ABSTRACT—Today’s ever smaller computing systems are increasingly spreading in our ubiquitous environment. Being available ubiquitously in the devices and appliances that we use everyday and everywhere, these embedded computing systems are accessible to mobile users via hand-held devices connected over wireless networks. A mobile ad hoc network (MANET) is one of the important wireless networks. In a MANET a reliable key management system is required to generate and distribute symmetric encryption/decryption keys. The key management schemes proposed in MANETs so far have used trusted third parties (TTP) which have limitations because of the mobility of nodes. A Distributed Key Pre-distribution Scheme was proposed based on a probabilistic method without relying on any TTP but with results identical to TTP-based schemes. The scheme utilized cover-free family (CFF) properties. However, the precondition of the probabilistic method was claimed to be falsely deduced.

In this paper, we propose two distributive key management schemes using maximum distance separable codes (MDS). First, we will construct a practical \((n, t+1)\)-threshold key management system. Second, we propose a key pre-distribution scheme achieving CFF properties. We use a global MDS code instead of the probabilistic method to generate node keys. The scheme is secure enough against malicious nodes’ fraud and tapping. The effects of block size and network parameters are also studied.

Key Words: cover-free family, distributed key pre-distribution, maximum-distance separable code, mobile ad hoc networks, symmetric key, threshold key management.
1. INTRODUCTION

Today’s ever smaller computing systems are increasingly spreading in our ubiquitous environment. Being available ubiquitously in the devices and appliances that we use everyday and everywhere, these embedded computing systems are accessible to mobile users via hand-held devices connected over wireless networks. However, due to the nature of wireless communication and lack of any security infrastructure, networks are susceptible to attacks ranging from passive eavesdropping to active interference. In recent years, network security has received critical attention in wireless networks.

A mobile ad hoc network (MANET) is one of the important wireless networks, and the MANET provides multimedia communication and intelligence services in a ubiquitous and pervasive computing environment. It is imperative in such applications to have efficient means of developing trust between the nodes of the network. For example, nodes in MANETs have to perform authenticated exchanges for building a routing table, or relaying messages between other nodes. In such a scenario, a malicious action by a single node could have a potentially disruptive effect over the entire network. Security of ad hoc networks basically needs a cryptographic key structure to be known among the communicating parties. This key must be available before any routing operations take place.

Most current secure protocols proposed for MANETs assume the existence of a reliable key mechanism to generate/distribute the key and deliver them to communicating parties. However, it is improper in MANETs to have any routing functionality before ensuring security. In other words, we can not trust the routing protocol to deliver the keys to the nodes due to disappearance of any trust relationship before the deployment. Thus, any malicious node can fake routing messages and deliver wrong keys to the participant bringing the network into collapse.

Among the schemes proposed for key management in MANETs, distributed schemes are more suitable than central schemes which are exposed to a single point of failure. Distributed schemes are classified into partially and fully distributed schemes. In partially distributed schemes, there are \( k \) servers (central points) used to generate the key and distribute certificates. Fully distributed schemes let the nodes co-operate to generate the key. In latter case, compromising any node will not reveal any information about other nodes’ keys.

Blom [6] proposed a distributed symmetric key generation system (SKGS) based on the key pre-distribution scheme. In SKGS, a central server is responsible for the creation and distribution of nodal key chains. Nodes can derive any future key from the key chain they received from the main server. SKGS provides security against cooperative malicious attacks: to retrieve the key of the node the attacker has to compromise a minimum number of nodes \((t+1)\), and with any number less than \(t+1\), the attacker can not retrieve the key chain for the attacked node. The main problem in the SKGS approach is the single point of failure of the central server: if an attacker compromises the central node, all the keys it has can be retrieved, and the security of the network is exposed.

Threshold cryptography [14] is a proposed cryptography scheme suited for mobile networks to provide robustness and defense against a single point of failure in the central server. In threshold cryptography, the central server function is distributed among a group of servers. When one of these servers is compromised or fails, the rest of the group can take the place of this single server. This distribution provides a co-operative security model and robustness against the single point of failure. The main drawback in threshold cryptography is the difficulty in applying the distributive function. Until now, there has been no practical security model that could use threshold cryptography efficiently.

