An Efficient Random Key Pre-distribution Scheme

Mahalingam Ramkumar
Department of Computer Science and Engineering
Mississippi State University
Mississippi State, MS 39762

Nasir Memon
Department of Computer and Information Science
Polytechnic University
Brooklyn, NY 11201

***Abstract*— Any key pre-distribution (KPD) scheme is inherently a trade-off between complexity and security. By sacrificing some security (KPD schemes need some assurance of the ability to *limit sizes of attacker coalitions*), KPD schemes gain many advantages. We argue that random KPD schemes, in general, perform an "advantageous" trade-off which renders them more suitable for practical large scale deployments of resource constrained nodes. We introduce a novel random KPD scheme, HAshed Random Preloaded Subsets (HARPS), which turns out to be a generalization of two other random KPD schemes - random preloaded subsets (RPS), and a scheme proposed by Leighton and Micali (LM). All three schemes have *probabilistic* measures for the "merit" of the system. We analyze and compare the performance of the three schemes. We show that HARPS has significant advantages over other KPD schemes, and in particular, over RPS and LM.**

I. Introduction

In many evolving applications involving distributed, loosely controlled, resource constrained nodes, it is imperative to have efficient means of developing "trust" between nodes. For example, nodes forming Mobile Ad hoc NETworks (MANET) have to perform *authenticated* exchanges for building a routing table, or relaying messages between other nodes. In addition it is also necessary to ensure *privacy* of inter-nodal exchanges. The two basic requirements, authentication and privacy, could be provided by a key distribution scheme (KDS).

However, for applications involving autonomous resource constrained nodes, the KDS may have many constraints. For instance, the nodes need to operate *without active involvement of a trusted authority* (TA). The KDS should *scale well* (support large network sizes). Further, addition of new nodes should not require any kind of *reconfiguration of the system*. Additionally, the KDS should not introduce significant *overheads* - both in terms of *computational complexity* and *bandwidth*.

The need for ad hoc interaction of nodes, and the resource constraints make traditional KDS solutions like Kerberos and public key infrasructure (PKI) impractical. A third option is key pre-distribution. A key pre-distribution (KPD) scheme, consists of N nodes with *unique* IDs, and a TA. The TA chooses P secrets (or keys). Each node is preloaded with k secrets, such that any two nodes A and B (with unique IDs ID_A and ID_B respectively) can "discover" a shared secret K_{AB} *independently*, by just exchanging their IDs, and without subsequent interaction with the TA. The shared secret K_{AB} could be used for ensuring privacy of communications between nodes A and B. As the shared secret is generally a function of the node IDs, simultaneous (mutual) *authentication* of the node IDs is also achieved. A KPD may also enable arbitrary *groups* of nodes to arrive at a shared *group* secret (privy only to the members of the group).

For the most basic form of KPD, shared secrets for each possible group a node may be a member of, may be preloaded in each node. However, this would imply an unreasonably large amount of storage requirement in each node, especially if the network size (the total number of nodes in the system) N is large. Fortunately, it is possible for KPD schemes to perform *trade-offs between complexity and security*.

Key pre-distribution schemes can be classified broadly into *deterministic* and *random* KPD schemes. The contribution of this paper is a novel random KPD scheme, HARPS. In Section II we present formal introduction to random KPD schemes, followed by a description of HARPS. Section III presents an analysis of the performance of HARPS, and a comparison of HARPS with two other random KPD schemes (RPS [1] and LM [2], which turn out to be special cases of HARPS). It is shown that HARPS performs significantly better than RPS and LM.

While any KPD scheme is inherently a trade-off between security and complexity, the specific *nature of the trade-off* employed results in KPD schemes assuming drastically different forms. In Section IV various deterministic and random KPD schemes are compared based on the security-complexity trade-offs they employ. The advantageous trade-offs performed by random KPD schemes are elucidated. Conclusions are presented in Section V.

II. Random Key Pre-distribution

A (r, n, p) (or r-conference, n-secure with probability $1-p$) KPD scheme [1], is a systematic method for generation of symmetric keys or secrets. In an (r, n, p) scheme, k secrets are preloaded in each node in such a way that

1) Any set of (up to) r nodes $\{u_1, \ldots, u_r\} = \mathcal{G}$, can discover a group secret $K_{\mathcal{G}}$ independently, with probability p_d.
2) The probability that the group secret *can also be discovered* by a coalition of n nodes $a_1 \cdots a_n \notin \mathcal{G}$ is p (or less).

