A GENERALISED FORMULA FOR CALCULATING THE RESILIENCE OF RANDOM KEY PREDISTRIBUTION SCHEMES

ED KENDALL, MICHELLE KENDALL, AND WILFRID S. KENDALL

Abstract. A commonly used metric for comparing the resilience of key predistribution schemes is fail_s, which measures the proportion of network connections which are 'broken' by an adversary which has compromised s nodes. In 'Random key predistribution schemes for sensor networks', Chan, Perrig and Song present a formula for measuring the resilience in a class of random key predistribution schemes called q-composite schemes. We present a correction to this formula for schemes where more than one key may be used to secure a link between a pair of nodes. Our corrected formula features an additional parameter which makes it applicable to a wider variety of random key predistribution schemes, including the original Eschenauer Gligor scheme. We also present a simplification of the formula for calculating connectivity.

We refer to the recent paper by Yum and Lee which also claims to correct the original formula for the q-composite scheme. However the resulting formula is complicated, computationally demanding, and hard to understand. The formula which we propose and prove is easily computable and can be applied to a wider range of schemes.

1. Introduction

In large, resource-constrained networks such as wireless sensor networks, cryptographic protection of communications can be a non-trivial task. *Key predistribution schemes* (KPSs) are methods for allocating keys to the devices or 'nodes' of a network before they are deployed into their chosen environment. They are designed for networks where public key cryptography would be too computationally demanding for the nodes, and so instead equip nodes with symmetric keys. All KPSs make trade-offs between the number of keys which nodes have to store, the connectivity of the network, and the resilience against an adversary. Accordingly, the metrics which are typically used for key predistribution schemes are:

- Key storage - the number of keys which each node is required to store. In the schemes we will be considering, the key storage will be constant, and we will denote it by k.
- Connectivity - the proportion of nodes which are 'connected' by sharing keys. A common way to measure this is *local connectivity* Pr_1, which is the probability that a randomly-chosen pair of nodes share at least $q \geq 1$ keys, where q is an intersection threshold dicated by the KPS. Many KPSs only require nodes to have a single key in common in order to be connected.
- Resilience - the proportion of node-node connections 'broken' by an adversary which has compromised s nodes. A common measure of resilience is fail_s, which is defined to be the probability that a randomly-chosen link between a pair of uncompromised nodes is broken after the adversary has compromised s nodes. By 'broken', we mean that the key or keys securing that link are all known to the adversary. Notice that this is a conditional probability, conditioning on the two nodes in question being connected.

  Some papers such as [1, 2] calculate the resilience in this way without calling it 'fail_s'. If $\mathsf{fail}_s = 0$ for all $1 \leq s \leq v-2$ then it is said that the network has *perfect resilience*. We note that lower values of fail_s represent better resilience.

We model the adversary by assuming that nodes are compromised at random. It is of course possible in practice that the adversary could employ a better strategy, for example by targeting two nodes which appear not to be communicating with each other, in the hope of learning $2k$ keys. (If the nodes were communicating then they must share at least q keys, and so the adversary would learn at most $2k - q$ keys by their compromise.) The random adversary model can therefore be thought of as calculating a lower bound on fail_s, hence an upper bound on the resilience, and is useful as a metric for comparison of KPSs.

Correct analysis of schemes is fundamental to proper assessment of KPSs. We present an adjustment to the formula given in [1] for the resilience of random key predistribution schemes where nodes may use more than one key to secure their connections. In Section 2 we give the details of random key predistribution schemes and demonstrate the proofs of their connectivity and resilience parameters. In Section 3 we state the previously proposed formulae for the resilience of q-composite schemes and discuss issues arising in their proofs, before presenting and proving our generalised formula for fail_s in Section 4. It is rigorously proven, may be applied to a range of random key predistribution schemes and is more easily computable than that given in [3]. Finally, in Section 5 we analyse the difference between our formula and that given in [1], which can be considered an upper bound on the true value.

2. Background - random key predistribution schemes

Key predistribution schemes can be deterministic or random. In deterministic schemes, fail_s can usually be computed using exact knowledge of how many nodes store each key. In [4], Paterson and Stinson generalise the fail_s calculation across a range of deterministic schemes.

Here we present two examples of random key predistribution schemes. We derive their respective connectivity and resilience calculations in order to demonstrate some of the methods for proving our main result, the generalised formula for fail_s in random key predistribution schemes. We also provide a simplified formula for the probability of two nodes having exactly i keys in common.

