On Improving Key Pre-distribution Schemes for Sensor Networks

Majid Khabbazian, Ian F. Blake *Fellow, IEEE,* Vijay K. Bhargava, *Fellow, IEEE,* Hosna Jabbari

***Abstract*— In this work, we show how to improve the resilience or computational cost of two primary key pre-distribution schemes. First, we consider the primary key pre-distribution scheme proposed by Eschenauer and Gligor and its extension by Chan, Perrig and Song. We propose a modified version of their schemes and prove that it provides significantly higher resilience than the original schemes at almost no extra cost. The second part of this work deals with the primary key pre-distribution scheme proposed by Blom and its extension by Du, Deng Han and Varshney. The key pre-distribution scheme by Blom and its extension offer much higher resilience than random key pre-distribution schemes at the cost of higher computational cost. We show that the computational cost of the Blom scheme can be significantly reduced at the cost of slight reduction in resilience or a small increase in memory requirement. It is expected that aspects of the techniques introduced here, suitably adapted, can be applied to other key distribution schemes to improve efficiency.**

I. Introduction

Sensor devices (simply called nodes) may be distributed in hostile environments. In such environments, confidential communication has to be encrypted since wireless communication is exposed to interception. This requires establishing pairwise keys between sensor devices. Key establishment in Wireless Sensor Networks (WSNs) is challenging due to the significant limitations of sensor devices in terms of computational power, storage and battery lifetime. In many networks, key establishment is achieved using a Public-Key Infrastructure (PKI). Although some public-key cryptographies such as Elliptic Curve Cryptography (ECC) and RSA have been implemented on small wireless devices [9], [15], they are in general regarded as unsuitable for WSNs mainly because of their expensive computational cost.

An alternative approach proposed for WSNs is to use a Key Pre-Distribution (KPD) scheme. A KPD consists of at least the following two stages: 1) *key preloading stage*, where each node is preloaded with a set of keys before they are deployed; 2) *key discovery stage*, where two nodes attempt to find/compute a pairwise key using information stored in the preloading stage. Note that in the second stage, two nodes may fail to find/compute a pairwise key due to the lack (or insufficient amount) of shared information. An important objective of KPD schemes is to reduce the probability of key establishment failure for the given available memory at each node. Clearly, the size of memory is a fundamental constraint. Suppose there are N nodes in the network. When there is enough memory, every node u_i can store $N-1$ keys, each of which is only known to u_i and u_j, where $1 \leq i,j \leq N$ and $i \neq j$. This trivial key pre-distribution scheme provides a perfect resilience against node capture since capturing any number of nodes does not reveal any information about the pairwise keys between uncaptured nodes. Another trivial key pre-distribution scheme is to preload all the nodes with the same “master key”. This scheme requires the minimum amount of memory but is very vulnerable to node capture.

Practical KPD schemes trade resilience against node capture for reducing memory requirement. They typically require each node to store only a constant number of keys with respect to N. However, as noted in [4], this may not bring infinite scalability as capture of $x \geq x_0$ (for some $x_0 \geq 1$) will compromise a fraction of all the communications between uncaptured nodes. Particularly, it was proven that when the probability of key establishment success is one, KPD schemes are not able to achieve perfect resilience if the number of captured nodes is more than or equal to the number of keys stored in each node [1], [3].

The computational/communication overhead is also an important factor in designing practical KPD schemes. Recall that a PKI is not suitable for WSNs because of its computational/communication overhead. In general, the following criteria are used to evaluate the performance of a KPD.

- Connectivity: Probability of key establishment success.
- Storage: Amount of memory required to store the keys.
- Overhead: The computation/communication cost of key establishment.
- Resilience: The number/fraction of pairwise keys between uncaptured nodes compromised after capturing a given number of nodes.

The aim of this paper is to develop techniques that improve the resilience and overhead of KPD schemes without any significant deterioration in their connectivity or storage requirements. Two primary KPD schemes will be targeted in this endeavor. The first group of KPD schemes considered are the random key pre-distribution scheme proposed in the seminal work by Eschenauer and Gligor [8] and its extension by Chan, Perrig and Song [4]. We prove that a modified version of their scheme provides significantly higher resilience without increasing the storage requirement or decreasing the connectivity. In the second part of the paper, we consider the key generation system proposed in the pioneering work by Blom [2] and its extension by Du, Deng Han and Varshney [7]. The KPD scheme by Blom offers much higher resilience than

M. Khabbazian is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA.

Ian F. Blake and V. K. Bhargava are with the Department of Electrical and Computer Engineering, University of British Columbia, Canada.

the random KPD by Eschenauer and Gligor and its extensions at the cost of higher computational cost. We show that the computational cost of the Blom scheme can be significantly reduced at the cost of slight reduction in resilience or a small increase in memory requirement.

The rest of this paper is organized as follows. In the next section we consider a modification of the q-composite KPD scheme of [4] where the size of the keys stored may be smaller than the c bits, the required security level, while the size of the pairwise established key remains of size c. The resilience of this scheme and the computational overhead required is analyzed and significant overall performance gains of the scheme is noted. In Section III, a technique that significantly reduces the computation required in the nodes is explored. This is achieved by restricting the generator matrix to be a binary matrix which allows the computation of the shared key, an inner product of two vectors in a finite field, to be achieved via simple XOR circuits rather than finite field multiplier circuits. This is achieved at the cost of a slight increase of stored key size for the same level of security and resilience.

II. Improving q-Composite Random Key Pre-distribution Schemes

In their seminal work, Eschenauer and Gligor [8] introduced a random key pre-distribution scheme (called the basic scheme) for wireless sensor networks. In the basic scheme, each node is preloaded with a set of k keys (called key ring) randomly selected (without replacement) from a large key pool of size $P \gg k$. The size of each key is typically between 64 and 128 bits. The probability that two nodes share a key can be easily computed as

$$\mathcal{P}_{connect} = \begin{cases} 1 - \frac{\binom{P-k}{k}}{\binom{P}{k}} & P \geq 2k \\ 1 & P < 2k \end{cases}$$

where $\mathcal{P}_{connect}$ is the probability of key establishment success. Note that in practice the value of P is much larger than $2k$. For example, to achieve $\mathcal{P}_{connect} = 0.5$, k can be as low as $\lceil \sqrt{\ln 2}\sqrt{P} \rceil$.