The set of k keys in each node is collectively referred to as the node's "key-ring." The parameter k is then the size

of the key-ring. The probability $p_o = 1 - p_d$ (which is the probability that r nodes *cannot* discover a shared secret), is referred to as the *outage probability*. Pairwise, or unicast communications, can be considered as a special case of r-conferences where $r = 2$. For coalitions of *more* than n nodes, say $a_1, \ldots, a_{n+l} \notin \mathcal{G}, l > 0$, the probability of obtaining the group secret is typically *greater* than p. We refer to the probability p as the *eavesdropping probability*.

For a KPD scheme to be "secure" even when an attacker has exposed secrets from n nodes, the outage probability p_o and the eavesdropping probability p should be "very close" to 0. Obviously, if the group of r nodes *cannot* discover a shared secret (which happens with a probability p_o) an attacker can compromise exchanges between such nodes *even if he has not compromised any other node* (or $n = 0$). In other words, $p \geq p_o$ *includes* the outage probability[1]. More specifically, $p = p_o$ for $n = 0$, and $p \geq p_o$ for $n > 0$.

A. HARPS

HARPS is a random KPD scheme defined by 3 parameters[2] (P, k, L), and two public functions $h()$ and $F()$. In other words,

1) P - number of secrets stored by the TA
2) k - number of preloaded secrets in each node, $k \leq P$.
3) L - Maximum hash depth
4) $h()$, a cryptographic hash (one-way) function, and
5) $F()$, a public-key-generation function.

Each node is assigned a unique ID. The TA chooses P secrets (or *root* keys) $[M_1 \cdots M_P]$. From each root key one can get up to L "derived keys" by repeated application of the one-way function $h()$. The j^{th} derived key of the i^{th} root key M_i is represented by $K_i^j = h^j(M_i)$ (M_i is repeatedly hashed j times - or j is the hash "depth" of K_i^j), where $1 \leq i \leq P$, $1 \leq j \leq L$. The parameter L is the *maximum hash depth*.

For a node A with ID ID_A

$$\{(A_1, a_1), (A_2, a_2), \ldots (A_k, a_k)\} = F(ID_A). \tag{1}$$

The first coordinate of the ordered pairs, viz. $(A_1 \cdots A_k)$, is a *partial random permutation*[3] of integers between 1 and P, $1 \leq A_j \leq P \, \forall 1 \leq j \leq k$ and $A_j \neq A_i, i \neq j$. The second coordinate $\{a_1 \cdots a_k\}$ is a sequence of *uniformly distributed* numbers between 1 and L. Each node is preloaded with k secrets. The preloaded secrets in any node depends on the node's public key. The keys preloaded in node A for instance, are

$$[K_{A_1}^{a_1} \cdots K_{A_k}^{a_k}] = [h^{a_1}(M_{A_1}) \cdots h^{a_k}(M_{A_k})]. \tag{2}$$

In other words, the first coordinate of $F(ID)$ of a node, represents the indexes of the root keys that are chosen to be preloaded in the node. The second coordinate represents the number of times each chosen key is hashed before they are actually preloaded in the node.

1) Calculating Group Secrets: Let $[(B_1, b_1) \cdots (B_k, b_k)] = F(ID_B)$. With the knowledge of the node IDs, the two nodes A and B can independently arrive at the *indexes* of the shared *root* keys as $[s_1 \cdots s_m]$ by application of the (public) function $F()$ (on their IDs). Corresponding to these shared root keys, denote the m derived keys in node A as $[K_{s_1}^{a_1} \cdots K_{s_m}^{a_m}]$, and in node B, $[K_{s_1}^{b_1} \cdots K_{s_m}^{b_m}]$. Let $d_1 = \max(a_1, b_1) \cdots d_m = \max(a_m, b_m)$. The session key (or the group secret) K_{AB} is then obtained as

$$K_{AB} = h(h^{d_1}(M_{s_1}) | h^{d_2}(M_{s_2}) | \cdots | h^{d_m}(M_{s_m})) \tag{3}$$

Note that the group secret can calculated independently by both nodes. Without loss of generality, let us assume that $d_1 = \max(a_1, b_1) = b_1$. This implies that node A would arrive at the term $h^{d_1}(M_{s_1}) = K_{s_1}^{d_1}$ in Eq (3) by hashing its key $K_{s_1}^{a_1}$, $(b_1 - a_1)$ times. As one of B's preloaded key is $K_{s_1}^{b_1} = K_{s_1}^{d_1} = h^{d_1}(M_{s_1})$, B does not have to hash forward for this particular key. For each shared key however, at most one of the nodes has to hash forward to reach a common hash depth. Thus each shared key is first brought to same depth by both nodes. All such keys, hashed together, serve as the shared key between the nodes A and B as shown in Eq (3).

