Comments on a sensor network key redistribution technique of Cichon, Golebiewski and Kutylowski

Douglas R. Stinson*
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1, Canada

May 23, 2011

Abstract

Cichon, Golebiewski and Kutylowski ([2]) proposed a technique for "key redistribution" in sensor networks. The idea is that long-term keys held by the sensor nodes are used to encrypt temporal keys that a base station then broadcasts to the network. The temporal keys are used as session keys by the nodes in the sensor network. It is argued that this provides increased connectivity and resilience as compared to a standard Eschenauer-Gligor key predistribution scheme, as well as providing some additional advantages.

In this paper, we provide some simpler proofs of some results from [2]. As well, we give a precise analysis of the resilience of Cichon, Golebiewski and Kutylowski's scheme, and we discuss modifications of the scheme based on defining a suitable intersection threshold.

1 Introduction

Wireless sensor networks (*WSNs*) are comprised of small *sensor nodes* that have very limited storage, power and computational capabilities. The nodes in wireless sensor networks should be able to communicate with each other in order to accumulate information and relay it to a base station in a secure manner. Because this communication often takes place in a hostile environment, encryption and/or authentication should be used. This requires the establishment of secure keys between the sensor nodes in the WSN.

*Research supported by NSERC grant 203114-2011

A frequently used approach is to employ a *key predistribution scheme* (or *KPS*) that installs secret keys in each node before the sensor nodes are deployed. In the seminal paper by Eschenauer and Gligor [3], a probabilistic approach to key predistribution for sensor networks is proposed: every node is assigned a random k-subset of keys chosen from a given pool of secret keys. The other main approach used for key predistribution is deterministic, making use of combinatorial designs (see [4, 5] for a recent overview of these kinds of schemes). In this paper, we assume that keys are predistributed according to the Eschenauer-Gligor model.

Two nodes A_1 and A_2 form a *link* if they have at least one common key. Suppose that A_1 and A_1 have exactly $\ell \geq 1$ common keys, say $\mathsf{key}_{a_1}, \ldots, \mathsf{key}_{a_\ell}$, where $a_1 < a_2 < \cdots < a_\ell$. Then they can each compute the same pairwise secret key,

$$K = h(\mathsf{key}_{a_1} \parallel \ldots \parallel \mathsf{key}_{a_\ell} \parallel i \parallel j),$$

using an appropriate public *key derivation function* h, which has suitable input and output sizes. Such key derivation functions could be constructed from a secure public hash function, e.g., SHA-1.

Chan, Perrig and Song [1] proposed a modification of the Eschenauer-Gligor scheme where two nodes will compute a pairwise key only if they share at least η common keys, where the integer $\eta \geq 1$ is a pre-specified *intersection threshold*. Given that two nodes have at least η common keys, they use all their common keys to compute their pairwise key, by means of an appropriate key derivation function, as described above.

The most studied adversarial model in wireless sensor networks is *random node compromise* [3], where an adversary compromises a fixed number of randomly chosen nodes in the network and extracts the keys stored in them. A link formed by two nodes A_1 and A_2 will be *broken* if a node $B \neq A_1, A_2$ is compromised, where $A_1 \cap A_2 \subseteq B$. More generally, if nodes $B_1, \ldots, B_s$ are compromised, then a link formed by two other nodes A_1 and A_2 will be broken whenever

$$A_1 \cap A_2 \subseteq \bigcup_{i=1}^{s} B_i.$$

1.1 Metrics

There are various metrics that quantify different performance and security aspects of a key predistribution scheme for a wireless sensor network. We summarise three of these metrics now.

Storage requirements

The number of keys stored in each node is equal to k. We want k to be "small" (e.g., $k \leq 100$), due the the limited storage in sensor nodes.

Network connectivity

It is common to measure local connectivity of a network by computing the probability, denoted by Pr_1, that a randomly chosen pair of nodes is a link. We want Pr_1 to be "large" (e.g., $\mathsf{Pr}_1 \geq 0.6$).

