HARPS - Hashed Random Preloaded Subset Key Distribution

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

Abstract

In this paper, we introduce HAshed Random Preloaded Subset (HARPS) key distribution, a scalable key predistribution scheme employing only symmetric crypto primitives. HARPS is ideally suited for resource constrained nodes that need to operate without a trusted authority (TA) for extended periods (as is the case for nodes forming mobile ad hoc networks (MANETs)). The performance of HARPS is compared with that of two other key predistribution schemes. The first, RPS [1], is a based on random intersection of keys preloaded in nodes. The second, is (a slight modification to) a scheme proposed by Leighton and Micali (LM) in [2]. HARPS is a generalization of both RPS and LM. All the three schemes, rely on some degree of resistance to hardware tampering, and have probabilistic measures for the "merit" of the system. The merit of the schemes is a function of the probability that an attacker who has compromised n nodes (or has access to keys buried in n nodes) can "eavesdrop" on a conversation between r nodes ($r = 2$ for unicast communications). We analyze and compare the performance of the three schemes for unicast and multicast communications. We show that HARPS has significant performance advantage over RPS and LM.

1 Introduction

In many evolving applications, there is a paradigm shift to distributed rather than a centralized mode of operation. It seems imperative in such applications involving distributed computing, to have efficient means of developing "trust" between "strangers". For example, "strange" mobile nodes forming ad hoc networks (MANETs) have to perform authenticated exchanges for "higher purposes" - perhaps for building a routing table, or relaying messages between other nodes. In such applications, malicious action by a single node could have a potentially disruptive effect over the entire network. The needed trust could be provided by a suitable key management scheme which implicitly provides authentication and encryption.

Applications involving mobile nodes have some special requirements. Typically

1. Due to resource constraints in mobile nodes, the protocol for secure communication should not use asymmetric crypto primitives.

2. The nodes need to operate for extended periods without a TA. More specifically, for their "normal" course of operations the nodes should not need to communicate with the TA.

3. The nodes should be able to communicate with any other node instantaneously. This property is very useful under the event of jamming.

4. Lastly, and perhaps one of the most important requirement - scalability. The key distribution system should ideally be independent of the total number of nodes in the system. By the very nature of the application, it may be very difficult to predict the number of nodes. New nodes may be added at any time. Addition of new nodes should not require any kind of *reconfiguration of the system.*

If asymmetric cryptography cannot be used, and if the nodes cannot rely on a central TA, then it is intuitive that some form of *key predistribution* is needed. In any key predistribution scheme, k secrets are preloaded in each node by a TA. The secrets are chosen in such a way that any two nodes wishing to communicate can *independently* arrive at a session key based on the secrets they possess. However, no node should be able to arrive at the session key of two *other* nodes.

HARPS is a key predistribution scheme that employs only symmetric crypto primitives. HARPS is highly scalable - the scheme does not depend on the number of nodes in the system. In addition, HARPS does not rely on a TA during its normal mode of operation. A set of nodes wishing to communicate can establish a secure channel instantaneously. HARPS is also renewable. However, renewal of the keys need interactions with the TA.

In HARPS, the TA generates P secret keys also referred to as the "root" keys. From each root key, upto L other keys may be derived by repeated hashing of the root keys. A set of k keys preloaded in each node is arrived at by selecting a subset of the $k \leq P$ keys from the pool, and hashing each key a variable number of times. A set of r nodes wishing to communicate can independently arrive at a session key by calculating the intersection of the "root" keys (corresponding to the derived keys they possess), and hashing forward to reach the maximum "hash depth" for each root key among the corresponding derived keys in the r nodes. All such maximum hash depth keys are concatenated and hashed to yield the session key. Two key predistribution schemes proposed in literature, RPS [1] and LM [2] turn out to be special cases of HARPS.

The rest of the paper is organized as follows. In the next section we introduce and analyze HARPS. In Section 3 we review some prior work, including RPS and LM. In Section 4 we evaluate and compare the performance of HARPS, RPS and LM key predistribution schemes. Conclusions are presented in Section 5.

2 HARPS

HARPS is defined by 3 parameters (P, k, L), and two public functions - $h()$, a cryptographic hash (one-way) function and $F_{\mathcal{H}}()$, a public key generation function. The parameter P is the number of secret keys in a "key pool". The parameter k is the number of keys preloaded in each node. The parameter L is the maximum "hash depth". Further, each node is given a unique ID.

The TA generates P secret keys $[M_1 \cdots M_P]$. We shall refer to these P keys as the "root keys". The one way function $h()$ is used to derive more keys by repeated application of $h()$ on the root keys. The parameter L is the maximum number of times the function $h()$ may be applied. Thus from each root key one can get L "derived keys". The j^{th} derived key from the i^{th} root key is represented as K_i^j, where $1 \leq i \leq P$ and $1 \leq j \leq L$. More formally

$$K_i^j = h^j(M_i) \tag{1}$$

where $h^n(X) = \overbrace{h(\cdots h(h(X))\cdots)}^{n \text{ times}}$.

