_{g=1}^{\ell} F_{i_g}(x, y) \bmod q$. Therefore, the symmetric polynomial $F(x, y)$ is implicitly generated by the ℓ nodes and each node P_{i_j} has $F(x, i_j)$ of $F(x, y)$.

Subsequently, the ℓ nodes can distribute $F(x, i_j)$ to every remaining node P_{i_j}, for $\ell+1 \leq j \leq n'$, in the group as follows.

If $\ell \leq t$: For each g, $1 \leq g \leq \ell$, the node P_{i_g} sends the polynomial $F_{i_g}(x, i_j)$ to the remaining node P_{i_j} through the existing secure channel. The node P_{i_j}, after receiving the polynomials $F_{i_g}(x, i_j)$, for $1 \leq g \leq \ell$, computes a polynomial $f_{i_j}(x) = \sum_{g=1}^{\ell} F_{i_g}(x, i_j) \bmod q = F(x, i_j)$.

If $\ell > t$: Any $t + 1$ of the ℓ nodes perform the steps described in Section 4.3.

B Computation of Polynomial $f(x)$

Having $t + 1$ points $F(i_{n'+1}, h_1), \ldots, F(i_{n'+1}, h_{t+1})$, the new node $P_{i_{n'+1}}$ computes the polynomial $f(x)$ such that $f(x) = F(x, i_{n'+1})$ using Langrange interpolation as follows. Let $\Phi = \{h_1, \ldots, h_{t+1}\}$.

$$f(x) \quad = \quad \sum_{h \in \Phi} F(i_{n'+1}, h) \times \psi(\Phi, h) \bmod q \ ,$$

where $\psi(\Phi, h) = \prod_{e \in \Phi, e \neq h} \frac{x-e}{h-e} \bmod q$.