Network resilience

Resilience against node capture is commonly measured by computing the probability that a random link is broken by the compromise of a set of s random nodes not in the link, for suitable values of s. For simplicity, we only consider the case $s = 1$ in this paper, and we denote the probability of interest by fail. We call this the *failure probability* of the scheme. We desire a "small" value of fail (e.g., $\mathsf{fail} < 0.01$), which means that resilience of the network is high.

It is trivial to optimise any two of three metrics; see, for example, [5]. A more interesting and challenging problem is, for a given (relatively small) value of k, to constuct a scheme that has a high value for Pr_1 and a low value for fail.

1.2 Key redistribution

In a recent paper by Cichon, Golebiewski and Kutylowski ([2]), an interesting technique for "key redistribution" in sensor networks is described. As described earlier, there is a *key pool* $\mathcal{K} = \{\mathsf{key}_1, \ldots, \mathsf{key}_n\}$ consisting of n random keys. These are termed *long-term keys*. Each sensor node holds a random k-subset of $\mathcal{K}$ in its key ring; this is the basic method introduced by Eschenauer and Gligor [3]. However, in [2], the n long-term keys are used to encrypt n/m *temporal keys*, where m is a small constant. Each temporal key is encrypted using m long-term keys; the choice of which long-term key is used to encrypt which temporal key is made randomly. The base station then broadcasts the temporal keys to the network. The temporal keys are used as session keys by the nodes in the sensor network.

The paper [2] argues that their scheme provides increased connectivity and decreased resilience as compared to the EG-scheme, as well as providing some additional advantages. One attractive feature of this scheme is that a broken link can be "restored" by a subsequent broadcast of temporal keys. However, it should be noted that the requirement of a base station that can

broadcast online messages to all sensor nodes is a rather strong assumption that might not be realistic in all application scenarios for WSNs.

Our main contributions are as follows. In Section 2, we provide some simpler proofs of some results from [2]. In Section 3, we give a precise analysis of the resilience of the scheme in [2]. Then, in Section 4, we briefly discuss a modification of the scheme based on defining a suitable intersection threshold.

2 Expected number of shared keys

In this section, we give some simplified proofs of results from [2]. First, we establish the combinatorial framework that we will use in the rest of our paper.

We have noted that each node contains a k-subset of keys from $\mathcal{K}$. The indices of these keys forms a k-subset of $X = \{1, \ldots, n\}$ that we term a *block*. Thus, each node can be identified with the block that is associated with the keys that the node holds; henceforth we will use the terms "node" and "block" interchangably. Note that every block is a k-subset of $\{1, \ldots, n\}$ that is chosen independently and uniformly at random from the set of all $\binom{n}{k}$ possible k-subsets.

In [2, Theorem 1], formulas are proven for the expected number of shared long-term keys and the expected number of shared temporal keys for two nodes. The proofs given in [2] use some heavy machinery involving generating functions. However, this theorem has a quick, simple proof based on the linearity of expectation of random variables.

First we look at [2, Theorem 1 (part 2)], which asserts that the expected number of temporal keys shared by two nodes is $\frac{n}{m}\left(1 - \frac{\binom{n-m}{k}}{\binom{n}{k}}\right)^2$. Suppose that $G_1, \ldots, G_{n/m}$ partition the n-set $\{1, \ldots, n\}$ into n/m disjoint m-sets. A and B are random blocks. The number of temporal keys shared by A and B is

$$t_{A,B} = |\{i : A \cap G_i \neq \emptyset \text{ and } B \cap G_i \neq \emptyset\}|.$$

For $1 \leq i \leq n/m$, define a random variable $\tilde{\mathsf{X}}_i = 1$ if $A \cap G_i \neq \emptyset$ and $B \cap G_i \neq \emptyset$, and define $\tilde{\mathsf{X}}_i = 0$, otherwise. Let $\tilde{\mathsf{X}} = \sum_{i=1}^{n/m} \tilde{\mathsf{X}}_i$. Then $\tilde{\mathsf{X}}$ computes $t_{A,B}$ and $E[\tilde{\mathsf{X}}]$ is the expected value of $t_{A,B}$. It is obvious that

$$\mathsf{Pr}[A \cap G_i \neq \emptyset] = \mathsf{Pr}[B \cap G_i \neq \emptyset] = 1 - \frac{\binom{n-m}{k}}{\binom{n}{k}}$$

and hence

$$E[\tilde{\mathsf{X}}_i] = \mathsf{Pr}[A \cap G_i \neq \emptyset \text{ and } B \cap G_i \neq \emptyset] = \left(1 - \frac{\binom{n-m}{k}}{\binom{n}{k}}\right)^2.$$