A set of k derived keys are preloaded in each node. The choice of the derived keys is determined by the "public key" of the node. The public key, in turn, is determined by the node ID and the public key generation function $F_{\mathcal{H}}()$. The public key is a set of k ordered pairs, $(I_1, D_1) \cdots (I_k, D_k)$. The first "coordinate" of the ordered pairs, $I_1 \cdots I_k$ is a partial *random permutation sequence* of integers between 1 and P. Therefore $1 \leq I_j \leq P \, \forall 1 \leq j \leq k$ and $I_j \neq I_i, i \neq j$. Without any loss of generality, the sequence $I_1 \cdots I_k$ can be considered as the first k numbers in a random permutation sequence of numbers $1 \cdots P$. The second coordinate $D_1 \cdots D_k$ is a sequence of uniformly distributed numbers between 1 and L. The public key of a node A with ID ID_A is therefore derived as follows:

$$(I_{1_A}, D_{1_A}) \cdots (I_{k_A}, D_{k_A}) = F_{\mathcal{H}}(ID_A). \tag{2}$$

The function $F_{\mathcal{H}}()$ may thus be assumed to be a random sequence generator seeded by the ID. A uniform random sequence is used for generating the second coordinate (depth). A (possibly different) uniform random sequence may also be used to obtain a random permutation of integers between 1 and P, from which the first coordinates are derived.

The preloaded keys in a node A are therefore

$$[K_{I_{1_A}}^{D_{1_A}} \cdots K_{I_{k_A}}^{D_{k_A}}] = [h^{D_{1_A}}(M_{I_{1_A}}) \cdots h^{D_{k_A}}(M_{I_{k_A}})]. \tag{3}$$

Let $[(I_{1_B}, D_{1_B}) \cdots (I_{k_B}, D_{k_B})] = F_{\mathcal{H}}(ID_B)$ be the public key of node B. With the knowledge of the node IDs, the two nodes can independently arrive at the shared *root* keys as

$$[M_{s_1} \cdots M_{s_m}] = [M_{I_{1_A}} \cdots M_{I_{k_A}}] \bigcap [M_{I_{1_B}} \cdots M_{I_{k_B}}] \tag{4}$$

Let $[d_{1_A} \cdots d_{m_A}]$ and $[d_{1_B} \cdots d_{m_B}]$ be the hash depths of the root keys $[M_{s_1} \cdots M_{s_m}]$ common to nodes A and B. In other words, m if k keys in node A are

$$[K_{s_1}^{d_{1_A}} \cdots K_{s_m}^{d_{m_A}}] \tag{5}$$

and in node B

$$[K_{s_1}^{d_{1_B}} \cdots K_{s_m}^{d_{m_B}}]. \tag{6}$$

Let $d_1 = \max(d_{1_B}, d_{1_A}) \cdots d_m = \max(d_{m_B}, d_{m_A})$. The session key K_{AB} is then

$$K_{AB} = h(h^{d_1}(M_{s_1}) | h^{d_2}(M_{s_2}) | \cdots | h^{d_m}(M_{s_m})) \tag{7}$$

The session key is calculated by node A as

$$K_{AB} = h(h^{(d_1 - d_{1_A})}(K_{s_1}^{d_{1_A}}) | \cdots | h^{(d_m - d_{m_A})}(K_{s_m}^{d_{m_A}})) \tag{8}$$

and by node B as

$$K_{AB} = h(h^{(d_1 - d_{1_B})}(K_{s_1}^{d_{1_B}}) | \cdots | h^{(d_m - d_{m_B})}(K_{s_m}^{d_{m_B}})) \tag{9}$$

Figure 2 is an illustration of HARPS for $P = 8$, $k = 4$ and $L = 4$. In the example, Alice and Bob share $m = 2$ keys (root keys M_2 and M_5. The index of the shared keys are $s_1 = 2, s_2 = 5$, and their corresponding hash depths, $d_{1_A} = 3, d_{1_B} = 2, d_1 = \max(3,2) = 2$, and $d_{2_A} = 2, d_{2_B} = 3, d_2 = \max(3,2) = 2$. Note that both nodes can *independently* arrive at the shared keys K_{AB}.