HARPS is actually a generalization of both LM [2] and RPS [1]. RPS is a special case of HARPS for $L = 0$ - the keys are *not* hashed before they are preloaded in the nodes. LM is a special case of HARPS for $P = k$ - each node has a hashed version of *all* root keys.

Network Size: For most KPD schemes, there is practically *no limit on the network size*. For HARPS, the number of possible unique key-rings is $\binom{P}{k} L^k$. As the keys (and hence key-rings) are chosen randomly, in order to avoid "collision" (taking "birthday paradox" into account) we could restrict the size of the network, N_{max} to $\sqrt{\binom{P}{k} L^k}$. For example, for $P = 2560$, $k = 256$ and $L = 64$, $N_{max} \approx 1.3 \times 10^{411}$. In practice the limitation on the size of the network would be based on the *number of bits allocated* for representing the (unique) ID of each node. For a system with 32 bit IDs for instance, the network size would be limited to about 4 billion.

III. Analysis of HARPS

An analysis of HARPS involves calculation of the eavesdropping probability p (more specifically $p(n, r)$), that an attacker who *controls a coalition*[4] of n nodes, can discover the shared secret of a group of r nodes.

To analyze[5]

[1]Which is the reason why p_o (or p_d) is not explicitly included in the notion of an (r, n, p) KPD

[2]These parameters **do not** represent the parameters (r, n, p) used to describe a general KPD!

[3]the sequence $(A_1 \cdots A_k)$, for instance, can be considered as the first k numbers in a random permutation of numbers $1 \cdots P$.

[4]Typically, due to the need to limit attacker coalition sizes, KPD schemes would involve some form of tamper-resistance of the nodes preloaded with keys. Thus "ownership" of a node alone is not a sufficient condition for a node to considered as a part of an "attacker's coalition." In order to be part of the coalition, the attacker must have been able to tamper with and *expose* all secrets buried in a tamper-resistant node.

[5]The analysis presented in this section is motivated in part by Dyer et al. [9], for calculating the probability that an intersection of r sets is contained in an union of n sets.

Consider an arbitrary key (say index i) from the pool of P keys. Let ξ_{ij} represent the probability that the ith key $1 \leq i \leq P$, is chosen as a candidate for node j, $1 \leq j \leq N$. As each key is chosen with the same probability, and as any node is equally likely choose any key as a candidate, we have

$$\xi_{ij} = \xi = \frac{k}{P} \forall i, j. \tag{4}$$

In order for the key i to be "safe", the following conditions should be met

1) The r nodes trying to establish a shared secret, should have chosen the i the key.
2) None of the n attacker's nodes should have chosen the ith key, or if some of the attackers have the ith key, all such keys should have hash depths higher than the maximum hash depth of the r nodes.

Now the probability that the ith key is safe, is

$$\varepsilon = \sum_{u=0}^{n} \mathcal{P}(\xi, r, n, u) \mathcal{Q}(L, r, u) \tag{5}$$

where $\mathcal{P}(\xi, r, n, u)$, is the probability that r nodes *and* $0 \leq u \leq n$ out of n compromised nodes in the attacker's coalition pick the ith key from the key pool, or

$$\mathcal{P}(\xi, r, n, u) = \xi^r \cdot \binom{n}{u} \xi^u (1-\xi)^{n-u}, \tag{6}$$

and, $\mathcal{Q}(L, r, u)$ is the probability that the minimum of u realizations of a uniform distribution is greater than the maximum of r realizations. In other words,

$$\mathcal{Q}(L, r, u) = \Pr\{\beta > \alpha\} = \sum_{i=1}^{L} \frac{i^r - (i-1)^r}{L^r} \left(\frac{L-i}{L}\right)^u, \tag{7}$$

where $1 \leq u \leq n$. Further, $\mathcal{Q}(L, r, u = 0) = 1$ and $\mathcal{Q}(L = 0, r, u) = 0$.

Therefore the probability that a coalition of n attackers can successfully arrive at the r-group secret is

$$p(n, r) = (1-\varepsilon)^P = \left\{(1-\varepsilon)^{\frac{1}{\xi}}\right\}^k. \tag{8}$$

For the special cases of RPS ($L = 0$) and LM ($\xi = 1$), the expressions for ε are respectively

$$\varepsilon = \begin{cases} \varepsilon_R = \mathcal{P}(\xi, r, n, 0) = \xi^r (1-\xi)^n & L = 0, \xi \neq 1 \\ \varepsilon_L = \mathcal{Q}(L, r, n) & \xi = 1, L \neq 0 \end{cases}. \tag{9}$$

Note that the expression for ε in Eq (5) for HARPS has $n+1$ terms corresponding to $u = 0 \cdots n$. For LM ($\xi = 1$), only the term corresponding to $u = n$ is non zero. For RPS, only the term corresponding to $u = 0$ is non-zero. For the same value of ξ obviously $\varepsilon_R \leq \varepsilon$. Or for the same P, k, the eavesdropping probability for HARPS ($L \geq 0$) is less, or at worst equal, to that of RPS (with $L = 0$). Or as L increases, the eavesdropping probability decreases.