By linearity of expectation,

$$E[\tilde{\mathsf{X}}] = \frac{n}{m}\left(1 - \frac{\binom{n-m}{k}}{\binom{n}{k}}\right)^2, \tag{1}$$

which proves [2, Theorem 1 (part 2)].

To prove [2, Theorem 1 (part 1)], we just set $m = 1$ in the formula derived above. We have

$$\frac{n}{1}\left(1 - \frac{\binom{n-1}{k}}{\binom{n}{k}}\right)^2 = n\left(1 - \frac{n-k}{n}\right)^2 = \frac{k^2}{n},$$

which proves the desired result.

2.1 Estimates

In the previous section, we reproved the exact formula (1) for the expected number of shared temporal keys. In [2, Corollary 1], an estimate for (1) is given when k is roughly $\sqrt{n}$. Here we provide a simple and quite accurate estimate that holds for a range of values of k.

First, we estimate

$$\frac{\binom{n-m}{k}}{\binom{n}{k}} \approx \frac{(n-m)^k}{n^k} = \left(1 - \frac{m}{n}\right)^k,$$

so

$$E[\tilde{\mathsf{X}}] \approx \frac{n}{m}\left(1 - \left(1 - \frac{m}{n}\right)^k\right)^2.$$

Next,

$$\left(1 - \frac{m}{n}\right)^k \approx 1 - \frac{km}{n} + \frac{k^2m^2}{2n^2},$$

so

$$E[\tilde{\mathsf{X}}] \approx \frac{n}{m}\left(\frac{km}{n} - \frac{k^2m^2}{2n^2}\right)^2 = \frac{k^2m}{n}\left(1 - \frac{km}{2n}\right)^2.$$

Finally, if we expand the square and ignore the last term, we get

$$E[\tilde{\mathsf{X}}] \approx \frac{k^2m}{n}\left(1 - \frac{km}{n}\right). \tag{2}$$

If $k = \sqrt{n}$, then our estimate (2) is

$$\frac{k^2 m}{n} - \frac{k^3 m^2}{n^2} = \frac{k^2 m}{n} - \frac{m^2}{\sqrt{n}}.$$

The estimate given in [2] is

$$\frac{k^2 m}{n} + O\left(\frac{1}{\sqrt{n}}\right).$$

However, in [2], m is assumed to be fixed and the big-oh hides an unspecified constant that depends on m.

Our estimate is quite accurate for reasonable values of the parameters. For example, when $n = 10000$, $k = 100$ and $m = 16$, the exact value of $E[\check{\mathsf{X}}] = 13.81$ while the estimate (2) is 13.44. Here is another data point: when $n = 10000$, $k = 50$ and $m = 16$, the exact value of $E[\tilde{\mathsf{X}}] = 3.718$ while the estimate (2) is 3.680.

3 Resilience

Resilience of a scheme measures the probability that a link between two nodes will be broken by the compromise of a third node. Here we are interested in resilience with respect to temporal keys. (Recall that two nodes A and B form a link if they contain at least one common temporal key, and this link is broken by the compromise of a node C if C holds all the temporal keys that are held by both A and B.) [2] studies the resilience of their key redistribution scheme, but they make several simplifying assumptions. Here we give a much more general analysis and we derive general formulas for resilience.

In [2, Theorem 2], it is assumed that two nodes A and B have exactly m temporal keys in common. In view of the estimates provided in the last section, this is roughly the expected number of common temporal keys when $k = \sqrt{n}$. Under this assumption, [2, Theorem 2] estimates the probability that a random node C contains these m common temporal keys to be $(km/n)^m$.