Scheme 1 (**Eschenauer Gligor Random Key Predistribution**)**.** The seminal paper by Eschenauer and Gligor [5] presented the first randomised approach to key predistribution. The scheme is straightforward: a *key pool* $\mathcal{K}$ of n symmetric keys is generated from the space of all possible keys. Each node is independently assigned a randomly selected k-subset of keys from the key pool. (That is, each node stores k distinct keys; for each node the keys are chosen without replacement.) Nodes are deployed into the environment and use a *shared key discovery* protocol such as those described in [6, 7] to identify the other nodes with which they share keys.

Two nodes are said to be 'connected' if they have at least one key in common. If they have more than one key in common, they should select a single one of their common keys at random to use to secure their communications. To be precise, we introduce a parameter d which is the maximum number of common keys which two nodes may use to secure their communications. For the Eschenauer Gligor scheme, $d = 1$.

We now present the probability Pr_1 of two nodes being connected in this scheme. The original paper presents and proves an equivalent expression of this formula using factorials; we use the binomial coefficient notation for consistency with the majority of later literature.

Lemma 1 (Eschenauer Gligor connectivity)**.** *The probability of two nodes being connected in an Eschenauer Gligor random key predistribution scheme with key pool size n and key storage k is*

$$\mathsf{Pr}_1 = 1 - \frac{\binom{n-k}{k}}{\binom{n}{k}} \quad .$$

Proof. Suppose that two nodes N_i, N_j store key sets U_i, U_j respectively. The probability that they are connected is

$$1 - \mathsf{Pr}[\text{they have no keys in common}] = 1 - \mathsf{Pr}[U_i \cap U_j = \emptyset].$$

Fix U_i. Then there are $\binom{n-k}{k}$ ways to pick a k-subset of keys for node N_j so that $U_i \cap U_j = \emptyset$, out of the total possible $\binom{n}{k}$ ways to pick U_j. □

Eschenauer and Gligor do not calculate fail_s in the way that we have defined. They do, however, make the observation that in a simulation, only 50% of the keys from the key pool were used to secure links: 30% were used to secure a single link, 10% to secure two links and 5% to secure three links. Thus the compromise of a single key compromises one other link with probability 0.1.

The standard metric fail_s for Eschenauer Gligor schemes is indirectly stated within another result in [1]. Here we state and prove it formally.

Lemma 2 (Eschenauer Gligor resilience)**.** *In an Eschenauer Gligor random key predistribution scheme with key pool size n and key storage k, the resilience is given by*

$$\mathsf{fail}_s = 1 - \left(1 - \frac{k}{n}\right)^s \quad . \tag{1}$$

Proof. Fix a random link in the network between uncompromised nodes N_i and N_j, and suppose that they use key κ_i to secure their connection.

We begin by considering $s = 1$, that is, the adversary has compromised a single node. Let X be a uniformly random k-subset of the key pool $\mathcal{K} = \{\kappa_1, \ldots, \kappa_n\}$, so that it represents the keys known to the adversary after compromising one node. Then

$$\begin{aligned}
\mathsf{fail}_1 &= \mathsf{Pr}[\kappa_i \in X] \\
&= 1 - \mathsf{Pr}[\kappa_i \notin X] \\
&= 1 - \frac{\binom{n-1}{k}}{\binom{n}{k}} \\
&= 1 - \left(1 - \frac{k}{n}\right) \quad .
\end{aligned}$$

Now we generalise for $s > 1$. Let $X_1, \ldots, X_s$ be independent uniformly random subsets of the key pool, each of size k. Then

$$\begin{aligned}
\mathsf{fail}_s &= \mathsf{Pr}[\kappa_i \in X_1 \cup \cdots \cup X_s] \\
&= 1 - \mathsf{Pr}[\kappa_i \notin X_1 \cup \cdots \cup X_s] \\
&= 1 - (\mathsf{Pr}[\kappa_i \notin X_1])^s \\
&= 1 - \left(1 - \frac{k}{n}\right)^s \quad .
\end{aligned}$$

□

In many key predistribution schemes, it is possible that a pair of nodes N_i and N_j have more than one key in common. If $d = k$, nodes may use all of their $c \leq k$ common keys $U_i \cap U_j = \kappa_{t_1}, \kappa_{t_2}, \ldots, \kappa_{t_c}$ to secure the link, for example by calculating their shared key to be

$$\kappa_{ij} = h(\kappa_{t_1} || \kappa_{t_2} || \cdots || \kappa_{t_c}) \ ,$$

where h is a suitable function such as a hash function (see chapter 4 of [8] for an introduction to hash functions), and where there is a well-defined ordering on the keys $t_1 < t_2 < \cdots < t_c$ so that κ_{ij} is uniquely defined. Since an adversary would have to learn all c keys to compromise the link, such schemes have better resilience than those where $d = 1$, such as Scheme 1. However, changing d does not affect the connectivity.