The distributed key pre-distribution scheme (DKPS) is a fully distributed scheme for MANETs proposed by Chan [1]. One key idea of the DKPS is that each node individually picks a
set of keys from a large publicly-known key space following some procedure. This set of keys is called the node key chain. At the end of all DKPS phases, the key chains of all the nodes satisfy the following exclusion property with a high probability: any subset of nodes can find from their key chains at least one common key not covered by a collusion of, at most, a certain number of nodes outside the subset. The DKPS is based on the precondition under which the key sets distributed to the network nodes can form a cover-free family (CFF) [13]. However, Wu and Wei [2] found that the precondition was falsely deduced. They claim that the probabilistic method (Chan used this method) cannot yield CFF practical for key distribution.

In this paper, we propose two schemes using maximum distance separable codes (MDS). First, we will construct a practical \((n, t + 1)\)-threshold key management system. Second, we propose a key pre-distribution scheme achieving CFF properties. We use a global MDS code to generate node keys. For this purpose, a publicly known MDS generator matrix will be available in the network for every node. The scheme is secure enough against malicious nodes’ fraud and tapping. The effects of block size and network parameters on satisfying CFF properties are also studied.

This paper is organized as follows. Section 2 gives an overview about the MDS code and the CFF which will be used throughout the paper. Also, previous work is briefly introduced. Section 3 proposes a threshold key-management scheme. Section 4 proposes a key pre-distribution scheme, shows the mathematical model for the proposed scheme, and shows our simulation results. Finally, Section 5 concludes this paper.

2. PRELIMINARY AND RELATED WORK

2.1 Threshold Cryptography

The basic idea of threshold cryptography is to protect encryption keys by making a group of nodes (servers) co-operate to provide fault-tolerant distribution of keys within the security model. Threshold cryptography allows one to distribute a piece of secret information (keys) among several servers in a way that meets the following two requirements:

- no group of compromised servers (smaller than a given threshold) can figure out what the secret is, even if they cooperate.
- when it becomes necessary that the secret information be reconstructed, any large enough number of servers (a number larger than the above threshold) can always do it.

The primary objective of sharing of a key among multiple servers (shareholders) in threshold cryptography system is to have distributed architecture in a hostile environment. Other than sharing keys or working in a distributed manner, threshold cryptography can be implemented to redundantly split the message into \(n\) pieces such that, with \(t\) or more pieces the original message can be recovered. This ensures secure message transmission between two nodes over \(n\) multiple paths.

An \((n, t + 1)\)-threshold cryptography scheme allows \(n\) parties to share the ability to perform a cryptographic operation (e.g., creating a digital signature), so that any \(t + 1\) parties can perform this operation jointly, while it is infeasible for at most \(t\) parties to do the same operation even by collusion. In our proposed key management scheme, the \(n\) servers of the key management service share the ability to create the key chain for the node. For the service to tolerate \(t\)-compromised servers, we employ an \((n, t + 1)\) threshold cryptography scheme and divide the secret symmetric matrix \(D\). The matrix is used in conjunction with the generator matrix of the MDS code, to generate the key chains for the participant nodes in the network into \(n\) shares \((s_1, s_2, \ldots, s_n)\), assigning one share to each server. We call \((s_1, s_2, \ldots, s_n)\) shares of the secret matrix \(D\).
In order for a node to have its key chain, it will ask $t+1$ on $n$ shareholder to generate a partial share of the key chain. Each one of those $k$ servers generates a partial sum for this key chain and sends it to the sender, who will in turn perform the operation of complete key chain retrieval from these partial sums.

### 2.2 Maximum Distance Separable (MDS) Codes

**Code** is a group of code words with known length, such that the mapping between the data we are going to transmit and the code words is a one-to-one function. If we have data $m$, of length $k$, using codes we can convert it to a code word $c$, with length $n$, where $n > k$ using some defined encoding scheme. The code word $c$ will have $n-k$ redundant elements. Those elements are used for parity checking and error detection purposes on the decoding side.

The distance $d$ of the code $C[n, k]$ is the number of different elements between any two code words. Singleton [4] found that the minimum distance for code $C[n, k]$ must satisfy $d(C) \leq n-k+1$. When Singleton bound is at maximum (i.e., $d(C) = n-k+1$), the code $C[n, k]$ will be an MDS code.