3.1 Number of temporal keys

As before, suppose that $G_1, \ldots, G_{n/m}$ partition an n-set $X = \{1, \ldots, n\}$ into $v = n/m$ disjoint m-sets. Suppose A is a random block (i.e., a k-subset of X) and define

$$\mathcal{I}(A) = \{i : A \cap G_i \neq \emptyset\}.$$

$\mathcal{I}(A)$ is the set of indices of the temporal keys held by A. Then let

$$k_A = |\mathcal{I}(A)|;$$

k_A is the number of temporal keys held by A.

Fix any i-subset $I \subseteq \{1, \ldots, v\}$. Define

$$M(i) = |\{A : \mathcal{I}(A) = I\}|.$$

Note that $M(i)$ counts the number of possible nodes whose set of temporal keys is equal to I. The value $M(i)$ does not depend on the particular i-subset I that was chosen.

It is easy to see that

$$|\{A : \mathcal{I}(A) \subseteq I\}| = \binom{im}{k}. \tag{3}$$

We can derive a formula for $M(i)$ from (3) by applying the principle of inclusion-exclusion.

Lemma 3.1. *For $i \geq 1$, we have*

$$M(i) = \sum_{j=0}^{i-1} (-1)^j \binom{(i-j)m}{k} \binom{i}{j}. \tag{4}$$

Next, define

$$N(i) = |\{A : k_A = i\}|.$$

$N(i)$ is the number of possible nodes holding exactly i temporal keys. The following is an immediate consequence of (4).

Lemma 3.2. *For $i \geq 1$, we have*

$$N(i) = \binom{v}{i} M(i) = \sum_{j=0}^{i-1} (-1)^j \binom{v}{i} \binom{(i-j)m}{k} \binom{i}{j}. \tag{5}$$

3.2 Intersection of two blocks

Next, we consider intersections of two blocks. For $t \geq 1$, define a *t-link* to be an ordered pair of two nodes that contain exactly t common temporal keys. Let $P(t)$ denote the number of possible t-links; then

$$P(t) = |\{(A, B) : |\mathcal{I}(A) \cap \mathcal{I}(B)| = t\}|.$$

We have the following formula for $P(t)$:

Lemma 3.3. *For $t \geq 1$, we have*

$$P(t) = \sum_{i=t}^{k} \sum_{j=t}^{k} \binom{v-i}{j-t} \binom{i}{t} N(i)M(j). \tag{6}$$

For $t = 0$, we have

$$P(0) = \sum_{i=1}^{k} \sum_{j=1}^{k} \binom{v-i}{j-t} N(i)M(j). \tag{7}$$

Proof. Denote $i = k_A$ and $j = k_B$. We can choose A in $N(i)$ ways. For each choice of A, choose t indices in $\mathcal{I}(A)$ and choose $j - t$ indices in $\{1, \ldots, v\} \backslash \mathcal{I}(A)$. Let the set of the j chosen indices be denoted by J. Then choose B such that $\mathcal{I}(B) = J$; there are $M(j)$ ways to do this. □

Remark 3.1. *We can "numerically" verify the formulas (6) and (7) by checking that the following equations hold for various values of n, m and k:*

$$\sum_{t=0}^{k} P(t) = \binom{n}{k}^2$$

and

$$\frac{\sum_{t=1}^{k} tP(t)}{\binom{n}{k}^2} = \frac{n}{m} \left(1 - \frac{\binom{n-m}{k}}{\binom{n}{k}} \right)^2.$$

3.3 Compromised links and resilience

Suppose that (A, B) is a t-link. Then define

$$S(t) = |\{C : \mathcal{I}(A) \cap \mathcal{I}(B) \subseteq \mathcal{I}(C)\}|.$$

$S(t)$ denotes the number of possible nodes that will compromise the t-link (A, B), and it does not depend on the particular choices of A and B.