For the case of multicasting between r nodes with public keys

$$\{(I_{1_1}, D_{1_1}) \cdots (I_{k_1}, D_{k_1})\} \cdots \{(I_{1_r}, D_{1_r}) \cdots (I_{k_r}, D_{k_r})\},$$

let

$$[M_{s_1} \cdots M_{s_m}] = [M_{I_{1_1}} \cdots M_{I_{k_1}}] \bigcap [M_{I_{1_2}} \cdots M_{I_{k_2}}] \bigcap \cdots \bigcap [M_{I_{1_r}} \cdots M_{I_{k_r}}]. \tag{10}$$

be the shared root keys and $[d_{1_1} \cdots d_{m_1}] \cdots [d_{1_r} \cdots d_{m_r}]$ the corresponding hash depths. Also, let $d_i = \max(d_{i_1} \cdots d_{i_r})$. The session key $K_{1 \cdots r}$ is given by (identical to Eq (7))

$$K_{1 \cdots r} = h(h^{d_1}(M_{s_1}) | h^{d_2}(M_{s_2}) | \cdots | h^{d_m}(M_{s_m})). \tag{11}$$

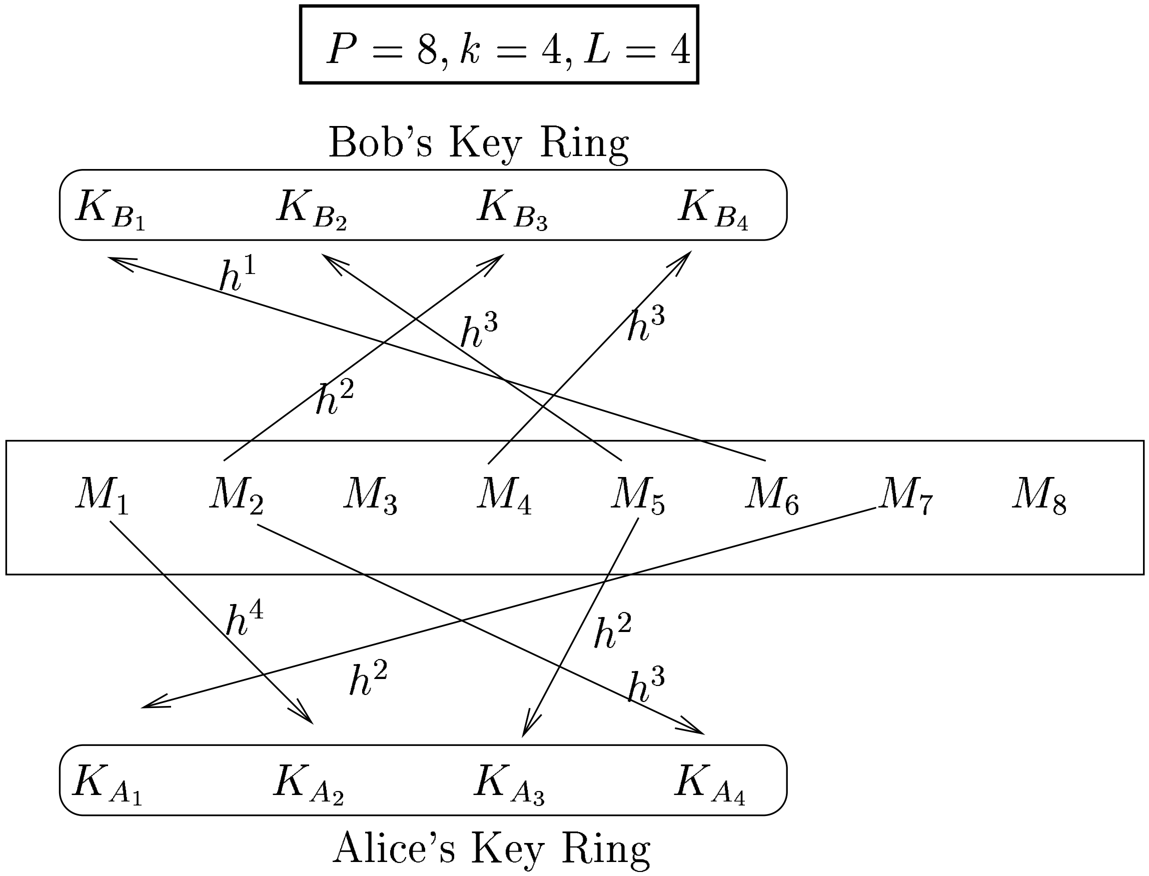
For an "attacker" with access to keys buried in one n nodes, with public keys

$$\{(I_{1_1}^a, D_{1_1}^a) \cdots (I_{k_1}^a, D_{k_1}^a)\} \cdots \{(I_{1_n}^a, D_{1_n}^a) \cdots (I_{k_n}^a, D_{k_n}^a)\},$$

to be able to "eavesdrop" on a conversation between r nodes with public keys

$$\{(I_{1_1}, D_{1_1}) \cdots (I_{k_1}, D_{k_1})\} \cdots \{(I_{1_r}, D_{1_r}) \cdots (I_{k_r}, D_{k_r})\}, \tag{12}$$

the following conditions should be satisfied:



Shared Root Keys M_2, M_5

Max Hash Depth of Shared Keys
$M_2 \rightarrow 3, M_5 \rightarrow 3$

$K_{AB} = h(h^3(M_2)|h^3(M_5))$

Alice Derives K_{AB} as
$K_{AB} = h(K_{A_4}|h^1(K_{A_3}))$

Bob Derives K_{AB} as
$K_{AB} = h(h^1(K_{B_3})|K_{B_2})$

Public Keys

Alice	Bob
$(7, 2)$	$(6, 1)$
$(1, 4)$	$(5, 3)$
$(5, 2)$	$(2, 2)$
$(2, 3)$	$(4, 3)$

Figure 1: Illustration of HARPS for $P = 8, k = 4, L = 4$.