After deployment, two nodes can discover whether they have a common key by, for example, broadcasting the list of their key identifiers in plaintext. A key identifier is a short integer assigned to each key in the preloading stage. The size of each identifier is $\mathcal{O}(\log(P))$ bits; thus the communication cost of broadcasting key identifiers is $\mathcal{O}(k\log(P))$. Fortunately, using a technique explained in [12], [13] and [16] the communication cost can be reduced to $\mathcal{O}(\log(N))$, where N is the total number of nodes in the network.

In the basic scheme, nodes which have at least one common key identifier can establish a pairwise key by selecting one of their shared keys (if there are more than one) and verifying it through a challenge-response protocol. The verification is required to ensure both nodes essentially hold the same key. The basic scheme was modified to the q-composite random KPD scheme by Chan, Perrig and Song [4]. The q-composite KPD scheme differs from the basic scheme in two ways. First, using the q-composite KPD scheme, two nodes can establish a key if and only if they share ($q > 1$) keys, instead of only one in the basic scheme. Second, rather than using a single shared key as the pairwise key, in the q-composite KPD scheme, nodes use the hash of concatenation of all their shared keys as their pairwise key. In [4], the authors show that their proposed q-composite KPD scheme achieves higher resilience than the basic scheme when small number of nodes have been captured (small scale attack) and is more vulnerable than the basic scheme when a large number of nodes have been compromised. As noted by the authors, this may be a desirable trade-off since small scale attacks are more likely and easier to launch and are harder to detect compared to the large scale attacks. In the next section, we modify the q-composite KPD scheme with the objective of increasing its resilience against node capture. As will be shown, the modified version of the q-composite KPD scheme is significantly more resilient than itself. Interestingly, this improvement comes at almost no extra cost.

A. *Modified q-Composite Key Pre-distribution Scheme*

Let us set the minimum security level to 2^c, or simply, c bits security. The value of c is typically between 64 to 128 bits for WSNs. Suppose each node is capable of storing $m \times c$ bits (i.e., m keys of size c bits). In the basic and the q-composite KPD schemes, c and m are given parameters. For a given value of m, the size of key pool (P) is set such that

$$\mathcal{P}_{connect} \geq \mathcal{P}^* \tag{1}$$

where $\mathcal{P}^*$ is the minimum probability of key establishment success desired between two nodes. For the q-composite KPD scheme we have

$$\mathcal{P}_{connect} = 1 - \sum_{i=0}^{q-1} \mathcal{P}(i), \tag{2}$$

where

$$\mathcal{P}(i) = \frac{\binom{k}{i}\binom{P-k}{k-i}}{\binom{P}{k}} \tag{3}$$

is the probability that two nodes have exactly i keys in common, $k = \lfloor \frac{mc}{c'} \rfloor$ is the number of keys stored in each node and $c' = c$ is the size of each key. Note that, for a fixed value of k, the resilience of the basic and the q-composite KPD schemes increases as P increases. On the other hand, the probability of key establishment success decreases as the size of key pool (P) increases. Therefore, the optimal value of P is the largest integer that satisfies (1). The desirable probability of key establishment success ($\mathcal{P}^*$) is set such that the graph of secure links [1] is connected with some high probability (for example 0.999).

In the basic scheme, the q-composite KPD scheme and their extensions, size of keys (in bits) assigned to the nodes in the preloading stage is equal to c (the size of pairwise key), thus $k = m$. Unlike the existing random KPD schemes, in our proposed modified q-composite KPD scheme, the size of

[1] There is an edge between two nodes if and only if they are in the communication range of each other and can establish a pairwise key.

assigned keys can be any number c', where $1 \leq c' \leq c$, hence $k = \lfloor \frac{m \times c}{c'} \rfloor \geq m$. Since we require c bits security level (the pairwise key has to be at least c bits) two nodes have to share at least $q \geq \lceil \frac{c}{c'} \rceil$ keys in order to securely communicate. The aim of next section is to find values of c', and P which achieve the highest resilience. Note that we now have one more parameter, c', to use towards improving the resilience.

B. Resilience of the Modified q-Composite KPD Scheme Against Node Capture

Let $x \geq 0$ be the number of nodes captured. The probability that a uniformly selected key is not compromised (i.e., does not exist in the key ring of any node captured) is $(1 - \frac{k}{P})^x$. Thus, the probability that a secure link between two uncaptured nodes is compromised is

$$\mathcal{P}_{comp}(x) = \sum_{i=q}^{k} B(x,i) \frac{\mathcal{P}(i)}{\mathcal{P}_{connect}}, \tag{4}$$

where $\mathcal{P}(i)$ and $\mathcal{P}_{connect}$ can be computed using Equations (2) and (3), respectively, and

$$B(x,i) = \sum_{j=0}^{\lceil \frac{c}{c'} \rceil - 1} \binom{i}{j} \left((1 - \frac{k}{P})^x \right)^j \left(1 - (1 - \frac{k}{P})^x \right)^{i-j}$$

is the probability that the number of non-compromised keys is less than $\lceil \frac{c}{c'} \rceil$. The probability that a secure link becomes compromised in the q-composite KPD scheme is the special case of Equation (4) for $c' = c$, i.e.

$$\sum_{i=q}^{k} \left(1 - (1 - \frac{k}{P})^x \right)^i \frac{\mathcal{P}(i)}{\mathcal{P}_{connect}}.$$

Knowing values of q, c' and P, we can use Equation (4) to numerically compute $\mathcal{P}_{comp}(x)$ for a given x. However, it is complicated to use Equation (4) to find the relationship between different parameters. In the following, we analytically study the relationship between different parameters in an attempt to compute the values of the key pool size and c' that maximize resilience.