Lemma 3.4. *For any $t > 0$, we have*

$$S(t) = \sum_{i=t}^{k} \binom{v-t}{i-t} M(i). \tag{8}$$

Proof. Let $i = k_C$. Choose $i - t$ indices in

$$\{1, \ldots, v\} \backslash (\mathcal{I}(A) \cap \mathcal{I}(B)).$$

Let J denote the i-set consisting of the $i - t$ chosen indices along with $\mathcal{I}(A) \cap \mathcal{I}(B)$. Then choose C such that $\mathcal{I}(C) = J$; there are $M(i)$ ways to do this. □

Finally, define

$$T(t) = |\{(A, B, C) : |\mathcal{I}(A) \cap \mathcal{I}(B)| = t \text{ and } \mathcal{I}(A) \cap \mathcal{I}(B) \subseteq \mathcal{I}(C)\}|.$$

$T(t)$ counts triples (A, B, C) where (A, B) is a t-link compromised by C. It is clear, applying (8), that the following formula holds.

Lemma 3.5. *For any $t > 0$, we have*

$$T(t) = P(t)S(t) = \sum_{i=t}^{k} \binom{v-t}{i-t} M(i)P(t).$$

Now we are in a position to compute some resilience parameters. Recall that the failure probability fail denotes the probability that a random link (A, B) is compromised by a random node C.

Theorem 3.6. *The failure probability is given by*

$$\mathsf{fail} = \frac{\sum_{t=1}^{k} T(t)}{\sum_{t=1}^{k} P(t)\binom{n}{k}}. \tag{9}$$

Proof. The total number of possible t-links with $t \geq 1$ is

$$\sum_{t=1}^{k} P(t),$$

so the total number of triples (A, B, C) where (A, B) is a link is

$$\sum_{t=1}^{k} P(t)\binom{n}{k}.$$

The total number of triples (A, B, C) where (A, B) is a link and C compromises this link is

$$\sum_{t=1}^{k} T(t).$$

The resilience is just the quotient of these two quantities. □

Define fail_t to denote the the probability that random t-link (A, B) is compromised by a random node C. We have the following obvious result.

Lemma 3.7. *For any $t \geq 1$, we have*

$$\mathsf{fail}_t = \frac{S(t)}{\binom{n}{k}}. \tag{10}$$

Lemma 3.7 provides another way to derive the formula (9) for fail. Let λ_t denote the probability that a random link is a t-link. It is clear that

$$\lambda_t = \frac{P(t)}{\sum_{i=1}^{k} P(i)} \tag{11}$$

and

$$\mathsf{fail} = \sum_{t=1}^{k} \lambda_t \mathsf{fail}_t. \tag{12}$$

Then, from (10), (11) and (12), we have

$$\begin{aligned} \mathsf{fail} &= \sum_{t=1}^{k} \lambda_t \mathsf{fail}_t \\ &= \sum_{t=1}^{k} \frac{P(t)S(t)}{\sum_{i=1}^{k} P(i)\binom{n}{k}} \\ &= \frac{\sum_{t=1}^{k} T(t)}{\sum_{t=1}^{k} P(t)\binom{n}{k}}, \end{aligned}$$

agreeing with (9).

3.4 Numerical examples

First, we give an example to illustrate the computation of resilience parameters.

Example 3.1. *Suppose $n = 1000$, $m = 4$ and $k = 31$. Then the expected number of temporal keys held by a node, given by (1), is* 3.511857771*, which is a bit less than* 4*. [2, Theorem 2] estimates fail_4 by computing the quantity*

$$\left(\frac{km}{n}\right)^m = 0.0002364213760.$$

A more accurate estimate for fail_4*, based on the analysis in [2], would be*

$$\frac{\binom{v-m}{k-m}}{\binom{v}{k}} = 0.0001980391200.$$

However, from (10), the exact value $\mathsf{fail}_4 = 0.0001651542962$.

In contrast, the overall resilience of the scheme is determined from (9); here, we get

$$\mathsf{fail} = 0.01330121549.$$

This is quite a bit higher than fail_4*, primarily because links consisting of fewer than four temporal keys (which occur frequently) are compromised with higher probability. This can be seen in the following tabulation of values* λ_t *and* fail_t*:*

t	λ_t	fail_t
1	0.08756777557	0.1185218591
2	0.1843995070	0.01364696407
3	0.2407996311	0.001524883082
4	0.2188569817	0.0001651542962
5	0.1472998707	0.00001731603382
6	0.07626527018	0.000001755184555

Our next example considers the effect of varying the parameter k.