Now let us consider the expression for eavesdropping probability for RPS, for $r = 2$ (pairwise communications). Also, by choosing $\xi = \frac{C}{n+2}$, where $0 \leq C \leq 2$, and replacing P with $\frac{k}{\xi} = \frac{k(n+2)}{C}$, we have

$$\begin{aligned} p_R(n, 2) &= (1 - \xi^2 (1-\xi)^n)^P \\ &\leq \exp\left(-\frac{kC(n+2-C)^n}{(n+2)^{(n+1)}}\right), \end{aligned} \tag{10}$$

which readily indicates that for RPS, for any desired probability of eavesdropping, as n increases, we have to increase k in a linear proportion. In other words, $k \not> O(n)$. Obviously, for HARPS too, as for the same ξ, $\varepsilon \geq \varepsilon_R$, $k \not> O(n)$. It is perhaps pertinent to point out here that Blundo et. al [4], with information theoretic arguments, have shown that it is *not* possible for any KPD to do better than $\binom{n+r-1}{r-1}$ preloaded keys per node (for an n-secure r-conference KPD), which for $r = 2$ (or pairwise key establishement) translates to $n+1$ or $O(n)$.

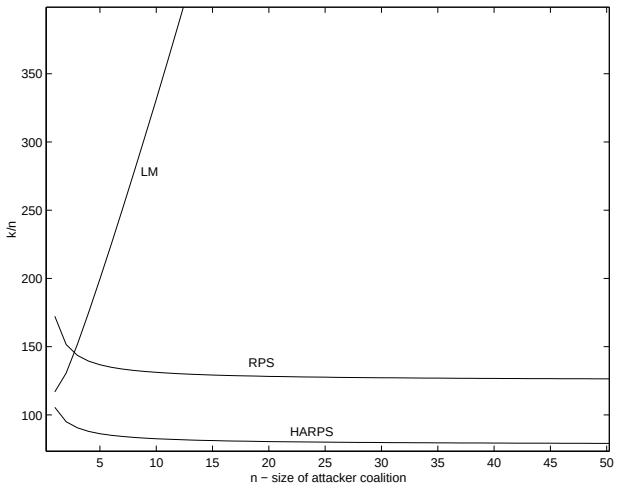

Fig. 1. Plot of $\frac{k}{n}$ vs n for HARPS, RPS and LM for $p = 10^{-20}$.

A. Optimization of Parameters

Optimization of random KPDSs is about minimizing the resource utilization in the nodes for a desired level of security. As all three random KPDSs have easily achievable computational requirements, the optimization focuses on minimizing k, the size of the key-ring, for a desired level of security (or desired $p(n, r)$) - we are not too concerned about the size of the key pool P. For a given value of k there are two parameters under the control of the designer - P (or equivalently, ξ), and the maximum hash depth L. Obviously, as value of the hash depth L is increased, p reduces. However, for $L > 64$ the pay-off is negligible.

¿From Eq (8), it is readily seen that for minimizing k we need to minimize $(1-\varepsilon)^{\frac{1}{\xi}}$. For the LM scheme there is no scope for optimization as ξ is fixed at one ($k = P$). For RPS and HARPS on the other hand, an optimal choice of ξ is very crucial. It can be easily shown that for both (RPS and HARPS), $\xi \propto \frac{1}{n}$ ($\xi n \approx 1.7$ for HARPS and $\xi n \approx 0.95$ for RPS).

Figure 1 is a comparison of HARPS, RPS and LM in terms of the number of preloaded keys (or the size of the key-ring k)

needed to achieve an eavesdropping probability of less than 10^{-20} for various values of the size of attacker's coalition n. The plot of $\frac{k}{n}$ vs n clearly depicts that $k > O(n^2)$ for LM and $k \approx O(n)$ for RPS and HARPS. Specifically, for **HARPS** $k \approx 75n$ and for **RPS** $k \approx 128n$, for $p < 10^{-20}$ and large n. Also, for the same ξ, an increase of k would result in a corresponding *exponential reduction* of p. The relations $k = O(n)$, and $\xi = \frac{k}{P} \propto \frac{1}{n}$, imply that $P \approx O(n^2)$ for RPS (and HARPS)[6]. For $p \leq 10^{-20}$, this implies $P \approx 42n^2$ for HARPS, and $P \approx 135n^2$ for RPS. However, the dependency $P = O(n^2)$ for RPS and HARPS is not a serious limitation - as P is the *number of keys that the TA needs to store - not the nodes*.