MDS codes are a class of error control codes created to deal with the problem of communication over a noisy channel, where some of the bits of the message may be corrupted en route. Such codes are used in cryptography and coding theory applications [10] [11]. The MDS code satisfies Singleton Bound [4]. The value of distance $d$ determines the maximum error $t$ that can be corrected using this code ($d \geq 2t+1$). We can view MDS codes as a linear independent set of vectors that form a subspace in the vector space GF($q^k$). The basis of these vectors can form the generator matrix ($G$) for that MDS code.

All MDS code words (i.e., the key chains for the nodes) can be generated using a carefully chosen special matrix called a generator matrix $G$, an $[n, k]$-MDS code with $k$ rows and $n$ columns. The relation between $G$ and code words $C$ can be expressed as

$$C = v.G,$$

where $v$ is the message represented as a vector and $C$ is the code word generated for $v$.

Let $C$ be an $[n, k]$ linear code in GF($q^k$) with minimum distance $d$, we can assume $C$ to be MDS if the following properties were satisfied [4]: (1) Generator matrix $G$ for this code is an MDS code. (2) The code $C^\perp$ dual to $C$ is an MDS code. (3) Any $k$ columns of a generator matrix for $C$ are linearly independent. (4) If a generator matrix for $C$ is in standard form $[I, A]$, then every square sub matrix of $A$ is nonsingular. (5) Given any $d(G)$ coordinate positions, there is a minimum weight code word whose non-zero entries are in precisely these positions.

The simplicity of MDS codes relies on the fact, that $G$ is globally known to all the nodes in the network, and each node does not need large memory space to save all its keys. Using matrix transformation to generate code $C$ from a secret symmetric matrix $D$, the security of MDS codes relies on the idea that all columns of $C$ are linearly independent and any coalition of less than $k$ nodes can not retrieve the secret. Reed and Solomon [5] have found an equation to generate the $G$ matrix for any MDS-[n, k, d] in GF($q^k$), by using the value $\alpha$, which is a primitive root of GF($q$).

Reed-Solomon codes are very common MDS codes in the communication world. They are famous for their optimized generator matrix $G$ in GF($q$). Using $\alpha$ which is a primitive root of $q$ (i.e., $\alpha^x \equiv 1 \mod q$, the least value of $x$ is $q-1$), we define the generator matrix for Reed-Solomon codes as $G_{ij} = \alpha^{(j-1)(i-1)} \mod q$.

Using the Reed-Solomon formula, the node who wants to retrieve $G$ asks its neighbor to send the values of $\alpha$, $n$, $k$ and $q$. This can be done with only two extra packets.
2.3 Cover-Free Families (CFF)

Cover-free families were first introduced by Kautz and Singleton [13] to investigate super-imposed codes [12]. Erdős and Frankl considered CFF as a combinatorial problem. Next, Mitchel defined the concept of key distribution patterns, which in fact are a generalized type of CFF.

A set system is a pair \((X, F)\), where \(X\) is a set of points and \(F\) is a set of blocks of \(X\). We define \(N = |X|\) and \(T = |X|\). The classical definitions of cover-free families [13] can be written as follows.

\[
B_0 \not\subseteq \bigcup_{j=1}^{r} A_j.
\]

Definition 1. A set system \((X, F)\) is called a \(r\)-cover-free family (or \(r\)-CFF) provided that, for any \(r\) blocks \(A_1, A_2, \ldots, A_r \in F\) and any other block \(B_0 \in F\), we have

\[
\bigcap_{i=1}^{w} B_0 \not\subseteq \bigcup_{j=1}^{r} A_j.
\]

Definition 2. A set system \((X, F)\) is called a \((w, r)\)-cover-free family (or \((w, r)\)-CFF) provided that, for any \(w\) blocks \(B_1, B_2, \ldots, B_w \in F\) and any other \(r\) blocks \(A_1, A_2, \ldots, A_r \in F\), we have

\[
\bigcap_{i=1}^{w} B_0 \not\subseteq \bigcup_{j=1}^{r} A_j.
\]

Definition 3. A set system \((X, F)\) is called a \((r; d)\)-cover-free family (or \((r; d)\)-CFF) provided that, for any block \(B_0 \in F\) and any other \(r\) blocks \(A_1, A_2, \ldots, A_r \in F\), we have

\[
\left| B_0 \bigcup_{j=1}^{r} A_j \right| > d.
\]