Example 3.2. *Suppose* $n = 1000$ *and* $m = 4$*. We compute the values of* fail *for various choices of* k*:*

k	fail
5	0.01925413575
10	0.03349126556
15	0.03904935504
20	0.03548705708
25	0.02588255435
30	0.01518790238
35	0.007187785428
40	0.002776219702
45	0.0008938567010
50	0.0002464139425

It is interesting to observe that fail *at first increases, and then decreases, as* k *increases. The higher values of* fail *for small values of* k *reflect the fact that the network has fewer links and the links that do exist are more easily compromised.*

Our next example considers the effect of varying the parameter m.

Example 3.3. *Suppose* $n = 5040$ *and* $k = 50$. *We compute the values of* fail *for various choices of* m*:*

m	fail
2	0.01182106347
3	0.01334509061
4	0.01321072373
5	0.01211454057
6	0.01055743581
7	0.008871668296
8	0.007256884957
9	0.005816967246
10	0.004592239563

The interesting thing to note here is that fail *decreases as* m *increases beyond* 3*, but the decrease is gradual and not very dramatic.*

4 Intersection thresholds

We discussed the idea of an intersection threshold in Section 1. Basically, as η increases, resilience increases and connectivity decreases. We now develop formulas for these metrics, that depend on the intersection threshold of the scheme.

The connectivity of a scheme is measured by computing the probability Pr_1 that a random pair of nodes is a link. The following result gives a formula for Pr_1.

Theorem 4.1. *For a scheme with intersection threshold* η*, we have that*

$$\mathsf{Pr}_1 = 1 - \frac{\sum_{i=0}^{\eta-1} P(i)}{\binom{n}{k}^2}. \tag{13}$$

Proof. There are $\binom{n}{k}^2$ possible pairs of nodes, of which $\sum_{i=0}^{\eta-1} P(i)$ do not form links. □

The formula (9) for resilience is generalised as follows.

Theorem 4.2. *For a scheme with intersection threshold* η*, the failure probability is given by*

$$\mathsf{fail} = \frac{\sum_{t=\eta}^{k} T(t)}{\sum_{t=\eta}^{k} P(t)\binom{n}{k}}. \tag{14}$$

Proof. The proof is a straightforward modification of the proof of Theorem 3.6. □

We now revisit Example 3.1.

Example 4.1. *Suppose $n = 1000$, $m = 4$ and $k = 31$, as in Example 3.1. We compute the connectivity and failure probabilities for various values of η.*

η	Pr_1	fail
1	0.9809852766	0.01330121549
2	0.8950825780	0.003202999469
3	0.7141893766	0.0005577036219
4	0.4779684839	0.00007970558807
5	0.2632730072	0.00001002335465

The use of an intersection threshold allows a suitable tradeoff between connectivity and resilience. Observe that resilience increases quickly as η is decreased; however, connectivity decreases at the same time. For $\eta > 5$, the connectivity is too low to be practical. In this example, $\eta = 2$ or 3 provides a good way to "balance" connectivity and resilience.

References

[1] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proceedings of the 2003 Symposium on Security and Privacy*. IEEE Computer Society, 197–213.

[2] J. Cichon, Z. Golebiewski and M. Kutylowski. From key predistribution to key redistribution. *Lecture Notes in Computer Science* **6451** (2010), 92–104.

[3] L. Eschenauer and V. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*. ACM Press, 2002, pp. 41–47.

[4] K. M. Martin. On the applicability of combinatorial designs to key predistribution for wireless sensor networks. *Lecture Notes in Computer Science* **5557** (2009), 124–145.

[5] M. B. Paterson and D. R. Stinson. A unified approach to combinatorial key predistribution schemes for sensor networks. Cryptology ePrint Archive: Report 2011/076, `http://eprint.iacr.org/2011/076`.