Chan et al. present a random KPS which requires nodes to have $q > 1$ keys in common in order to be connected. We give the formal details of their scheme below. Intuitively, for the same key pool size n and key storage k, nodes are less likely to be connected in the Chan et al. scheme than the Eschenauer Gligor scheme, but the resilience increases with q. Such a trade-off may be advantageous for some applications, and the sizes of n, k and q can be adapted to provide a desirable level of connectivity with as high a resilience as possible.

Scheme 2 (**q-composite scheme**). In [1], Chan et al. present the q-composite scheme. It is similar to Scheme 1, except that nodes must share at least $q > 1$ keys before they are allowed to compute a common key, and $d = k$. That is, nodes with fewer than q keys in common will not be able to communicate directly, and nodes with q or more keys in common should hash all of their common keys to create their link key.

We begin by considering the probability of connectivity in the q-composite scheme, that is, the probability that a pair of nodes share q or more keys. We omit the full proof here because it is given in [1] and reproduced in [3]. However, we provide an improvement: the value of $p(i)$, the probability of a pair of nodes sharing exactly i keys, has previously been given as

$$p(i) = \frac{\binom{n}{i}\binom{n-i}{2(k-i)}\binom{2(k-i)}{k-i}}{\binom{n}{k}^2} \ ,$$

but we state an equivalent, simpler expression. Our formula for $p(i)$ can be derived from the original by expanding the binomial coefficients and rearranging, but we provide a direct combinatorial proof.

Theorem 1. *In a q-composite scheme with key pool size n and key storage k, the connectivity probability is*

$$\mathsf{Pr}_1 = 1 - \sum_{i=0}^{q-1} p(i) \ , \tag{2}$$

where

$$p(i) = \frac{\binom{n-k}{k-i}\binom{k}{i}}{\binom{n}{k}} \tag{3}$$

is the probability of a pair of nodes sharing exactly i keys.

Proof of (3). We consider the probability of two nodes N_1 and N_2 having exactly i keys in common. Fix i keys from N_1's set of keys. For N_2 to have $(k-i)$ keys which are *unknown* to N_1, it must have $(k-i)$ keys chosen from the $(n-k)$ keys unknown to N_1. Thus there are $\binom{n-k}{k-i}$ ways to do this, out of the $\binom{n}{k}$ ways to choose keys for N_2. Finally, we multiply by the number of ways to fix i keys from N_1's set of k keys. □

Notice that (2) is a generalised formula for the probability of connectivity, which agrees with that given in Lemma 1 for the Eschenauer Gligor scheme: setting $q = 1$ into (2) gives

$$\begin{aligned} \mathsf{Pr}_1 &= 1 - p(0) \\ &= 1 - \frac{\binom{n-k}{k}}{\binom{n}{k}} \ . \end{aligned}$$

3. Previous formulae for the resilience of the q-composite scheme

We now discuss approaches which have been proposed for calculating the resilience of the q-composite scheme.

3.1. **Chan, Perrig and Song.** In [1], Chan et al. give the following formula:

$$\mathsf{fail}_s = \sum_{i=q}^{k} \left(1 - \left(1 - \frac{k}{n}\right)^s\right)^i \frac{p(i)}{\mathsf{Pr}_1} \ . \tag{4}$$

However, the proof is informal and incorrectly assumes independence between certain events, as we explain below.

Before we explain why this formula for resilience is incorrect, we first note an aspect of the notation in the original formula which has caused some confusion in the subsequent literature. Chan et al. consider a parameter p, defined to be the minimum node-node connectivity probability needed to make the whole network connected with some high probability. They then define $p_{connect} = 1 - \sum_{i=0}^{q-1} p(i)$ and state that the key pool size n should be chosen to be the largest (integer) such that $p_{connect} \geq p$, which is a sensible way to reduce unnecessary connectivity and keep resilience high. However, in their resilience formula, they redefine p to equal $p_{connect}$. This has caused errors to be made in its reproduction, for example in [3]. We will always use the notation Pr_1 as defined in (2) to avoid confusion, and for consistency with much of the deterministic key predistribution literature.