Set q equal to its minimum value, i.e. $q = \lceil \frac{c}{c'} \rceil$. and suppose u and v are two uncaptured nodes. The probability that a given key in u's key ring is also in v's key ring is $\frac{k}{P}$. Therefore, the average number of shared keys between u and v is

$$\bar{q} = k \times \frac{k}{P} = \frac{k^2}{P}.$$

Let us define the random variable X as the number of shared keys that are not compromised after the capture of x nodes. The expected value of X can be computed as

$$E(X) = k \times \frac{k}{P} \times (1 - \frac{k}{P})^x = \bar{q}(1 - \frac{k}{P})^x$$

because every key of u is shared by v with probability $\frac{k}{P}$ and is not compromised with probability $(1 - \frac{k}{P})^x$. The probability distribution of X is a sum of k nearly independent Bernoulli variables. Therefore, the random variable X approximately follows a binomial distribution with mean $E(X)$; this can be approximated by the normal distribution

$$N\left(\mu_x = E(X), \sigma_x^2 = (1 - \frac{k}{P}(1 - \frac{k}{P})^x) \times E(X) \right)$$

when $\mu_x = E(X)$ and $k - \mu_x$ are greater than, for example 10. In our analysis we are mainly concerned with values of x for which $E(X) \geq \lceil \frac{c}{c'} \rceil$. Also, our modified q-composite KPD scheme deals with small values of c'. Therefore, both $E(X)$, and $k - E(X)$ are typically large enough, hence the probability distribution of X can be approximated by a normal distribution.

We define two critical points x_c and x_s in our analysis. Let x_c be the solution to the equation $E(X) = q$, i.e., x_c is the number of captured nodes for which the average number of non-compromised keys shared between two uncaptured nodes is approximately equal to the minimum number of required keys for key establishment. Since X approximately follows a normal distribution with mean $E(X)$, almost half of all secure communications are compromised if x_c nodes are captured. We consider x_c as the approximation of the point where the whole network is compromised.

A large fraction of secure communications are not compromised if $q \leq u_x - 3\sigma_x$. It is because at least $(1 - \frac{1}{r^2}) \times 100\%$ of the values of a random variable are within r standard deviations from its mean. In particular, for normal distribution, more than 99.7% of all values are within three standard deviations of the mean. In the binomial distribution and its approximated normal distribution, we have

$$\sigma_x^2 = \mu_x \times (1 - \frac{k}{P}(1 - \frac{k}{P})^x).$$

Therefore, $\sigma_x \leq \sqrt{\mu_x}$, thus

$$\mu_x - 3\sigma_x \geq \mu_x - 3\sqrt{\mu}_x = E(X) - 3\sqrt{E(X)}.$$

Suppose x_s is the solution to

$$E(X) - 3\sqrt{E(X)} = q, \quad E(X) \geq q.$$

Therefore, we get

$$E(x_s) = q + 3\sqrt{q + \frac{9}{4}} + \frac{9}{2}.$$

For any real numbers x, $r > 0$, we have

$$(1 - t)^r \leq e^{-rt}.$$

Also, when $0 \leq t \leq 1$ and $r \gg 1$, we can approximate $(1-t)^r$ by e^{-rt}. Using this, we can find approximate values of x_c and x_s as follows.

$$\begin{aligned}
& & E(x_c) = q \\
\Rightarrow & & \bar{q}(1 - \tfrac{k}{P})^{x_c} = q \\
\Rightarrow & & (1 - \tfrac{k}{P})^{x_c} = \tfrac{q}{\bar{q}} \\
\Rightarrow & & x_c \approx -\tfrac{P}{k} \times \ln(\tfrac{q}{\bar{q}}) \\
\Rightarrow & & x_c \approx -\tfrac{P}{k^2} \times k \times \ln(\tfrac{q}{\bar{q}}) \\
\Rightarrow & & x_c \approx -\tfrac{k}{\bar{q}} \times \ln(\tfrac{q}{\bar{q}})
\end{aligned}$$

Hence,

$$x_c \approx \frac{k}{q} \times \frac{\ln \beta}{\beta},$$

where $\frac{\bar{q}}{q} = \beta$. Using a similar approach, we can approximate x_s as

$$x_s \approx \frac{k}{q} \times \frac{\ln\left(\frac{\bar{q}}{q+3\sqrt{q+\frac{9}{4}}+\frac{9}{2}}\right)}{\beta} = \frac{k}{q} \times \frac{\ln(\alpha_q \beta)}{\beta},$$

where

$$\alpha_q = \frac{q}{q+3\sqrt{q+\frac{9}{4}}+\frac{9}{2}}.$$

The function $f(x) = \frac{\ln(\alpha x)}{x}$, $\alpha, x > 0$, is maximized at $x = \frac{e}{\alpha}$. Therefore, x_c and x_s are approximately maximized when $\beta = e$ and $\beta = \frac{e}{\alpha_q}$, respectively. Note that α_q is a non-decreasing function of q and $\alpha_q \to 1$ as $q \to \infty$. Consequently, the maximum resilience increases as q increases. Also, as $q \to \infty$, we get

$$x_s = x_c = \frac{k}{q} \times \frac{1}{e} \approx \frac{m}{e},$$

when we set $\beta = e$. Moreover, by setting $\beta = e$, we get $\mathcal{P}_{connect} \approx 1$. It is because q, the minimum number of required keys to establish a pairwise key, is far (about three times of the standard deviation) from $\bar{q}$, the average number of shared keys. Finally, note that to maximize $q = \lceil \frac{c}{c'} \rceil$, we need to reduce c' to its minimum value, i.e., $c' = 1$.

***Example** 1:* Suppose $m = 200$ and the minimum security bits is $c = 128$. Therefore, each node is able to store $m = 200$ keys of size 128 bits. Let set $c' = 1$. Thus,

$$k = \lfloor \frac{mc}{c'} \rfloor = mc = 200 \times 128 = 25600,$$

and the minimum number of required keys to establish a secure communication is $q = \lceil \frac{c}{c'} \rceil = 128$. The resilience of the KPD scheme is maximized when

$$\bar{q} = \frac{k^2}{P} \approx e \times q.$$

Therefore, the optimal size of the key pool can be approximated as

$$P_{opt} \approx [\frac{k^2}{e \times q}] = 1883543.$$

Since q is sufficiently large, x_c and x_s can be approximated as

$$x_c = \frac{m}{e} = 73.6 \quad \text{and} \quad x_s = \frac{m \ln(e \times \alpha_q)}{e} = 54.1.$$

As shown in Figure 1 the numerical results obtained using Equation (4) confirm analytical results.