1. **C1**: If $[M_{s_1} \cdots M_{s_m}]$ are the root keys shared by the r nodes involved in a multicast (for unicast communications $r = 2$), then the union of nk keys from the compromised nodes (the attacker's pool of nk keys) should contain *at least* one derived key for each of the root keys $[M_{s_1} \cdots M_{s_m}]$.

2. **C2**: Given that an attacker has at least one derived key for each node, the attacker can have a maximum of n derived keys for each of the shared root keys. In the attacker's pool of nk keys, let $n_{e^j}, 1 \leq j \leq m$, be the number of occurrences of derived keys corresponding to each of the m shared root keys $[M_{s_1} \cdots M_{s_m}]$, and the corresponding depths $d^a_{j_1} \cdots d^a_{j_{n_e^j}}, 1 \leq j \leq m$. Further, let $d_{j_1} \cdots d_{j_r}$ be the hash depth of the j^{th} shared root key amongst the r nodes involved in the multicast. The condition to be satisfied for successful eavesdropping is

$$\min(d^a_{j_1} \cdots d^a_{j_{n_e^j}}) \leq \max(d_{j_1} \cdots d_{j_r}) \forall 1 \leq j \leq m. \tag{13}$$

In the next section we shall analytically evaluate the probability of eavesdropping for HARPS.

2.1 Analysis of HARPS

- Let $\mathcal{U}_P$ represent a set of cardinality P. Mathematically

$$|\mathcal{U}_P| = P. \tag{14}$$

- Let $\mathcal{A}^j_k$ represent a subset (with index j and cardinality k), of $\mathcal{U}_P$. The set $\mathcal{A}^j_k$ is obtained by *randomly* choosing k elements from the set $\mathcal{U}_P$ without replacement.

$$\begin{aligned} \mathcal{A}^j_k &\in \mathcal{U}_P \\ |\mathcal{A}^j_k| &= k \end{aligned} \tag{15}$$

- Let $p_{S^{k_1,k_2}_m}$ represent the probability that the intersection $\mathcal{A}_{k_2} \bigcap \mathcal{A}_{k_1}$ has a cardinality of m.

$$p_{S^{k_1,k_2}_m} = \Pr\left\{|\mathcal{A}_{k_2} \bigcap \mathcal{A}_{k_1}| = m\right\} \tag{16}$$

- $p_{S^r_m}$ represent the probability that the intersection $\mathcal{A}^1_k \bigcap \cdots \bigcap \mathcal{A}^r_k$ of r sets, has a cardinality m.

$$p_{S^r_m} = \Pr\left\{|\mathcal{A}^1_k \bigcap \cdots \bigcap \mathcal{A}^r_k| = m\right\} \tag{17}$$

  Let ϕ_r represent the *expected* value of m. Or, $\phi_r = E[m] = \sum_{m=1}^{k} m p_{S^r_m}$.

- Let $p_{C^q_{n,k}}$ represent the probability that the union $\mathcal{A}^1_k \bigcup \cdots \bigcup \mathcal{A}^n_k$, of n sets has a cardinality q, where $k \leq q \leq q_{max} = \min(nk, P)$.

$$p_{C^q_{n,k}} = \Pr\left\{|\mathcal{A}^1_k \bigcup \cdots \bigcup \mathcal{A}^n_k| = q\right\}. \tag{18}$$

  Let $\theta_n = E[q] = \sum_{q=k}^{q_{max}} q p_{C^q_{n,k}}$, be the expected value of q;

- Let

$$p_{E^q_m} = \Pr\left\{\mathcal{A}_m \in \mathcal{A}_q\right\} \tag{19}$$

  where $\mathcal{A}_m \in \mathcal{U}_P$ and $\mathcal{A}_q \in \mathcal{U}_P$ are arbitrarily chosen sets of cardinality m and q respectively.

- Let

$$p_{E_{\mathcal{R}}}(P,k,n,r) = \Pr\left\{\{\mathcal{A}_k^1 \bigcap \cdots \bigcap \mathcal{A}_k^r\} \in \{\mathcal{A}_k^{r+1} \bigcup \cdots \bigcup \mathcal{A}_k^{r+n}\}\right\}. \tag{20}$$

  In other words, $p_{E_{\mathcal{R}}}(P,k,n,r)$ represents the probability that the union of n sets $\mathcal{A}_k^{r+1} \bigcup \cdots \bigcup \mathcal{A}_k^{r+n}$ contains the intersection of r other sets $\mathcal{A}_k^1 \bigcap \cdots \bigcap \mathcal{A}_k^r$.