For the plots in Figure 1, the value of k is calculated by optimizing ξ for each value of n. However, optimization is performed *before* deployment. And compromises occur *after* deployment. Thus for a deployment of any KPD it is important to know how a system optimized for *some* value of n behaves for *larger values* of n. Figure 2 is a plot of the logarithm of the eavesdropping probability ($\log(p)$) vs n for HARPS, RPS and LM systems, all three designed to provide a eavesdropping probability of $p < 10^{-20}$ for $n = 20$ (the intersection of the 3 curves occurs at $p \approx 10^{-20}$ at $n = 20$). In addition to the *obvious advantage of needing the least value* of k to achieve this requirement, HARPS also has a *slower rate of "degradation of security"* with increasing n. For the systems compared, to compromise inter-nodal exchanges with a probability greater that 0.5, the attacker needs to compromise 220 nodes for HARPS ($P = 19390, k = 1610, L = 64$), and only 106 nodes for RPS ($P = 53840, k = 2565$). Also note that the comparison is not "fair" for HARPS (the comparison is between LM with $k = 12559$, RPS with $k = 2565$ and HARPS with $k = 1610$).

Note that $k = 1610$ is the *minimum* value of k needed to ensure an eavesdropping probability $p < 10^{-20}$ for $n = 20$ for HARPS. With a 10% increase in k and maximizing P (resulting in $P \approx 35,000$) to ensure that $p < 10^{-20}$ for $n \leq 20$, the attacker would need to compromise 300 nodes before he can achieve $p = 0.5$. For a 25% increase in k, the figure goes up to 371 nodes (a 70% increase from 220).

The justification for fixing $L = 64$ can be seen from the plot of the p vs n for$P = 19390, k = 1610$ for various values of L ranging from 32 to 512 in Figure 3. As expected the eavesdropping probability p reduces as L increases. But the improvement is only marginal. For instance, for HARPS with $P = 19390, k = 1610, L = 64$,the eavesdropping probability $p = 0.5$ when $n = 220$ (220 nodes are compromised). Increasing L to 512 provides only a marginal improvement (to $n = 233$ for $p = 0.5$). The small improvement perhaps does not warrant the increase in computational complexity that would result by increasing L.

[6]This can also be readily seen from Eq (10) be replacing k with $\xi P = \frac{CP}{n+2}$.

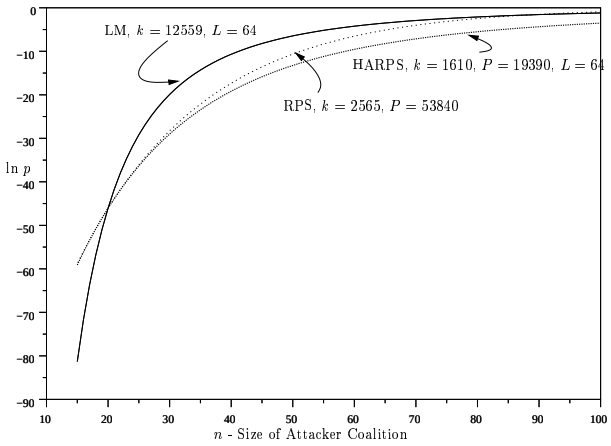

Fig. 2. Performance of HARPS, RPS and LM designed for $p \approx 10^{-20}$ at $n = 20$ for various values of n. Note the slower degradation of security for HARPS compared to RPS and LM.

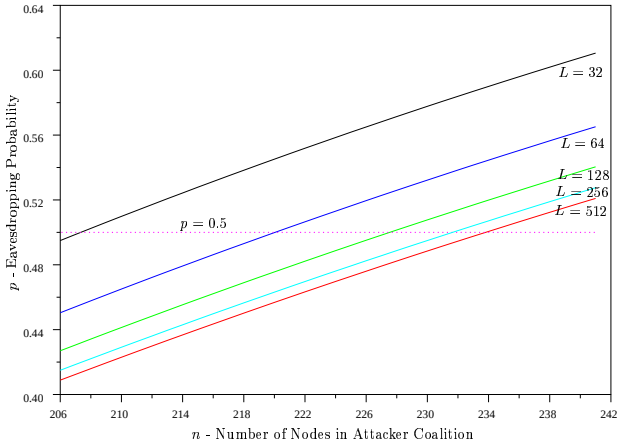

Fig. 3. Plots of eavesdropping probability vs n for various values of L. For all plots $P = 19390, k = 1610$.