Definition 4. A set system \((X, F)\) is called a \((w, r; d)\)-cover-free family (or \((w, r; d)\)-CFF) provided that, for any \(w\) block \(B_1, B_2, \ldots, B_w \in F\) and any other \(r\) blocks \(A_1, A_2, \ldots, A_r \in F\), we have

\[
\left| \bigcap_{i=1}^{w} B_0 \bigcup_{j=1}^{r} A_j \right| > d.
\]

The last definition is the general definition of CFF, and it states that the intersection of any \(w\) blocks contains at least \(d\) points that are not in the union of \(r\) blocks.

2.4 Related Work

Key management protocols are classified into key distribution, key agreement, and key pre-distribution [1]. However, each class has limitations to be used in ad hoc networks. The key distribution protocols [16] [17] [18] rely on online trusted third parties (TTP) to distribute session keys to nodes. The protocols are infeasible for ad hoc networks because the TTP may be out of reach or not available to some of the nodes. In contributory key agreement protocols [19] [20], each node contributes an input to establish a common secret through successive pairwise message exchanges among the nodes in a secure manner. The protocols are not practical to ad hoc networks either. These protocols are fully distributed and self-organized without needing any TTP, but they are not robust to changing topology or intermittent links.

The key pre-distribution scheme (KPS), introduced by Blom [6], offers practical and efficient solutions to the key management problem in a variety of models including broadcast/multicast [21] [22] and sensor networks [23] [24]. In KPS, an offline TTP pre-initializes each node in a set
with some secret information, with which any subset of nodes later on can find or compute non-interactively, a common session key, which is secret, from a collusion of at most a certain number of nodes outside of the set. In ad hoc networks, since the nodes may have no knowledge about whom they will meet prior to network deployment, it is not possible for them to contact the same TTP in advance for offline initialization. Let us explain Blom’s scheme in more detail.

Blom proposed a symmetric key generation system (SKGS) [6] based on secret sharing systems. In SKGS, nodes are supplied with a relatively small amount of secret data that is used to derive all the node's keys. A central server (trusted authority) generates a global matrix $G$ of size $k \times n$ that is known to all the nodes in the network, and a symmetric secret matrix $D$ of size $k \times n$. The central node calculates the key matrix for the network as $K = (D \cdot G)^T \cdot G$. Because $D$ is symmetric, $K$ will be also symmetric, for rows $i$ and $j$ in $K$, we have $K_{i,j} = K_{j,i}$. So, $K_{i,j}$ is common between the rows $i, j$. If row $i$ is the key chain for node $i$, and row $j$ is the key chain for node $j$, the element $K_{i,j}$ will be the symmetric key between them.

Because $G$ is known by all the participants in the networks, while external nodes (malicious nodes) do not know this matrix $G$, the central server delivers the $i^{th}$ row of $(D \cdot G)^T$ to node $i$. Upon reception, node $i$ will calculates its key chain $k_i = i^{th}$ row of $(D \cdot G)^T \cdot G$. This division of key generation into more than one step adds more secrecy to their key chains, and makes it harder for the intruder to retrieve any information about other nodes. Blom showed in his paper that by using the SKGS scheme, at least $k$ users have to co-operate to get any information about keys they do not have. Thus, any coalition of less than $k$ nodes cannot reveal any information about the key chain of any other node.

The problem with this approach is that it relies on a central point to do all the work, making it vulnerable as a single point of failure. Another problem arising from key request and delivery is that some nodes may not be able to reach the central server due to the existence of broken links or untrustworthy routing information.

To apply Blom’s scheme to ad hoc networks, Chan [1] proposed a Distributed Key Pre-distribution Scheme (DKPS), which is a collection of distributed cryptographic protocols. The DKPS does not rely on any TTP but with results identical to a TTP-based KPS. The basic idea of DKPS is that each node individually picks a set of keys from a large publicly-known key space following some procedure, so that at the end, the key patterns of all the nodes satisfy the following exclusion property with a high probability: any subset of nodes can find from their key patterns at least one common key not covered by a collusion of, at most, a certain number of other nodes outside the subset. This scheme is based on the probabilistic method to construct $(w, r; d)$-CFF $(N, T)$. Here, the universal set $P$ will be created and divided into blocks. Every node picks a key from every $P$ block to construct its key chain. Chan shows by using cryptography homomorphism, that node keys yield $CFF-(N, T)$ with a probability of $1 - e^{-t}$, if $T$ is chosen carefully. However, Wu and Wei [2] found that the precondition was falsely deduced. They claim that the probabilistic method cannot yield a CFF practical for key distribution.