***Example** 2:* Suppose $m = 200$ and $c = 128$. The size of key pool is computed as

$$P_{opt} \approx [\frac{k^2}{e \times q}] = [\frac{k^2}{e \times \lceil \frac{c}{c'} \rceil}].$$

Figure 2 shows the resilience of our proposed KPD scheme for different values of c'. As analytically predicted, the best resilience is achieved when $c' = 1$. Note that, almost half of total communications are compromised at $x_c = 73.6$.

***Example** 3:* In this example, we set $c' = 1$, $m = 200$ and $c = 128$ and vary the key pool size. Figure 3 shows numerical results computed for different values of key pool size. Analytically, it was shown that resilience is approximately maximized when

$$P_{opt} \approx [\frac{k^2}{e \times q}] = 1883543.$$

As shown in Figure 3, numerical results verify what was predicted analytically.

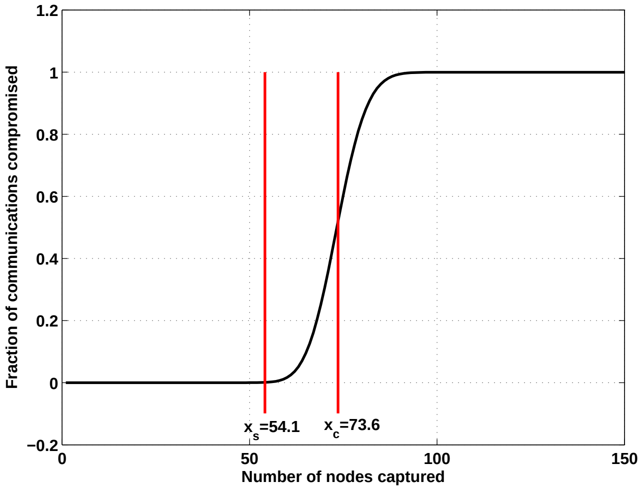

Fig. 1. Probability that a random link between two uncaptured nodes is compromised ($\mathcal{P}_{comp}(x)$). Verifying values of x_s and x_c with numerical results. $m = 200$, $c' = 1$ and $\mathcal{P}_{connect} = 1$.

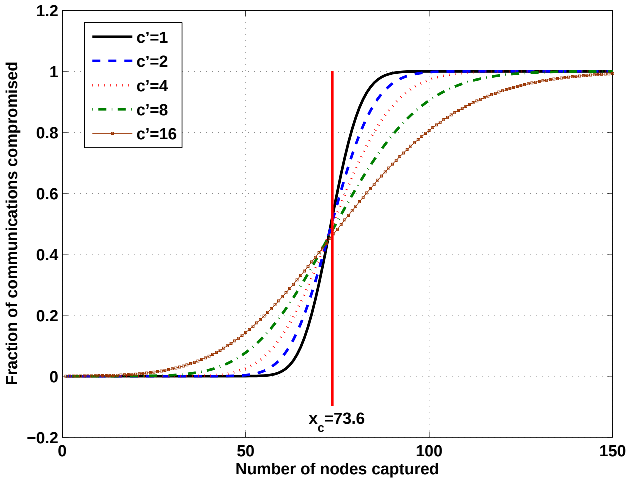

Fig. 2. Computing $\mathcal{P}_{comp}(x)$ for different values of c'.

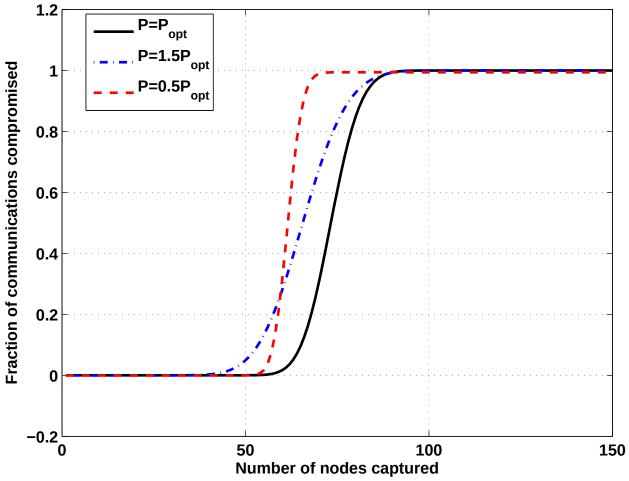

Fig. 3. Computing $\mathcal{P}_{comp}(x)$ for different key pool sizes.

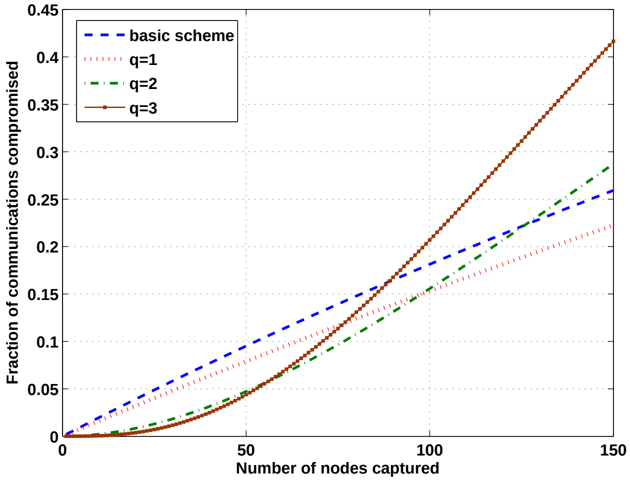

Fig. 4. Probability that a given random link between two uncaptured nodes is compromised. $m = 200$ and $\mathcal{P}_{connect} = 0.33$.