As Yum and Lee point out in [3], the problem with (4) is an incorrect assumption of independence. Suppose that, in a 2-composite scheme, a pair of nodes share keys κ_1 and κ_2. For an adversary to break the link between these nodes requires knowledge of both κ_1 and κ_2. Let A_{κ_i} be the event that an adversary knows key κ_i, and suppose that after compromising s nodes an adversary knows x keys. Equation (4) assumes that

$$\mathsf{Pr}[A_{\kappa_1} \wedge A_{\kappa_2}] = \mathsf{Pr}[A_{\kappa_1}]\mathsf{Pr}[A_{\kappa_2}] = \left(\frac{x}{n}\right)^2 \quad .$$

However, this is not true because the events are not independent. Consider the conditional probability $\mathsf{Pr}[A_{\kappa_2}|A_{\kappa_1}]$. If the adversary already knows key κ_1, then it is slightly less likely that the adversary also knows κ_2, indeed, $\mathsf{Pr}[A_{\kappa_2}] = \frac{x-1}{n}$. Thus, the calculation leads to an overestimation of the true value of fail_s, as we demonstrate in Section 5.

3.2. **Yum and Lee.** We note that [3] has the same scope as our paper. However, the formula in [3] is difficult to compute, as we now demonstrate.

In [3, Theorem 2], Yum and Lee propose that fail_s for the q-composite scheme is given by

$$\sum_{\tau=k}^{\min\{ks,n\}} \left[\binom{n}{\tau} \left(\sum_{j=q}^{k} \frac{\binom{\tau}{j}}{\binom{n}{j}} \frac{p(j)}{\mathsf{Pr}_1} \right) \left(\frac{\binom{\tau}{k}^s - \sum_{\lambda=1}^{\tau-k} (-1)^{\lambda+1} \binom{\tau}{\lambda} \binom{\tau-\lambda}{k}^s}{\binom{n}{k}^s} \right) \right] \quad . \tag{5}$$

This formula is complicated and computationally laborious to evaluate; in addition we had difficulty in following the proof. We present a direct proof of a computationally simpler formula in Corollary 2 below. We also note that, whilst we are able to compute (5) for small values of n such as $n = 17$, our results are different from those given in [3, Table 1]. We are unable to reproduce any of their sample values, either by interpreting the 'p' in (5) to mean Pr_1 or $p_{connect}$. We conclude that there must be a typographical error somewhere in the formula and/or the proof.

4. Generalised resilience for random key predistribution schemes

In order to generalise across many instantiations of random key predistribution schemes, we have introduced a parameter $d \leq k$, which acts as an upper bound on the number of shared keys which nodes can use. This allows us to derive a formula which describes the resilience of many different random KPSs, including the schemes described in Section 2. We show in Corollary 1 that our formula is equivalent to that of Scheme 1 in the special case when $q = d = 1$.

We now present our generalised formula for fail_s, which applies to any key predistribution scheme where:

(1) each node is allocated k keys, selected independently and uniformly at random without replacement from a pool of n keys;
(2) the intersection threshold is $q \geq 1$, that is, nodes may only establish a common key if they share at least q keys;
(3) the upper bound on the number of shared keys a pair of nodes may use is d, where $q \leq d \leq k$; if two nodes share more than d keys then they should pick d of their keys at random to compute their common key;

(4) the function (such as hash, XOR, etc.) for producing a single link key from multiple common keys is such that an adversary must know all of the c shared keys between a pair of nodes to break the link; if the adversary only knows at most $c-1$ of the keys then the link remains secure.