3. PROPOSED THRESHOLD KEY MANAGEMENT SCHEME

The assumptions we made and the notation we will use are as follows: (1) Symmetric matrix $D$ is of size $k \times k$. (2) MDS code generator matrix $G$ is of size $k \times n$ with $k < n$. (3) Number of servers is equal to $n$. (4) All the keys value are in $GF(q)$. (5) Any node is able to reach $k$ server when it requests a key. (6) $K$ denotes key matrix.

In order to construct a practical key management system for MANETs, we used Blom's SKGS in conjunction with the theory of threshold cryptographic systems. In addition, we also used coding theory to overcome the shortcomings of central approaches in SKGS, and to provide a practical threshold cryptographic system capable of operating in a dynamic environment.
In our proposed approach, each node will be provided with a group of keys (key chain) when it joins the network. This key chain will be used to derive all the future keys for this node, such that any two nodes can calculate symmetric keys between them based on these key chains. Key authority servers are distributed among the network in a threshold way. There are $n$ key authority servers with $k$ threshold value. Generator matrix ($G$) for the MDS code used in our distributive key management system (with code distance $d = n - k + 1$) will be distributed among the $n$ key authority servers. Recall from MDS code properties that every set of $k$ columns in $G$ are linearly independent. Thus, it is enough to know $k$ columns of $G$ to construct it [15]. Now, any $k$ servers chosen from the $n$ authority servers are capable of reconstructing $G$.

Each of the $n$ servers in the network will be initialized with a column $r$ of $G$ during the network setup, such that any $k$ nodes can collaborate with each other and reconstruct $G$, while any number less than this threshold value $k$ will not be able to reconstruct $G$. On the other hand, since matrix $D$ is known only to the authority servers, regular nodes do not know any information about $D$ except what they receive from the servers during the key generation's phase.

Now, each of the $n$ servers knows what the secret matrix $D$ is, but knows only a part of $G$. When a node demands a key, it will initialize the key generation system, and $k$ servers will be asked to communicate with. One server will be elected as a collector, and every selected server (within the $k$ group) will send its share of $G$ to the collector, who in turn will construct $G$ and then calculate $K$ which contains shares from the other servers. Since the key chain of node $i$ will be row $i$, the collector will send the key chain to node $i$.

4. PROPOSED KEY PRE-DISTRIBUTION SCHEME

As mentioned in Section 2, Chan’s probabilistic method cannot yield CFF practical for key distribution, because its precondition was falsely deduced. In this section, we propose a key pre-distribution scheme achieving CFF properties.

4.1 Main Idea

The main idea in the proposed scheme is to use a global MDS code to generate a node’s key chain. For this purpose, a publicly known MDS generator matrix will be available in the network for every node. Code size must be chosen to be large enough to guarantee security and resist known attacks. Every node that wants to join the network can retrieve this $G$ matrix by asking neighbors for the parameters of the generator matrix (e.g., $\alpha$, $n$, $k$ and $q$), then the node will generate a random vector $v$ within GF$(q)$. This vector will be used to generate all the keys for this node, as we will describe later in this section.

To construct the global MDS generator matrix, we will use Reed-Solomon codes [5]. We can define $G$ in GF$(q)$ of $k$ rows and $n$ columns with distance $d = n - k + 1$ as the $[n, k, d]$ generator matrix. The values of $n$, $k$, $d$ and $q$ must satisfy the inequality $1 \leq k < n < q$, here $n$ will be the length of node key chain and $q^k >>$ number of nodes in the network. The elements of generator matrix $G_{ij} = \alpha^{(i-j)(i-1)}$ with $\alpha$ as a primitive root for $q$.