IV. Security - Complexity Trade-offs in KPD Schemes

Apart from the broad classification of "deterministic" and "random", KPDs can be classified into many "categories" based on the relationship between n (the size of attacker's coalition) and p (the eavesdropping probability).

In the first category of KPD schemes, is the *basic* KDS, where each node is preloaded with *all possible shared secrets it might require* for secure communication with other nodes. For all possible ($\binom{N}{2}$) pairwise communications, each node requires $N - 1$ keys. If we permit group sizes of up to r, the total system secrets needed would be $\sum_{i=2}^{r} \binom{N}{i}$ and each node would need to store $\sum_{i=1}^{r-1} \binom{N-1}{i}$ secrets. While this may be impractical for large scale networks, this is the most secure of all KPD schemes. Irrespective of the number of nodes that have been compromised, the nodes that have *not* been compromised can continue to function securely. In other

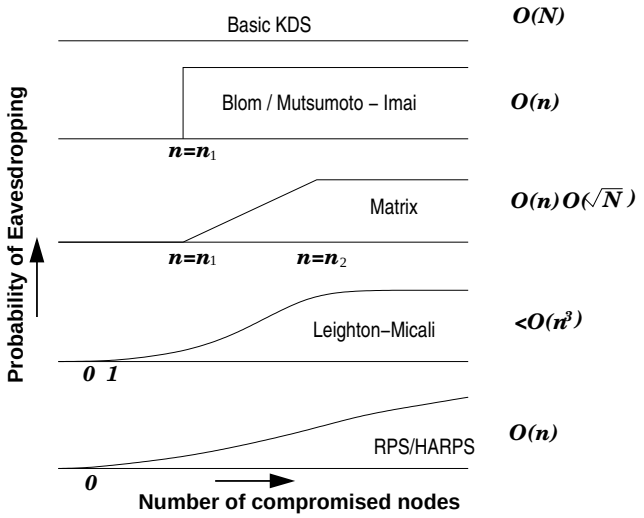

Fig. 4. A **qualitative** representation of the progression of KPD schemes with relaxation of security constraints. N is the network size and n represents the size of the attacker's coalition.

words for any n, the probability of eavesdropping is zero[7] (see Figure 3).

For the second category of KPD schemes, which typically use finite-field arithmetic techniques, the plot of p vs. n takes the form of a step function. For such schemes, the probability of eavesdropping is 0 as long as the size of the attacker's coalition n is *not greater* than some n_1. For $n > n_1$ however, the probability of eavesdropping is 1 - or the system is *completely compromised*. The obvious disadvantage of such schemes is the *catastrophic onset* of failure. Examples of such schemes are the Blom's KPD [3] (and its extension by Blundo [4] for $r > 2$), and the schemes proposed by Matsumoto et. al [5], and Stinson [10]. Another disadvantage of such schemes are that they are usually computationally expensive. For instance, for the scheme in [4], the nodes have to evaluate a n degree polynomial in $r-1$ variables, in a large prime (q) field $\mathbb{Z}_q$, to establish a shared group secret. The main advantage is that they are in general *very efficient* in terms of the size of the key-ring. Typically, $k \approx O(n)$. As the size of the key ring is not a function of the network size N, these schemes scale extremely well.

The third category of KPDs, the *subset intersection schemes* (SIS), address two major disadvantages of the second category - catastrophic onset of failure, and computational complexity. In SIS schemes, a pool of P keys (which constitute the system secrets) are distributed in a *deterministic* fashion such that each node gets k keys, $k < P$. The shared secret for a group of r nodes is simply a function of *all* keys in the intersection of the key-rings in the r nodes. For such a system resistant to attacker's coalition of n nodes, the keys are distributed such that the intersection of keys in r nodes is not contained in the union of any n nodes outside the group [6].

Thus, for the SIS schemes, the probability of eavesdropping is 0 as long as the size of the attacker's coalition $n \leq n_1$. For $n_1 \leq n \leq n_2$ the probability of eavesdropping gradually increases from 0 to 1. While this graceful degradation is a very desirable property, this is achieved at the *expense of scalability*. For example, for the matrix [7] scheme, $k \approx n\sqrt{N}$. The dependency of the key-ring size on the network size translates into severe restrictions on scalability of the scheme.