Theorem 2. *For any random key predistribution scheme which fulfils conditions (1)–(4) above, the resilience is given by*

$$\begin{aligned}\mathsf{fail}_s &= \frac{1}{\mathsf{Pr}_1}\left(\sum_{c=q}^{d}\left[1-\sum_{i=1}^{c}(-1)^{i-1}\binom{c}{i}\left(\frac{\binom{n-i}{k}}{\binom{n}{k}}\right)^s\right]p(c)\right)+\\ &\quad \frac{1}{\mathsf{Pr}_1}\left(\left[1-\sum_{i=1}^{d}(-1)^{i-1}\binom{d}{i}\left(\frac{\binom{n-i}{k}}{\binom{n}{k}}\right)^s\right]\sum_{c=d+1}^{k}p(c)\right)\quad . \qquad (6)\end{aligned}$$

Proof. Consider a randomly-chosen pair of uncompromised nodes which share c keys, where $q \le c \le d$. For ease of notation and without loss of generality, we label these keys $\{1,2,\dots,c\}$. The probability that all of these c keys are known to an adversary which has compromised s nodes is

$$\mathsf{Pr}[\{1,\dots,c\} \in X_1 \cup \cdots \cup X_s] = 1 - \mathsf{Pr}\left[\bigcup_{i=1}^{c}\{i \notin X_1 \cup \cdots \cup X_s\}\right] \quad .$$

Using inclusion-exclusion, we have

$$\begin{aligned}\mathsf{Pr}[\{1,\dots,c\} \in X_1 \cup \cdots \cup X_s] &= 1 - c\mathsf{Pr}[1 \notin X_1 \cup \cdots \cup X_s] + \\ &\quad \binom{c}{2}\mathsf{Pr}[1,2 \notin X_1 \cup \cdots \cup X_s] + \cdots \\ &\quad +(-1)^{i-1}\binom{c}{i}\mathsf{Pr}[1,\dots,i \notin X_1 \cup \cdots \cup X_s] + \cdots \\ &= 1-\sum_{i=1}^{c}(-1)^{i-1}\binom{c}{i}\left(\frac{\binom{n-i}{k}}{\binom{n}{k}}\right)^s \quad .\end{aligned}$$

The probability of a randomly-chosen connected pair of uncompromised nodes sharing exactly c keys ($q \le c \le d$) is $\frac{p(c)}{\mathsf{Pr}_1}$. Therefore, for $q \le c \le d$ we have

$$\mathsf{fail}_s = \frac{1}{\mathsf{Pr}_1}\left(\sum_{c=q}^{d}\left[1-\sum_{i=1}^{c}(-1)^{i-1}\binom{c}{i}\left(\frac{\binom{n-i}{k}}{\binom{n}{k}}\right)^s\right]p(c)\right) \quad .$$

For $d < c \le k$, the probability of two connected nodes sharing c keys is again $\frac{p(c)}{\mathsf{Pr}_1}$. However, only d of these keys will be used to secure the link, and the choice of these d is made a priori, uniformly at random, and so without loss of generality they can be labelled $1,2,\dots,d$. Therefore the probability of the adversary knowing all d keys is

$$\mathsf{Pr}[\{1,\dots,d\} \in X_1 \cup \cdots \cup X_s] = 1-\sum_{i=1}^{d}(-1)^{i-1}\binom{d}{i}\left(\frac{\binom{n-i}{k}}{\binom{n}{k}}\right)^s \quad ,$$

using the result above, and so for $d < c \le k$,

$$\mathsf{fail}_s = \frac{1}{\mathsf{Pr}_1}\left(\left[1-\sum_{i=1}^{d}(-1)^{i-1}\binom{d}{i}\left(\frac{\binom{n-i}{k}}{\binom{n}{k}}\right)^s\right]\sum_{c=d+1}^{k}p(c)\right) \quad .$$

Adding these two results gives the final formula for fail_s. □

We now demonstrate that our formula agrees with that given in Lemma 2, in the case where $q = d = 1$.

Corollary 1 (Eschenauer Gligor resilience revisited)**.** *Using Theorem 2, the resilience of a random KPS which fulfils conditions (1)–(4) and where* $q = d = 1$ *(such as the Eschenauer Gligor scheme* [5]*), is given by*

$$\mathsf{fail}_s = 1 - \left(1 - \frac{k}{n}\right)^s \quad .$$

Proof. Setting $q = d = 1$ in (6) gives

$$\begin{aligned}\mathsf{fail}_s &= \frac{1}{\mathsf{Pr}_1}\left(\left[1-(-1)^0\binom{1}{1}\left(\frac{\binom{n-1}{k}}{\binom{n}{k}}\right)^s\right]p(1)+\left[1-(-1)^0\binom{1}{1}\left(\frac{\binom{n-1}{k}}{\binom{n}{k}}\right)^s\right]\sum_{c=2}^{k}p(c)\right)\\ &= \frac{\sum_{c=1}^{k}p(c)}{\mathsf{Pr}_1}\left(1-\left(\frac{\binom{n-1}{k}}{\binom{n}{k}}\right)^s\right) \quad .\end{aligned}$$