$$G = \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & \alpha & \alpha^2 & \alpha^{n-1} \\
1 & \alpha^2 & (\alpha^2)^2 & (\alpha^{n-1})^2 \\
\vdots & \vdots & \ddots & \vdots \\
1 & \alpha^{k-1} & (\alpha^2)^{k-1} & (\alpha^{n-1})^{k-1}
\end{bmatrix}.$$
With $G$ in hand, node $i$ will generate a random vector $v_i$ of length $k$ and calculate its secret key chain ($K_i$). According to MDS code properties, any linear combination of $G$ rows is still MDS, and $K_i$ will have $d$ distance from any $K_{ij}$. Any two nodes can find the common keys between their key chains using the cryptography homomorphism [7] [8] with secure shared key discovery (SSD) [1] or Rivest scheme [9]. Recall that homomorphism is simply a special class of encryption function, which allows the encrypted data to be operated directly, without any knowledge about the decryption function (i.e., if the encrypted messages $E(x)$ and $E(y)$ are known, $E(x + y)$ can be calculated without any knowledge about the decryption function).

The detailed steps of the proposed approach for node $i$ is as follows: (1) Generate a generator matrix $G[n, k, d]$ in GF($q$). (2) Pick $v_i$ with length $k$. (3) Calculate $K_i = v_i \times G$ which is $n$ length vector. (4) Apply the SSD (or Rivest) with another node which node $i$ wants to communicate with. (5) Now, node $i$ has a list of common keys and can generate the session key from those keys.

### 4.2 Example

We will demonstrate the proposed approach using an example. First, we generate $G$ matrix with $\alpha = 5$ and $q = 7$ such that $\alpha$ is a primitive root of $q$. Recall that $G$ is of size $(k \times n)$ where $n$ is the degree of $\alpha$ primitive root. With these bounds, $G$ can be:

$$G = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 3 & 2 & 6 & 4 & 5 \\
1 & 2 & 4 & 1 & 2 & 4 \\
1 & 6 & 1 & 6 & 1 & 6
\end{bmatrix}$$

For node $i$ who wants to communicate with node $j$, each of $i$ and $j$ will construct its key chain from $G$. Node $i$ picks a random vector $v_i$ of length $k = 4$ within GF(7), and node $j$ will do the same. Now, $i$ and $j$ will construct their key chains. Since $G$ is global and all the nodes in the network can construct it, key chains (of length $k$) will be calculated in the nodes:

$$K_i = v_i \times G = \begin{bmatrix} 4 & 1 & 2 & 3 \end{bmatrix} \times G \mod 7 = \begin{bmatrix} 3 & 1 & 3 & 2 & 1 & 0 \end{bmatrix},$$

$$K_j = v_j \times G = \begin{bmatrix} 6 & 2 & 1 & 4 \end{bmatrix} \times G \mod 7 = \begin{bmatrix} 6 & 3 & 4 & 1 & 6 & 2 \end{bmatrix}.$$

Node $i$ and $j$ will perform either SSD or the Rivest scheme, which uses privacy homomorphism to find common keys between $K_i$ and $K_j$ chains. In this example $i$ and $j$ find that $[3, 1, 2]$ in their key chains are common between them. Thus, node $i$ and $j$ can communicate securely using a key value resulting from hashing the common keys $[3, 1, 2]$. To insure key security, we need a large value for $q$, $k$ and $n$. With large values for these parameters, there will be a probability of $e^{-t}$ ($t < \frac{d - 1}{2}$), that node $i$ and $j$ key chains do not satisfy CFF properties, or have no common values between their key chains.

### 4.3 Mathematical Models

The effect of the number of nodes (blocks) $T$, universal set size $N$, and key chain size $k$ to achieve CFF was not studied in the DKS. For a $k$-uniform $(w, r; 1)$-CFF $(N, T)$, Stinson [3] proved the lower bound for $N$ with respect to $w$, $r$ and $T$ must satisfy

$$N \geq \binom{W + r + 1}{W} \log(T - r - w + 2),$$

(2)
and

\[ N \geq r(w \log T - \log r - w \log w). \]  

With \((w, r; 1)\)-CFF \((N, T)\) in hand, it is easy to construct \((w, r; d)\)-CFF \(((d+1)N, T)\) by repeating each element \(d+1\) times in the block. Basically, the DKS approach can be viewed as \((2, r; d)\)-CFF \((N, T)\), because communicating parties will exchange keys pairwise. We modify (2) and (3) for \((w, r; d)\)-CFF \((X, T)\)

\[ X \geq (d + 1) \left( \frac{w+r+1}{w} \right) \log (T-r-w+2), \]

and

\[ X \geq r(d+1)(w \log T - \log r - w \log w), \]

with \(X = (d+1)N\).