C. Comparison

As mentioned earlier, the advantage of the q-composite KPD scheme over the basic scheme is that it reduces the incentive for small scale attacks. Figure 4 was used as an example in [4] to show that q-composite KPD scheme offers greater resilience when the number of nodes captured is small. To obtain the numerical results shown in Figure 4, we set $m = 200$ and $\mathcal{P}_{connect} = 0.33$. Figure 1 shows the resilience of our modified KPD scheme when $m = 200$. Note that, using our modified scheme, the probability of key establishment is very close to one (i.e. $\mathcal{P}_{connect} \approx 1$). Clearly, our modified KPD scheme achieves higher resilience at lower values of $\mathcal{P}_{connect}$.

To get $\mathcal{P}_{connect} = 0.33$ in our KPD scheme, we can generate three key pools of the same size and assign keys to a node only from one uniformly selected key pool. In this case, the probability that a given key is not compromised is

$$(1-\frac{k}{3\times P})^x$$

and the probability that two nodes share exactly i keys is

$$\mathcal{P}'(i) = \frac{1}{3}\times\mathcal{P}(i) = \frac{1}{3}\times\frac{\binom{k}{i}\binom{P-k}{k-i}}{\binom{P}{k}}$$

Thus,

$$\mathcal{P}'_{comp}(x) = \sum_{i=q}^{k} B'(x,i)\frac{\mathcal{P}'(i)}{\mathcal{P}'_{connect}}, \tag{5}$$

where

$$\mathcal{P}'_{connect} = \frac{1}{3}(1-\sum_{i=0}^{q-1}\mathcal{P}(i)) = \frac{1}{3}\times\mathcal{P}_{connect},$$

and

$$B'(x,i) = \sum_{j=0}^{\lceil\frac{c}{c'}\rceil-1}\binom{i}{j}\left((1-\frac{k}{3P})^x\right)^j\left(1-(1-\frac{k}{3P})^x\right)^{i-j}$$

Since $\mathcal{P}_{connect} \approx 1$ we get

$$\mathcal{P}'{}_{connect} \approx \frac{1}{3} \approx 0.33.$$

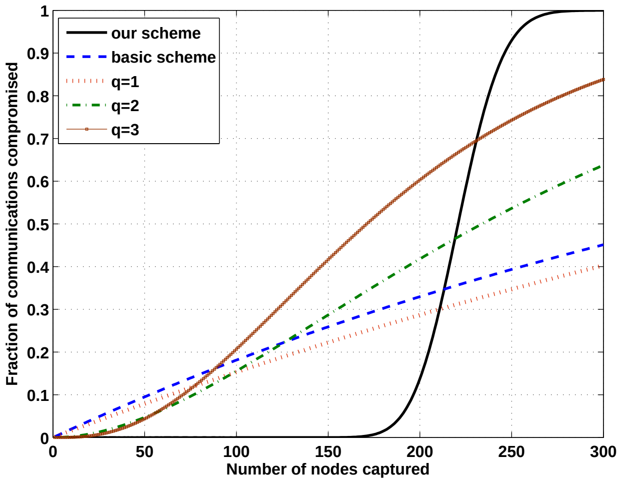

Fig. 5. Comparing the resilience of our scheme with that of the q-composite and basic shames. $m = 200$ and $\mathcal{P}_{connect} = 0.33$.

Note that

$$(1-\frac{k}{3P})^{3x} \approx (1-\frac{k}{P})^x.$$

Thus

$$\mathcal{P}'_{comp}(3x) \approx \mathcal{P}_{comp}(x).$$

In other words, when three key pools are used, the attacker has to capture about three times more nodes in order to compromise the same fraction of secure communications than in the case of having one key pool. It is because, on average, only key rings of one third of all the nodes captured can be employed to break a given secure link between two uncaptured nodes.

Figure 5, compares the resilience of our modified KPD scheme with that of the q-composite KPD scheme ($q = 1$, $q = 2$ and $q = 3$) and the basic scheme. In all schemes, we set $m = 200$ and $\mathcal{P}_{connect} = 0.33$. As shown in Figure 5, our scheme provides nearly perfect security when the number of nodes captured is less than 170, thus it substantially reduces the incentive of small-scale attacks. Interestingly, this improvement is not achieved at the cost of higher memory requirement. Finally, note that the multipath key reinforcement technique proposed by Chan, Perrig and Song [4] can be similarly used in our modified KPD to further improve the resilience against node capture.

D. Overhead of Shared Key Discovery

As mentioned earlier, in the basic and q-composite KPD schemes, two nodes can find all the keys they share through exchanging the list of their keys' identifiers. The communication overhead of broadcasting key identifiers is $\mathcal{O}(k\log(P))$. Clearly, this approach may not be suitable for our modified KPD scheme since values of k and P could be very large in our KPD scheme.

As suggested in [12], [13] and [16], the communication complexity of shared key discovery can be reduced to $\mathcal{O}(\log(N))$ by having each node broadcast only its ID. In this approach, the node ID is used as the input of a pseudorandom generator whose output is the list of key identifiers of the given node's ID. Using this approach, our KPD scheme has the same communication complexity as the basic and q-composite KPD schemes. Another advantage of using this scheme is that nodes are not required to store k key identifiers. This comes at the cost of computing the pseudorandom function (for two different inputs) per each shared key discovery.