Since Pr_1 is by definition the sum of the probabilities of having $1, 2, \ldots, k$ keys in common, the first fraction is equal to 1, and we have

$$\begin{aligned}\mathsf{fail}_s &= 1 - \left(\frac{(n-1)!}{(n-1-k)!k!}\Big/\frac{n!}{(n-k)!k!}\right)^s\\ &= 1 - \left(1 - \frac{k}{n}\right)^s \quad ,\end{aligned}$$

as required. □

It is now straightforward to derive the correct formula for the resilience of Scheme 2 from Theorem 2:

Corollary 2 (q-composite resilience)**.** *Using Theorem 2, the resilience of a random KPSs such as the* q*-composite scheme from* [1] *that fulfils conditions (1)–(4) and where* $q > 1$ *and* $d = k$*, is given by*

$$\mathsf{fail}_s = \frac{1}{\mathsf{Pr}_1}\left(\sum_{c=q}^{k}\left[1-\sum_{i=1}^{c}(-1)^{i-1}\binom{c}{i}\left(\frac{\binom{n-i}{k}}{\binom{n}{k}}\right)^s\right]p(c)\right) \quad . \tag{7}$$

Proof. Using Theorem 2, we observe that when $d = k$ the summation from $c = d + 1$ to $c = k$ vanishes, leaving the formula given above. □

5. Numerical examples

We now compare our corrected formula to the original expression for fail_s. In Tables 1 and 2 we contrast equations (4) and (7) for sample values within the q-composite scheme ($d = k$). We fix $n = 1000$ and $k = 100$, and in Table 1 we fix $q = 10$ and vary s from 1 to 20. In Table 2 we fix $s = 10$ and vary q from 1 to 20. Differences are given as a percentage difference, that is, the final column is given by $\frac{(4)-(7)}{(7)} \times 100$.

We find that (4) gives higher values for fail_s, that is, it underestimates the resilience. As the differences are small, (4) can be thought of as an upper bound on the correct value. We note that an asymptotic approximation can be derived by routine approximation of (7) (using the basic techniques of Poisson approximation to the Binomial distribution); for example, it is a simple exercise to show that

$$\mathsf{fail}_s \approx \sum_{c=q}^{k}\frac{c!}{\mathsf{Pr}_1}\binom{k}{c}^2 e^{-\frac{1}{n}\left(k^2-2kc+\frac{c(c+1)}{2}\right)}\left(\frac{1-e^{-\frac{sk}{n}}}{n}\right)^c$$

and numerically this presents as a lower bound. These approximations are weakest when Pr_1 is very small (when n is large in comparison to k), but such low connectivity is unlikely to be used in practice. For more appropriate values of Pr_1 for network connectivity, the approximations become more accurate.

It can be seen that the value of fail_s using either formula increases in s and decreases in q. Conversely, the percentage difference decreases in s and increases in q. However, as the largest values of the percentage

s	(4)	(7)	% difference
1	2.75×10^{-11}	1.79×10^{-11}	53.532634
2	1.85×10^{-8}	1.52×10^{-8}	21.345324
3	7.03×10^{-7}	6.25×10^{-7}	12.472689
4	8.27×10^{-6}	7.63×10^{-6}	6.037381
5	0.000051	0.000048	6.037381
6	0.000213	0.000204	4.544026
7	0.000669	0.000647	3.517733
8	0.001720	0.001674	2.776871
9	0.003794	0.003711	2.222941
10	0.007423	0.007292	1.797917
11	0.013196	0.013006	1.465379
12	0.021692	0.021434	1.201299
13	0.033414	0.033086	0.989159
14	0.048737	0.048342	0.817219
15	0.067871	0.067415	0.676889
16	0.090850	0.090342	0.561736
17	0.117530	0.116984	0.466844
18	0.147617	0.147046	0.388391
19	0.180696	0.180114	0.323365
20	0.216264	0.215683	0.269363