For \((1, r; 1)\)-CFF \((N, T)\) the key chain size \(k\) must satisfy the following equation

\[ T \leq \left( \begin{array}{c} N \\ T \end{array} \right) \left( \begin{array}{c} k-1 \\ t-1 \end{array} \right), \text{ where } t = \left\lceil \frac{k}{r} \right\rceil. \]  

Bound in (6) can be scaled for \((1, r; d)\)-CFF \((X, T)\)

\[ T \leq \left( \begin{array}{c} (d+1)N \\ T \end{array} \right) \left( \begin{array}{c} k-1 \\ t-1 \end{array} \right). \]  

Suppose we have \(T, w, r\) and \(l\), and we want to construct \((w, r; d)\)-CFF \((N, T)\) \([13]\), the lower bound for \(k\) can be formulated as

\[ k > \frac{8}{p} ((w+r)\log T - \log w! - \log !), \]

where \(p = \frac{(l-1)^r}{l^{w+r+1}}\) with a maximum distance

\[ d = \frac{pk}{l} + 1, \text{ with } N = k \times l. \]  

Wu and Wei \([2]\) proved that DKS does not guarantee CFF. Hence, the parameters calculated from the above equations to achieve CFF are not reasonable for mobile networks and the performance for probabilistic approach is not satisfactory in ad hoc network key schemes.

We can see that any two nodes can find a shared key with any other node in the network. Hence \(G\)'s rows are linearly independent, a coalition of less than \(T\) cannot reveal useful information about node \(i\) key chain.

**Theorem 1**: The proposed approach yields a \((2, r; d)\)-CFF \((N, T)\) with \(e^{-t}\) maximum error, with \(t = \frac{d-1}{2}\), when \(r\) is less than \(d\).

Proof: For node \(i, j, v_i\) and \(v_j\) to satisfy \([n, k, d]\) MDS code properties, any other node \(m\) will have
a distance \( d \) from \( i \) and \( j \). Let us define \( S = v_i \cap v_j \) and \( Z = \bigcup_{s=1}^{r} v_{s \neq \{i,j\}} \), \( S \) has at most \( n - d \) subkeys as \( v_i \) and \( v_j \) have. Node \( m \) will have \( n - 2d \) subkeys similar to \( S \). At the least, \( d \) nodes must cooperate to make \( Z \supset S \) and any number less than \( d \) cannot make \( Z \supset S \). So any value \( r < d \) will guarantee \( \bigcap_{i \neq j} v_i \setminus \bigcup_{s=1}^{r} v_{s \neq \{i,j\}} \geq d \). Thus, our system is a \((2, r, d)\)-CFF \((N, T)\).

General \((w, r; d)\)-CFF \((N, T)\) can be achieved using our system with a careful choice for the system parameters. According to (4) and (5), \( N \) must satisfy
\[
q^n k \geq \left( \frac{w + r + 1}{w} \right) \log(T - r - w + 2)
\]
and
\[
q^n k \geq r(d + 1) \log(T - \log r - \log w) \quad \text{with} \quad q^n k > N.
\]

**Theorem 2:** The proposed approach yields a \((w, r; d)\)-CFF \((N, T)\) with \( e^{-t} \) maximum error, with \( t = \frac{d - 1}{2} \), when \( r \) is less than \( d - w \).

Proof: For any \( w \) blocks \( v_1, v_2, \ldots, v_w \) and any other \( r \) blocks \( B_1, B_2, \ldots, B_r \) \((r < d - w)\) according to the discussion in Theorem 1, \( \bigcap_{s=1}^{w} v_s \) will have at least \( n - (d - w) \) different subkeys from other codes.

So \( r \) blocks \((r < d - w)\) can not make \( \bigcup_{i=1}^{r} B_i \supset \bigcap_{s=1}^{w} v_s \). Thus, to have \((w, r; d)\)-CFF \((N, T)\) the value for \( r \) must satisfy \( r < d - w \).