III. Improving Blom's Key Pre-distribution Scheme

In his pioneering work [2], Blom described a key predistribution scheme that allows any pair of nodes to establish a pairwise key. Suppose N is the maximum number of nodes in the network and $q > N$ is a prime power. In the preloading stage of the Blom KPD scheme, the server generates a public $(\lambda+1)\times N$ matrix G and a private $(\lambda+1)\times(\lambda+1)$ matrix D over a finite field F_q and computes $A = (DG)^T$, where $(DG)^T$ is the transpose of DG. The matrix D is a (non-singular) symmetric matrix whose elements are chosen uniformly at random from F_q. Each node u_i, $1 \le i \le N$, is preloaded with the ith row of matrix A, $A_{i,:}$, and ith column of matrix G, $G_{:,i}$. Let $K = AG$. Since D is symmetric we have

$$K^T = (AG)^T = G^T(DG) = G^T D^T G = AG = K,$$

thus K is a symmetric matrix, i.e. $K_{i,j} = K_{j,i}$. In the Blom KPD scheme, $K_{i,j}$ (or $K_{j,i}$) is used as the pairwise key between u_i and u_j. In the key establishment stage, nodes u_i and u_j exchange their public information, i.e. their columns of G, and compute the pairwise key as

$$K_{i,j} =< A_{i,:}, G_{:,j} >=< A_{j,:}, G_{:,i} >,$$

where $< A_{i,:}, G_{:,j} >$ denotes the inner product. If every $\lambda+1$ columns of the generator matrix G are linearly independent, it can be proven that no information about the pairwise keys between uncaptured nodes is revealed if the number of nodes captured is at most λ [2], [7]. In this case, the KPD scheme is called λ-secure. Let g be a primitive element in F_q. It can be shown that when $q > N$, every $\lambda+1$ columns of the following van der Monde matrix are linearly independent [11]. An advantage of using a van der Monde matrix as the generator matrix is that the node u_i only needs to store g^i, instead of the ith column of G ($G_{:,i}$), since $G_{:,i}$ can be constructed given g^i.

$$G = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ g & g^2 & g^3 & \dots & g^N \\ g^2 & (g^2)^2 & (g^2)^3 & \dots & (g^2)^N \\ & & & \vdots & \\ g^\lambda & (g^\lambda)^2 & (g^\lambda)^3 & \dots & (g^\lambda)^N \end{bmatrix}$$

In the Blom's KPD scheme, any pair of nodes can establish a pairwise key, thus $\mathcal{P}_{connect} = 1$ and the graph of secure links is a complete graph. To increase resilience of the Blom's KPD scheme, Du, Deng Han and Varshney [7] used the fact that the graph of secure links only needs to be connected although full connectivity is desirable and achieved higher resilience by combining the Blom's KPD scheme with the basic scheme of the previous section. In particular, in their scheme, the server constructs ω key spaces $S_i = (D_i, G)$, $1 \le i \le \omega$ by creating a van der Monde matrix G of size $(\lambda+1)\times N$ and ω symmetric matrices $D_1, \dots, D_\omega$ of size $(\lambda+1)\times(\lambda+1)$. The server loads u_i with the information about $2 \le \tau \le \omega$ randomly selected key spaces, $S_{i_1}, \dots, S_{i_\tau}$. Therefore, each node is required to store $m = (\lambda+1)\tau$ keys of size c bits, where c is the required security level of the pairwise key.

In the key discovery stage, node u_i broadcasts the indices of spaces it carries as well as g^i. Two nodes can directly establish a pairwise key if and only if they carry information from a common key space. To establish a key with u_i, u_j has to compute $G_{:,i}$ and the inner product $< (D_l G)^T_{j,:}, G_{:,i} >$, where l is the index of the common key space between u_i and u_j. The dominating computational cost of this operation is 2λ modular (finite field) multiplications in F_q. Our purpose is to show how this cost can be reduced without undue sacrifice of security or resilience.

The Blom's KPD scheme and its extension [7] achieve significantly higher resilience than the basic and q-composite schemes at the cost of large computational overhead. In this section, we show how to significantly reduce the computational cost of the Blom's KPD scheme (and its extensions) at the cost of a slight decrease in resilience or a slight increase in memory requirement. The main idea, is to use random (binary) linear matrices for the G matrix instead of van der Monde matrices over F_q. Some care is required however to ensure the required properties.

Neither Blom [2] nor Du et al [7] discuss the matrices D (or D_i respectively). It should be noted these symmetric matrices over F_q should be nonsingular. If they are not then it is likely that certain sets of $k < (\lambda+1)$ columns of the matrix $(DG)^T$ may well be dependent, leading to a lower than expected resilience.

A. Observations on random matrices over a finite field

The subject of random matrices over a finite field, especially over the binary field F_2, has been well investigated. Only a small number of results are needed for this work and in this section we draw on results from [5], [6], [10] and [14] without proof or extensive comment. We also adapt the notation in these references to match that used in this work.

Let G be a $(\lambda+1+m)\times N$, $m>0$, binary matrix with each element chosen independently and identically at random, each element being a 1 with probability p (and 0 with probability $1-p$). For reasons noted in [14] the only restriction on p is that it not to be too small, namely $2\ln(\lambda)/\lambda < p < 1-2\ln(\lambda)/\lambda$. The rank properties of the matrices are quite insensitive for such values. It is observed in [14] (using results from [5], [10] and [6]) that the probability that such a binary random $(\lambda+1+m)\times(\lambda+1),\ m>0$ matrix is of full rank $(\lambda+1)$ is given by:

$$Q_m = \prod_{i=m+1}^{\infty}\left(1-\frac{1}{2^i}\right), \quad m>0. \tag{6}$$

Note that while the result assumes $\lambda\rightarrow\infty$, extensive simulation has shown it to hold for even small values of λ [14].

Thus if the $(\lambda+1+m)\times N$ random binary matrix G is generated in this manner it is concluded that any set of $(\lambda+1)$ columns will be of full rank with high probability i.e. linearly independent. The condition of the matrix being of full rank is a slightly different one from the condition of all sets of $\lambda+1$ columns being independent. However, once having generated the random matrix G with N columns, in the random model assumed here, any particular set of $(\lambda+1)$ columns will behave approximately independently of other such sets. From large numbers of simulations conducted on this question in [14], the condition holds, even for small values of λ of the order of ten or more.

Thus the cost of achieving the linear independence of any set of $(\lambda+1)$ columns of the binary matrix, to ensure the same λ-resiliency of the van der Monde matrix previously constructed over F_q, the cost is to increase the length of the columns of G by m. In addition, the whole column may need to be stored in the nodes, rather than just the first element, but the stored elements are binary rather than from the (fairly large) finite field. As discussed in [14] the average value of m required is 1.60669515... However to ensure a probability greater than .999 of the linear independence (full rank) of the sets $(\lambda+1)$ columns a value of $m\geq 10$ will suffice. It is somewhat surprising that the full rank property required is achieved with such a small value.