- Let $1 \leq a_{i_j}, b_{i_j} \leq L$ represent integers which uniformly distributed random variables between 1 and L, where i and j arbitrary indices. Let

$$\begin{aligned} a_i &= \min(a_{i_1} \cdots a_{i_{n_e}}), & (21)\\ b_i &= \max(b_{i_1} \cdots b_{i_r}), 1 \leq i \leq m. & (22) \end{aligned}$$

  Let $p_{E_{\mathcal{LM}}}(L,m,n_e,r)$ represent the probability that $a_i \leq b_i \forall 1 \leq i \leq m$. Or

$$p_{E_{\mathcal{LM}}}(L,m,n_e,r) = \Pr\{a_i \leq b_i \forall 1 \leq i \leq m\}. \tag{23}$$

It can be easily shown [1] that

$$\begin{aligned} p_{S_m^{k_1,k_2}} &= \frac{\binom{k_1}{m}\binom{P-k_1}{k_2-m}}{\binom{P}{k_2}} = p_{S_m^{k_2,k_1}} = \frac{\binom{k_2}{m}\binom{P-k_2}{k_1-m}}{\binom{P}{k_1}} & (24)\\ p_{E_m^q} &= \frac{\binom{P-m}{q-m}}{\binom{P}{q}} = \frac{(P-m)!q!}{(q-m)!P!} & (25) \end{aligned}$$

and

$$\begin{aligned} p_{S_m^2} &= p_{S_m^{k,k}} & (26)\\ p_{S_m^3} &= \sum_{i=m}^{k} p_{S_i^{k,k}} \frac{\binom{i}{m}\binom{P-i}{k-m}}{\binom{P}{k}}. & (27)\\ p_{S_m^4} &= \sum_{i=m}^{k} p_{S_i^{k,k}} \sum_{j=m}^{i} \frac{\binom{i}{j}\binom{P-i}{k-j}}{\binom{P}{k}} \frac{\binom{j}{m}\binom{P-j}{k-m}}{\binom{P}{k}} & (28)\\ &\vdots \end{aligned}$$

Further, for $n = 1, 2, 3,$

$$\begin{aligned} p_{C_{1,k}^q} &= \delta(q-k) & (29)\\ p_{C_{2,k}^q} &= p_{S_{2k-q}^{k,k}} & (30)\\ p_{C_{3,k}^q} &= \sum_{i=0}^{i_{max}} p_{S_{k-i}^{k,k}} * p_{S_{2k-q+i}^{k+i,k}} & (31) \end{aligned}$$

respectively. The probability $p_{E_{\mathcal{R}}}(P,k,n,r)$ is therefore

$$p_{E_{\mathcal{R}}}(P,k,n,r) = \sum_{q=k}^{q_{max}} p_{C_{n,k}^q} \sum_{m=0}^{k} p_{S_m^r} p_{E_m^q}. \tag{32}$$

However, for higher n it becomes cumbersome to obtain the exact expression for $p_{C^q_{n,k}}$. To avoid obtaining the exact expression for $p_{C^q_{n,k}}$, we could use a first order approximation[1] of Eq (32) for large n, viz.,

$$p_{E_{\mathcal{R}}}(P,k,n,r) \approx \tilde{p}_{E_{\mathcal{R}}}(P,k,n,r) = \sum_{m=0}^{k} p_{S^r_m} p_{E^{\theta_n}_m}. \tag{33}$$

where θ_n can be obtained by a simple recursive equation

$$\theta_n = \theta_{n-1} + \frac{k}{P}(P - \theta_{n-1}) \tag{34}$$

starting with $\theta_0 = 0$.

Similar to the first order approximation for $p_{E_{\mathcal{R}}}(P,k,n,r)$ for large values of n, it is also possible to obtain an approximation for large values of r based on the *expected value* of the cardinality of the intersection of r sets[2], ϕ_r. It can be easily seen that

$$\phi_r = \frac{k^r}{P^{r-1}}. \tag{35}$$

Now we can define, for large r,

$$p_{E_{\mathcal{R}}}(P,k,n,r) \approx \hat{p}_{E_{\mathcal{R}}}(P,k,n,r) = \sum_{q=k}^{q_{max}} p_{C^q_{n,k}} p_{E^q_{\phi_r}}. \tag{36}$$

and for large r and n,

$$p_{E_{\mathcal{R}}}(P,k,n,r) \approx \ddot{p}_{E_{\mathcal{R}}}(P,k,n,r) = p_{E^{\theta_n}_{\phi_r}}. \tag{37}$$

As mentioned in the previous section conditions **C1** and **C2** have to be satisfied for successful eavesdropping by an attacker. The probability $p_{E_{\mathcal{R}}}(P,k,n,r)$ is the probability that condition **C1** is satisfied. Let $p_{E_{\mathcal{LM}}}(L,m,n_e,r)$ be the probability that condition **C2** is satisfied.