Even though other SIS schemes exist [8] - [11] for which $k = O(n \log N)$ (and thus do not suffer as much in terms of scalability), such schemes involve *complex constructions* (for the deterministic allocation of keys). Thus for two nodes to establish their shared secret they might either need to *execute the computationally intensive construction algorithm* to determine their shared keys or *exchange messages containing the indexes of the keys* they possess (this would imply exchanging of P-bit messages). However, the option of exchanging indexes of keys has another serious disadvantage. The inability of nodes to *ascribe a sequence of keys to an ID* implies that mutual *authentication* of the IDs is *not* implicitly achieved.

Random KPD schemes are characterized by the existence of a finite eavesdropping probability for all n. At fist sight, permitting a finite eavesdropping probability may seem like a serious disadvantage. In practice, it is not. Even for a KPDs for which $p = 0$ for some n, the final shared secret is a (symmetric) key with a *finite number of bits*. For instance, if the shared secret is a 64-bit key, there does exist a finite probability ($\frac{1}{2^{64}} > 10^{-20}$) that an attacker can "pull the secret out of a hat" (without the need to compromise any node). Thus permitting a small eavesdropping probability is not a disadvantage as long p is *comparable to the security offered by the key-length* of the final shared key (say $p \approx 10^{-20}$ for 64 bit keys).

The first random KPD was proposed by Leighton and Micali (LM) [2]. For the LM scheme, for a desired probability of eavesdropping when n nodes have been compromised, we need $O(n^2) < k < O(n^3)$. Further, as n increases the eavesdropping probability increases gracefully. Thus by "relaxing" the security requirement (even though the relaxation has very little implication in practice if p is low enough) the LM scheme is able to achieve what the subset intersection schemes set out to achieve, without the disadvantage of the dependency of k on the network size N. Further, the LM scheme offers a great deal of flexibility for increasing group size r by compromising some security. Additionally, unlike the subset intersection schemes, only the IDs need to be exchanged to establish shared secrets.

Several random KPD schemes, based on the same idea as SIS schemes, with a twist that the allocation of subset of keys to every node is done in a *random or pseudo random fashion* (instead of the deterministic allocation in SIS schemes) have been proposed in the recent past [1],[12]-[17]. We refer to these schemes *collectively as random preloaded subsets (RPS)*.

One of the major distinctions between [1], [16] and the methods in [12], [13] and [17] is the way shared secrets are discovered. In [12],[13] and [17] the shared secrets are dicovered through an interactive process, while in [1] and [16]

[7]The definition of p, *excludes* the n nodes from being part of any group.

it is achieved by just exchanging IDs - as the index of secrets preloaded in each node is derived from a one way function seeded by the ID. The approach of an interactive process has two disadvantages - (1) the additional bandwidth required and (2) implicit authentication of IDs is not achieved. Apart from this minor distinction the basic technique employed are however, similar. All use the concept of random preloaded subsets, and hence collectively referred to as RPS.

Like LM, RPS and its extension HARPS, proposed in this paper, also have a finite probability of eavesdropping p when one or more nodes are compromised. However, RPS and HARPS go even further by *permitting a small outage probability* $p_o = (1 - p_d) > 0$.

This additional relaxation of security requirements (by permitting a small outage probability) however goes a long way in reducing the complexity of the KPD. It is ther outage probability that gives RPS and HARPS the needed freedom to obtain drastic improvements in efficiency - from $k > O(n^2)$ for LM to $k \not> O(n)$. Once again, as long as the eavesdropping probability (which includes the outage probability) is of the same order as the security provided by the bit-length of the shared key, this is not an issue. Figure 6 depicts the *qualitative* progression of KPDs as a function of the probability of eavesdropping and the number of compromised nodes, under varying security assumptions.

It is pertinent to mention here that the methods in [15] and [17] are a little different from other RPS based schemes. In [15] and [17] (which are very similar methods), the authors combine RPS with Blom's [3] KPD.

While for a specific n (attacker coalition size), the schemes in [15] and [17] can be substantially more efficient than RPS (or even HARPS), their security of *deteriorate at a more rapid rate than RPS* [18] (while the degradation of HARPS is *more graceful* than RPS). This is not surprising considering that a combination of two systems - one deteriorating gracefully (RPS), with one that deteriorates catastrophically (Blom's scheme) would produce a system that deteriorates faster than RPS - though not as catastrophically as Blom's scheme.