Stinson [13] has proven the probability that the system is not a \((w, r; d)\)-CFF \((N, T)\) for \( e^{-t} \) if the bounds for \( d \) and \( n \) satisfy the following
\[
d = \frac{pN}{2} + 1, \quad \text{with} \quad p = \frac{(w)^r r^t}{(r + w)^t + w}
\]
and
\[
N \geq \frac{8}{p} ((w + r) \log T - t - \log r! - \log w!).\]
The proposed scheme satisfies all the bounds.

### 4.4 Simulation Results

The To test our model and show that it satisfies CFF, we run a test program to generate the keys as specified in the Section 3.1. Two nodes are used, and each constructs its key chain. Then, we compare the key chains intersections with any set of \( r \) nodes, and we test the model for different values of \( r \).

As shown in Figure 1, the proposed scheme yields \((2, r, d)\)-CFF \((N, T)\), with \( e^{-t} \) maximum error, with \( t = \frac{d - 1}{2} \), when \( r \) is less than \( d \). For example, when the distance \( d = 4 \) and \( r = 3 \), according to the mathematical model \((r < d)\) the intersection of two nodes key chains in \( 1 - e^{-1.5} \approx 77\% \) satisfies the CFF properties. In the simulation with the same \( r \) and \( d \) we got 79% which satisfies CFF. When \( d \) increases \((\text{while} \ r < d)\), the percentage increases and reaches around 95% when \( d = 7 \) \((\text{with} \ r < d)\). When \( d = 6 \) and \( r = 5 \), the mathematical model in \( 1 - e^{-2.5} \approx 91\% \) could satisfy the CFF properties, which agrees with the simulation results (92%).

We can see that as the distance increases, the error decreases and the percentage of CFF approaches 100%. When \( r \) is larger than \( d \), the scheme does not guarantee to find a secure key between the participants \((\text{less than} \ 1 - e^{-t} \text{of the keys satisfies CFF})\). As shown in Figure 1, when \( r = 4 \) and \( d = 3 \), only 25% of the keys satisfies CFF.
5. CONCLUSION

In this paper, we proposed a distributive key management system suitable for ad hoc networks. Based on threshold cryptography and MDS codes with SKGS, the proposed system is applicable and practical. We also proposed a simple, practical, model for key distribution in wireless ad hoc networks using MDS codes. The proposed scheme guarantees cover free families properties. Hence, the scheme is secure enough against malicious nodes’ fraud and tapping. Using this approach in peer to peer communication guarantees perfect CFF where group communication has $e^{-t}$ probability to deviate from CFF. Future work will include the study of the effect of dishonest nodes on the security of the proposed system.

REFERENCES

4. S. Yamiolkoski, “The Road to Maximum Distance Separable Codes,” Lecture notes in projective geometry.

ABOUT THE AUTHORS

M. Al-Shurman is currently a faculty member of computer engineering at Jordan University of Science and Technology (JUST). He received his B.E.E from (JUST) and his MS and PhD from the University of Alabama in Huntsville (UAH). His research interests are the area of wireless networks, mobile ad hoc networks, wireless routing protocols, sensor networks and wireless security. He has several publications including journals and conference proceedings.
S.-M. Yoo is an Associate Professor of Electrical and Computer Engineering at the University of Alabama in Huntsville (UAH). Before joining UAH, he was an Assistant Professor at Columbus State University, Columbus, Georgia – USA. He earned MS and PhD degrees in Computer Science at the University of Texas at Arlington. His research interests include computer network security, wireless network routing, and parallel computer architecture. He has co-authored over 60 scientific articles in refereed journals and international conferences.

B. Kim received the Ph.D. degree in Computer Science and Software Engineering from the Auburn University, Alabama, USA in 2006. She joined the School of Electrical and Computer Engineering, Chungbuk National University in March 2007. Dr. Kim is very active in international academic activities and has participated in the organization of many international workshops. Her current research interests are in the areas of wireless ad hoc and sensor networks, network security and MIPv6.

S. Park received the BE degree in civil engineering from Hanyang University, Korea, in 1973, MS degree in computer science from University of Texas at Arlington, in 1986, and PhD degree in computer science from Oregon State University, in 1993. He is currently an assistant professor in the Department of MNGT, MIS, and CS, University of Southern Indiana. His research interests include parallel algorithms, wireless ad hoc networks, and sensor networks.