Since there are P root keys which are equally likely to appear in the attacker's pool of nk keys, the expected value of the number of occurrences of each root key is $n_e = \frac{nk}{P}$. Obviously $E[n_{e^j}] = n_e = \frac{nk}{P}$. Therefore

$$\Pr\{\min(d^a_{j_1} \cdots d^a_{j_{n^j_e}}) \geq \max(d_{j_1} \cdots d_{j_r}) \forall j\} \approx \Pr\{\min(d^a_{j_1} \cdots d^a_{j_{n_e}}) \geq \max(d_{j_1} \cdots d_{j_r}) \forall j\} \tag{38}$$

It is easy to see that the probability

$$\Pr\{b_j = w\} = \frac{w^r - (w-1)^r}{L^r}, 1 \leq w \leq L. \tag{39}$$

where b_j is defined in Eq (22). Now,

$$\begin{aligned} \Pr\{a_j > b_j\} &= \sum_{i=1}^{L-1} \Pr\{b_j = i\} \prod_{w=1}^{n_e} \Pr\{a_{w_j} > i\} \\ &= \sum_{i=1}^{L-1} \left[\frac{i^r - (i-1)^r}{L^r} \prod_{w=1}^{n_e} \frac{L-i}{L} \right] \\ &= \sum_{i=1}^{L-1} \left[\frac{i^r - (i-1)^r}{L^r} \left(\frac{L-i}{L} \right)^{n_e} \right] = g_{LM}(L, n_e, r). \end{aligned} \tag{40}$$

[1] as n increases $p_{C^q_{n,k}}$ would approach an impulse function. So the approximation is justified for large n.

[2] as r increases $p_{S^r_m}$ would approach an impulse function. So, once again the approximation is justified for large r.

Therefore, from Eq (23)

$$p_{E_{\mathcal{L}M}}(L, m, n_e, r) = \Pr\{a_j \leq b_j, 1 \leq j \leq m\} = [1 - g_{LM}(L, n_e, r)]^m . \tag{41}$$

Thus the expressions for eavesdropping probability for HARPS is:

$$p_{E_{\mathcal{H}}}(P, k, L, n, r) \quad = \quad \sum_{q=k}^{q_{max}} p_{C^q_{n,k}} \sum_{m=0}^{k} p_{M^r_m} p_{E^q_m} p_{E_{\mathcal{LM}}}(L, m, n_e, r). \tag{42}$$

And the corresponding first order approximation (for large n)

$$p_{E_{\mathcal{H}}}(P, k, L, n, r) \approx \tilde{p}_{E_{\mathcal{H}}}(P, k, L, n, r) = \sum_{m=0}^{k} p_{M^r_m} p_{E^{\theta_n}_m} p_{E_{\mathcal{LM}}}(L, m, n_e, r) \tag{43}$$

For multicasting ($r > 2$), we can obtain approximations for the number of shared keys, ϕ_r, in a multicast of r nodes, resulting in

$$\begin{aligned} p_{E_{\mathcal{H}}}(P, k, L, n, r) \quad &\approx \quad \hat{p}_{E_{\mathcal{H}}}(P, k, L, n, r) = \sum_{q=k}^{q_{max}} p_{C^q_{n,k}} p_{E^q_{\phi_r}} p_{E_{\mathcal{LM}}}(L, \phi_r, n_e, r). & (44) \\ &\approx \quad \ddot{p}_{E_{\mathcal{H}}}(P, k, L, n, r) = p_{E^{\theta_n}_{\phi_r}} p_{E_{\mathcal{LM}}}(L, \phi_r, n_e, r). & (45) \end{aligned}$$

3 Prior Work

Refs. [3] and [4] use algebraic codes to facilitate key generation. A certain number of nodes need to collude to compromise one or all the keys. For instance, in Blom's scheme [3], resistant to collusion of n nodes, a central authority needs to transmit $n + 1$ elements to each node securely. The TA generates a polynomial

$$f(x, y) = \sum_{i=0}^{n} \sum_{j=0}^{n} a_{ij} x^i y^j \bmod P, \tag{46}$$

where $a_{ij} = a_{ji}$ are secret and P is a large prime. Every node is assigned a unique public ID. For instance, node A which has public ID r_A receives $g_A(x) = f(x, r_A)$ securely ($g_A(x)$ has $n + 1$ coefficients) from the TA. Two nodes A and B can calculate $K_{AB} = K_{BA} = f(r_A, r_B) = f(r_B, r_A) = g_A(r_B) = g_B(r_A)$. However, Blom's scheme is unsuitable for mobile nodes due to the computationally expensive operation of polynomial evaluation.