V. Conclusions

HARPS, a novel random KPD scheme introduced in this paper, significantly outperforms two previously proposed random KPD schemes (which are also identified as special cases of HARPS), and also has many advantages over other existing KPD schemes. As HARPS needs only $O(n)$ keys to be stored in each node, it scales extremely well. There is practically no limit on the network size N that HARPS can support.

Seemingly, the main disadvantage of HARPS (and RPS) is the existence of an outage probability (which was also included in the measure of eavesdropping probability). However, in practice, it suffices to keep the eavesdropping probability to low values. HARPS and RPS utilize this freedom to permit an outage probability in order to achieve their high efficiency ($k = O(n)$).

While the advantages of HARPS and RPS over LM is very clear, ($k = O(n)$ vs $k = O(n^3)$), HARPS also has several advantages over RPS. Firstly, it is more efficient in terms of number of preloaded keys needed. Secondly, the deterioration of performance of HARPS as more nodes are compromised is slower than that of RPS. Thirdly, HARPS can be readily extended to a tree hierarchical deployment. Each node by itself could act as a TA and distribute a subset of it's keys (hashed further) to it child nodes. Even though a tree hierarchical distribution is also possible in RPS, RPS does not provide the much desired "seperation" between levels. In other words in a tree hierarchical deployment of HARPS, compromise of nodes in a lower level have no impact on the security of nodes at a higher level (as long as the hash function is pre-image resistant). However, this is not the case for RPS.

References

[1] M. Ramkumar, N. Memon, R. Simha, "Pre-Loaded Key Based Multicast and Broadcast Authentication in Mobile Ad-Hoc Networks," Globecom-2003.

[2] T. Leighton, S. Micali, "Secret-key Agreement without Public-Key Cryptography,"*Advances in Cryptology* - CRYPTO 1993, pp 456-479, 1994.

[3] R. Blom, "An Optimal Class of Symmetric Key Generation Systems," *Advances in Cryptology: Proc. of Eurocrypt 84*, Lecture Notes in Computer Science, **209**, Springer-Verlag, Berlin, pp. 335-338, 1984.

[4] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences," Lecture Notes in Computer Science, vol 740, pp 471–486, 1993.

[5] T. Matsumoto, M.E.Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, **IT-22**(6), Dec. 1976, pp.644-654.

[6] P. Erdos, P. Frankl, Z. Furedi, "Families of Finite Sets in which no Set is Covered by the Union of r Others," *Isreal Journal of Mathematics*, **51**, pp 79–89, 1985.

[7] L. Gong, D.J. Wheeler, "A Matrix Key Distribution Scheme," *Journal of Cryptology*, **2**(2), pp 51-59, 1990.

[8] C.J. Mitchell, F.C. Piper, "Key Storage in Secure Networks," *Discrete Applied Mathematics*, **21** pp 215–228, 1995.

[9] M. Dyer, T. Fenner, A. Frieze and A. Thomason, "On Key Storage in Secure Networks," *Journal of Cryptology*, **8**, 189–200, 1995.

[10] D. R. Stinson, T. van Trung, "Some New Results on Key Distribution Patterns and Broadcast Encryption," *Designs, Codes and Cryptography*, **14** (3) pp 261–279, 1998.

[11] C. Padro, I. Gracia, S. Martin, P. Morillo, "Linear Broadcast Encryption Schemes," *Discrete Applied Mathematics*, **128**(1) pp 223–238, 2003.

[12] L. Eschenauer, V.D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," Proceedings of the Ninth ACM Conference on Computer and Communications Security, Washington DC, pp 41-47, Nov 2002.

[13] H. Chan, A. Perrig, D. Song, "Random Key Pre-distribution Schemes for Sensor Networks," IEEE Symposium on Security and Privacy, Berkeley, California, May 2003.

[14] R. Di Pietro, L. V. Mancini, A. Mei, "Random Key Assignment for Secure Wireless Sensor Networks," 2003 ACM Workshop on Security of Ad Hoc and Sensor Networks, October 2003.

[15] W. Du, J. Deng, Y.S. Han. P.K.Varshney, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks," Proceedings of the 10th ACM Conference on Computer and Communication Security, pp 42–51, 2003.

[16] S. Zhu, S. Xu, S. Setia S. Jajodia, "Establishing Pair-wise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach," Proc. of the 11th IEEE International Conference on Network Protocols (ICNP'03), Atlanta, Georgia, November 4-7, 2003.

[17] D. Liu, P.Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," Proceedings of the 10th ACM Conference on Computer and Communication Security, Washington DC, 2003.

[18] M. Ramkumar, N. Memon, "An Effi cient Key Predistribution Scheme for Ad Hoc Network Security," IEEE Journal on Selected Areas of Communication, to appear, March 2005.