The KPD scheme described requires the construction of the symmetric $(\lambda+1)\times(\lambda+1)$ matrix D (or D_i in the extended scheme) over the finite field F_q where $q>N$. This problem was not addressed in detail in the references [2], [7]. As noted we should insist the matrix be nonsingular to ensure resiliency properties of the scheme.

Assuming q large, one method might be to choose $(\lambda+1)$ elements from F_q at random and form a van der Monde matrix H from them. Since H is not symmetric, form the matrix $D=H+H^T$ and check to see if it is nonsingular. If not one could repeat the procedure until success. Notice that the matrix H need not be nonsingular in order for the symmetric matrix $D=H+H^T$ be nonsingular - the relationship of the ranks between the two matrices seems in general complicated. Since these computations are taking place in the server, the computational issues are not regarded as too heavy.

Similarly one could generate a random $(\lambda+1)\times(\lambda+1)$ matrix over F_q where, as noted, q is large, the nonzero elements being chosen equiprobable. The probability such a matrix is nonsingular is give by [5]:

$$\pi_q = \prod_{j=1}^{\infty}\left(1-\left(\frac{1}{q}\right)^j\right). \tag{7}$$

While the expression was determined as the size of the matrix tends to infinity, it is known to be an excellent approximation for quite small sizes of matrices. The expression converges to unity rapidly with q [5]:

q	2	3	5	7
π_q	.28878809	.56012607	.76033279	.8367954

As before a procedure to generate suitable candidates for the matrix D is to first generate a random matrix H over F_q and form the symmetric matrix $D=H+H^T$. While the matrix D is no longer random (being symmetric), the upper half triangle of it does contain random elements of F_q and it seems plausible the expression Equation (7) is an approximation to the probability of it being nonsingular. Regardless of this approximation, one can test the nonsingularity of the resulting matrix and repeat the procedure until successful. The expression Equation (7) gives an indication of the number of attempts one might have to make before success. For the ω spaces the procedure is repeated to obtain the ω matrices D_i.

B. Application of random matrices to the extended Blom scheme

The method of generating the required matrices in the extended Blom scheme described is straight forward. The point of having the G matrix binary is that the computation of the joint key in the nodes is greatly simplified. Similar to the case of using a van der Monde matrix, a node only needs to store a seed (for example its ID) since, given the seed, the entire column of G can be generated using a public pseudo random generator. Once the common key space is established however, the computation of the inner product

$$< (D_\ell G)^T_{j,:}, G_{:,i} >$$

is trivial. In particular, if the finite field F_q is chosen to have characteristic two ($q=2^\ell$), computing this inner product is a matter of computing the parity of each coordinate position of the elements of the row of $(D_\ell G)^T$ dictated by the nonzero elements of the column of $G_{:,i}$. This is easily achieved with a single XOR cell for each coordinate position. Thus the cost of simplifying the computation of the joint key in the nodes with this technique is to store $m\approx 10$ bits more per key in the nodes.

One might also consider allowing the elements of the matrix G to be selected uniformly at random from a subset $S\subset F_q$, chosen so that computation of the inner product is still very simple. Without loss of generality, assume that nodes

$u_1, u_2, \dots, u_x$, $x \geq 1$ have been captured. Clearly, the pairwise key between two nodes u_i and u_j is compromised if either $G_{:,i}$ or $G_{:,j}$ is a linear combination of $G_{:,1}, G_{:,2}, \dots, G_{:,x}$. Also, capture of nodes $u_1, u_2, \dots, u_x$ reveals no information about the pairwise key between u_i and u_j if neither $G_{:,i}$ nor $G_{:,j}$ is a linear combination of $G_{:,1}, G_{:,2}, \dots, G_{:,x}$.

Theorem *1:* Let u_i be an uncaptured node (i.e. $x < i \leq N$). The probability that $G_{:,i}$ is a linear combination of $G_{:,1}, G_{:,2}, \dots, G_{:,x}$ is at most $(\frac{1}{|\mathcal{S}|})^{\lambda+1-x}$, where $x \leq \lambda + 1$ and $|\mathcal{S}|$ denotes the cardinality of the set $\mathcal{S}$.

Proof: Without loss of generality assume that $G_{:,1}, G_{:,2}, \dots, G_{:,x}$ are independent. By performing elementary column operations on the $(\lambda+1) \times x$ matrix $[G_{:,1} \dots G_{:,x}]$ we can obtain a $(\lambda + 1) \times x$ matrix $G' = [G'_{:,1} \dots G'_{:,x}]$ such that for some $1 \leq t_1, \dots, t_x \leq \lambda + 1$

$$[(G'_{t_1,:})^T (G'_{t_2,:})^T \dots (G'_{t_x,:})^T] = I_{x,x}, \tag{8}$$

where $I_{x,x}$ denotes the identity matrix of size $x \times x$. Since $[G'_{:,1} \dots G'_{:,x}]$ is constructed by performing elementary column operations, $G_{:,i}$ is a linear combination of $G_{:,1}, G_{:,2}, \dots, G_{:,x}$, i.e.

$$G_{:,i} = a_1 G_{:,1} + a_2 G_{:,2} + \dots + a_x G_{:,x},$$

if and only if it is a linear combination of $G'_{:,1}, G'_{:,2}, \dots, G'_{:,x}$, i.e.