In [6], a set of N keys is distributed amongst N nodes. Each node is given a carefully selected set of $t\sqrt{N}$ keys. It is guaranteed that any two nodes have in common either $2t$ or $2t + \sqrt{N} - 2$ distinct keys. Two communicating nodes use a session key which is based on *all the keys they share*. However, the key distribution scheme in [6] is very closely tied to the number of nodes in the system (unlike Blom's scheme for instance), and thus does not scale very well. There also exists a certain number of groups of $t - 1$ other nodes which can collude to compromise the session key of any two nodes.

In Eschenauer et. al. [5], sensor nodes are preloaded with a randomly chosen set of k keys from a pool of P keys. Two nodes can exchange messages only if they share at least one key. The authors address the problem of connectivity of such a network, given an average of n neighbors for every node. The paper also addresses some schemes by which two nodes wishing to communicate determine their shared keys. However, unlike typical key distribution schemes, the authors do not consider "eavesdropping" (the ability of a unintended node being able to decipher the communication between two other nodes) as an issue.

In [7], Chan et al. propose a modification of the method by Eschenauer et. al. [5], called the q-composite random key distribution scheme where the nodes need to share at least q keys to form a secure link. The authors also address the issue of eavesdropping by an attacker who has compromised multiple nodes.

RPS [1] is a simple key distribution scheme which draws some ideas from both [5] and [6]. Like [5], the nodes are preloaded with randomly drawn k keys from a larger pool of P keys. However, unlike [5], two nodes do not need to go through a series of exchanges to determine the keys they share. In this respect, it is closer to the scheme in [6]. Also, the key used for encrypting communications between any two nodes is derived from *all the keys shared between the two nodes*, similar to the method in [6]. RPS was however, also generalized for secure multicast communications.

RPS is a special case of HARPS where $L = 0$. In other words there are no "derived" keys. A subset of root keys themselves are preloaded in the nodes. Therefore, for the expressions for probability of eavesdropping for RPS, only condition **C1** has to be satisfied. In fact, $p_{E_{\mathcal{R}}}(P, k, n, r)$ is the probability of eavesdropping for RPS. The "public key" for the RPS nodes is just the first coordinate $I_1 \cdots I_k$ instead of the k ordered pairs $(I_1, D_1) \cdots (I_k, D_k)$ for HARPS.

In the Leighton-Micali (LM) scheme [2], the TA has k secret "master keys". Each node is preloaded with k keys. The "public key" of each node is only the second coordinate $D_1 \cdots D_k$ of the k ordered pair used for HARPS. The i^{th} preloaded key is derived from the i^{th} master key through D_i applications of a one way function $h()$, where $1 \leq D_i \leq L$ are randomly chosen for each key (for each node). Thus the LM scheme is a special case of HARPS where $P = k$. The expression for the probability of eavesdropping in LM, which is very similar to Eq (41), is given by

$$p_{E_{\mathcal{LM}}}(L, k, n, r) = [1 - g_{LM}(L, n, r)]^k . \tag{47}$$

All communications share k "root" keys. An attacker with n compromised nodes has exactly n derived keys for each root key (unlike HARPS where the attacker has, on an average $n_e = \frac{nk}{P}$ keys).

It is pertinent to mention here that in [2], the authors obtain the *upper bound* for the expression $g_{LM}(L, n, 2)$ in Eq (41) as

$$p_{ub} = \max(g_{LM}(L, n, 2)) = \left(\frac{1}{n}\right)^2 \left(1 - \frac{1}{n}\right)^n \tag{48}$$

4 Results

In this section we compare the performance of the 3 schemes - RPS, LM and HARPS. For comparisons, the same value of k, the number of preloaded keys, is used for all three methods. Note that neither the value of P nor L has significant impact on the resources of each node. In all the figures $\log_{10} p_E$ represents the probability of eavesdropping. We omit the subscript $\mathcal{LM}, \mathcal{H}, \mathcal{R}$ as it will be clear from the context.

Figure 4 is a plot of the probability of eavesdropping for the LM scheme for $k = 256$, $r = 2$ (unicast) for different values of L. Note that for values of L greater than 64 not much improvement is obtained. The figure also shows a plot of the upper bound for the probability of eavesdropping which is calculated using Eq (48) for the value of $g_{LM}(L, n, r)$ in Eq (41).

As mentioned earlier, RPS is a special case of HARPS with $L = 0$. It can be immediately appreciated, that all other parameters being the same, the performance would improve with increasing L. Thus for some given P, k, n and r, HARPS (with $L > 1$) will *always* outperform RPS. A comparison of RPS and HARPS for two values of $\alpha = \frac{P}{k}$ (1.5 and 2), $k = 256$, $r = 2, 3$ and $n = 1 \cdots 5$ is shown in Figure 4. For HARPS $L = 64$, and for RPS $L = 0$.