Table 1. Comparison of formulae when $n = 1000$, $k = 100$, $q = 10$, hence $\mathsf{Pr}_1 = 0.555019$

q	Pr_1	(4)	(7)	% difference
1	0.999985	0.027080	0.026874	0.765811
2	0.999802	0.026966	0.026760	0.769232
3	0.998681	0.026520	0.026314	0.782573
4	0.994211	0.025397	0.025191	0.816410
5	0.981134	0.023337	0.023133	0.881074
6	0.951193	0.020382	0.020183	0.983376
7	0.895315	0.016889	0.016701	1.126261
8	0.807913	0.013337	0.013164	1.310152
9	0.690967	0.010113	0.009960	1.534359
10	0.555019	0.007423	0.007292	1.797917
11	0.416034	0.005313	0.005204	2.099974
12	0.289839	0.003730	0.003641	2.439908
13	0.187255	0.002580	0.002510	2.817340
14	0.112090	0.001765	0.001710	3.232105
15	0.062167	0.001197	0.001154	3.684213
16	0.031964	0.000806	0.000774	4.173823
17	0.015250	0.000540	0.000516	4.701213
18	0.006759	0.000360	0.000342	5.266761
19	0.002786	0.000240	0.000226	5.870935
20	0.001070	0.000159	0.000149	6.514278

Table 2. Comparison of formulae when $n = 1000$, $k = 100$, $s = 10$

differences correspond to the smallest values of fail_s for both equations, the absolute error in (4) remains small.

We therefore conclude that whilst our contribution is of mathematical importance it has limited impact on applications, as the original formula from [1] provides a close approximation to the true value.

Sample values for the formula from [3] are not given in the tables; for exact computation using Maple 15, the calculation did not terminate within an hour for input numbers of the magnitude given in the tables. By contrast, our formula can be evaluated within seconds on the same computer (AMD Phenom$^{\mathrm{TM}}$ II X4 970 CPU, 3.5 GHz, 16 GB RAM). Approximate calculation of (5) revealed answers appearing to converge to the results given by our formula, (7).

6. Conclusion

We have described two random key predistribution schemes and explained how the resilience of the q-composite scheme has been inaccurately presented in the literature. We have proposed a formula for fail_s which is rigorously proven, practical to compute, and applicable to a wide range of random key predistribution schemes because of the parameter d. Notice that if we take a scheme with $d = k$ and change d to be in the range $q \leq d < k$, then connectivity remains unchanged but resilience is reduced. Whilst this may be undesirable for many applications, setting $d < k$ may provide practical benefits such as reduced computation time, and an obstacle to an adversary in determining exactly which of the common keys have been hashed to create the key for a given link.

Correctly calculating resilience is important for accurately assessing and comparing KPSs; in particular, comparisons are often drawn between the performances of random and deterministic key predistribution schemes. It is therefore reassuring to know that the original equation in [1] produces probabilities which are similar to those given by the correct formula for fail_s. However, establishing the correct formula is of mathematical importance, and expressing it in a way which is computable is of practical importance.

References

[1] H. Chan, A. Perrig, and D. Song, “Random Key Predistribution Schemes for Sensor Networks,” in *SP ’03: Proceedings of the 2003 IEEE Symposium on Security and Privacy.* Washington, DC, USA: IEEE Computer Society, 2003, p. 197.

[2] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, “A pairwise key predistribution scheme for wireless sensor networks,” *ACM Trans.Inf.Syst.Secur.*, vol. 8, no. 2, pp. 228–258, 2005.

[3] D. H. Yum and P. J. Lee, “Exact Formulae for Resilience in Random Key Predistribution Schemes,” *IEEE Transactions on Wireless Communications*, pp. 1–5, 2012.

[4] M. B. Paterson and D. R. Stinson, “A unified approach to combinatorial key predistribution schemes for sensor networks,” *Cryptology ePrint Archive*, vol. Report 076, 2011. [Online]. Available: http://eprint.iacr.org/2011/076

[5] L. Eschenauer and V. D. Gligor, in *Proceedings of the 9th ACM conference on Computer and communications security*, New York, 2002.

[6] S. A. Camtepe and B. Yener, “Key Distribution Mechanisms for Wireless Sensor Networks: a Survey,” *Rensselaer Polytechnic Institute, Computer Science Department, Tech. Rep. TR-05-07*, 2005.

[7] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, “A survey of key management schemes in wireless sensor networks,” *Computer Communications*, vol. 30, no. 11-12, pp. 2314–2341, 2007.

[8] D. R. Stinson, *Cryptography: Theory and Practice.* Boca Raton, FL 33487-2742: Chapman & Hall/CRC, 2006.