$$G_{:,i} = b_1 G'_{:,1} + b_2 G'_{:,2} + \dots + b_x G'_{:,x}.$$

Note that $b_1, \dots, b_x$ has to be from the set $\mathcal{S}$ because of Property 8 and the fact that the elements of $G_{:,i}$ are from $\mathcal{S}$. The total number of vectors that can be written as

$$b_1 G'_{:,1} + b_2 G'_{:,2} + \dots + b_x G'_{:,x},$$

where $b_1, \dots b_x \in \mathcal{S}$, is at most $|\mathcal{S}|^x$. [2] The vector $G_{:,i}$ is a random vector from a vector set of size $|\mathcal{S}|^{\lambda+1}$. Consequently, the probability that $G_{:,i}$ is a linear combination of $G'_{:,1}, G'_{:,2}, \dots, G'_{:,x}$ (and hence a linear combination of $G_{:,1}, G_{:,2}, \dots, G_{:,x}$) is at most

$$\frac{|\mathcal{S}|^x}{|\mathcal{S}|^{\lambda+1}} = \frac{1}{|\mathcal{S}|^{\lambda+1-x}}.$$

■

Based on Theorem 1, the probability that the pairwise key between two uncaptured nodes u_i and u_j is compromised is at most

$$1 - (1 - (\frac{1}{|\mathcal{S}|})^{\lambda+1-x})^2,$$

when $x < \lambda + 1$ and is one when $x \geq \lambda + 1$ For example when $|\mathcal{S}| = 8$ and $\lambda + 1 = 200$, a pairwise key between two uncaptured node is secure with probability at least 0.999 if $x < 197$. Therefore, for relatively small values of $|\mathcal{S}|$, the modified Blom KPD scheme provides almost the same resiliency as the original scheme. We can carefully choose the finite field F_q and S so the inner product is still very simple. One approach is to use F_p, where p is a Mersenne prime (a prime which is one less than a power of two) and choose a

$$\mathcal{S} \subseteq \{0, 2^0, 2^1, 2^2, \dots 2^{\lfloor \log_2(p) \rfloor}\}.$$

In this case, computing the inner product only requires (circular) shifts instead of modular multiplications. Note that there are only four Mersenne primes whose size is less than 128 bits. These primes are

$$2^2 - 1, 2^7 - 1, 2^{31} - 1 \text{ and } 2^{127} - 1.$$

If the security level is, for example, 64 bits then we can choose $p = 2^7 - 1$, In this case, to achieve 64 bits security, we can generate a single generator matrix G and 10 symmetric matrices $D_1, \dots, D_{10}$. Then, every node u_i will be loaded with $(D_1 G)^T_{:,i}, \dots, (D_{10} G)^T_{:,i}$ and $G_{i,:}$. To compute a pairwise key, a node has to compute 10 inner products and combine the results. Note that, the computational cost of 10 inner products when the vector elements are 7 bits is approximately the same as the computational cost of one inner product when the vector elements are 70 bits.

IV. Conclusions

The paper has considered techniques for improving the efficiency and performance of pre-distribution schemes. A modification of the q-composite KPD scheme of [4] was considered that allowed the size of the keys stored to be smaller than the c bits, the required security level, while the size of the pairwise established key remained of size c. The resilience of this scheme and the computational overhead required was analyzed and shown to give performance gains. In Section III, a technique that significantly reduces the computation required in the nodes was explored. This involved restricting the matrix G to be a binary matrix which allowed the computation of the shared key, an inner product of two vectors in a finite field, to be achieved via simple XOR circuits rather than finite field multiplier circuits. The improvement in the computational requirements was obtained at the cost of a slight increase of stored key size for the same level of security.

While the two specific schemes of [7] and [8] were discussed the authors believe that similar considerations, appropriately adapted, would apply to other key pre-distribution schemes, leading to similar performance enhancements for them as well.

References

[1] A. Beimel and B. Chor. Interaction in key distribution schemes. Advances in Cryptology, Springer-Verlag, vol. 773, pp. 444-457, 1994.

[2] R. Blom. An optimal class of symmetric key generation systems. Proceedings of EUROCRYPT'84, Springer-Verlag LNCS vol. 209, pp. 335-338, 1985.

[3] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung. Perfectly-secure key distribution for dynamic conferences. Lecture Notes in Computer Science, vol. 740, pp. 471-486, 1993.

[4] H. Chan, A. Perrig and D. Song. Random key predistribution schemes for sensor networks. IEEE Symposium on Security and Privacy, pp. 197-213, May 2003.

[5] C. Cooper. On the rank of random matrices. Random Structures and Algorithms, vol. 16, pp. 209-232, 2000.

[6] C. Cooper. On the distribution of rank of a random matrix over a finite field. Random Structures and Algorithms, vol. 17, pp. 197-212, 2000.

[7] W. Du, J. Deng, Y.S. Han and P.K. Varshney. A pairwise key predistribution scheme for wireless sensor networks. Proceedings ACM Conference CCS'03, pp. 1-10, October, 2003.

[2] Note that not all such vectors have all elements in $\mathcal{S}$.

[8] L. Eschenauer and V.D. Gligor. A key-management scheme for distributed sensor networks. Proceedings ACM Conference CCS'02, pp. 41-47, November, 2003.

[9] N. Gura, A. Patel, and A. Wander and H. Eberle and S.C. Shantz. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. Cryptographic Hardware and Embedded Systems(CHES), pp. 119-132, August, 2004

[10] V.F. Kolchin. Random Graphs. Cambridge University Press, Cambridge, 1999.

[11] F.J. MacWilliams and N.J.A. Sloane. The theory of error-correcting codes. New York, NY: Elsevier Science Publishing Company, Inc., 1977.

[12] E.C. Park and Ian F. Blake. Reducing communication overhead of key distribution schemes for wireless sensor networks. International Conference on Computer Communications and Networks, pp. 1345-1350, 2007.

[13] R.D. Pietro, L.V. Mancini and A. Mei. Random key assignment for secure wireless sensor networks. ACM Workshop on Security of Ad Hoc and Sensor Networks, 2003.

[14] C. Studholme and Ian F. Blake. Random matrices and modes for the erasure channel. Algorithmica, 2008 .

[15] R. Watro, D. Kong, S-F Cuti, C. Gardiner, C. Lynn and P. Kruus. TinyPK: securing sensor networks with public key technology. Proceedings ACM workshop on Security of ad hoc and sensor networks, pp. 59-64, 2004.

[16] S. Zhu, S. Xu, S. Setia and S. Jajodia, Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach. IEEE International Conference on Network Protocols (ICNP), 2003.