Figure 4 depicts the probability of eavesdropping for HARPS for $k = 256$ and $L = 64$ for various values of $P = \alpha k$. The case where $P = k$ (or $\alpha = 1$) reduces to the LM scheme. Note that the HARPS has considerable performance advantage over LM, especially for larger values of n (number of compromised nodes). The main advantage of HARPS (and RPS) over LM stems from the dynamic dependence of the merit on the value of P (as opposed to the value of L in LM). Therefore in HARPS and RPS it is possible to choose P (or α) depending on the "nature of the application". For applications where the number of compromised nodes is expected to be high we can use large values of $\alpha = \frac{P}{k}$. However, the price paid for it

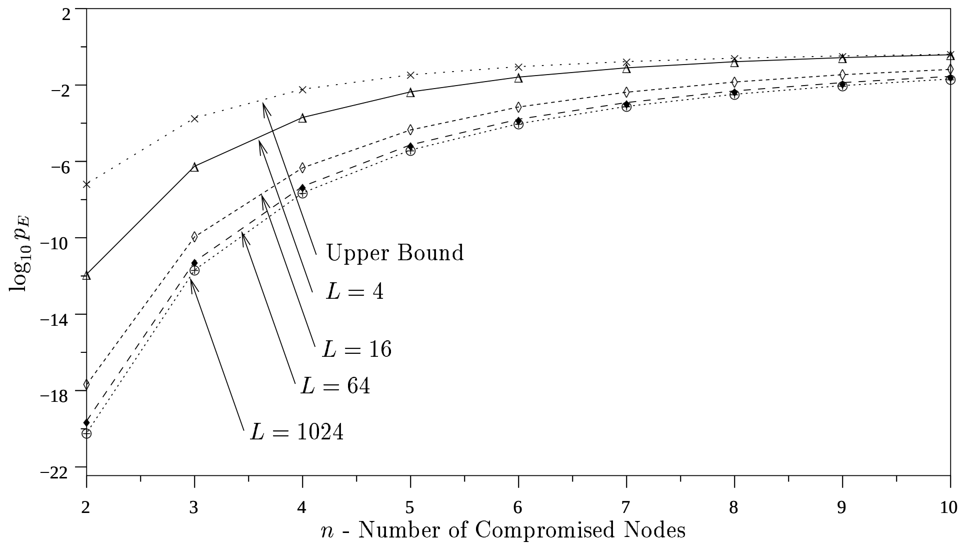


Figure 2: Probability of Eavesdropping $(p_{E_{\mathcal{L}M}}(L, k, n, r))$ for $k = 256$ and different values of L for $r = 2$. Also shown is the upper bound given in [2].

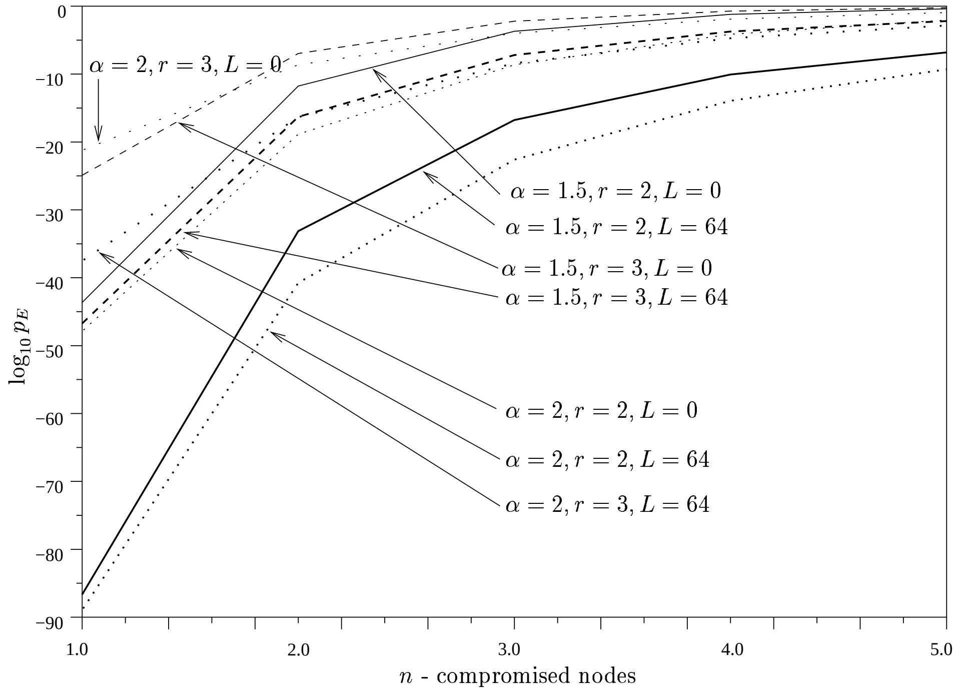


Figure 3: Comparison of HARPS and RPS for $n = 1 \cdots 5$, $r = 2, 3$, and $\alpha = 1.5, 2$. For HARPS $L = 64$, and for RPS $L = 0$.

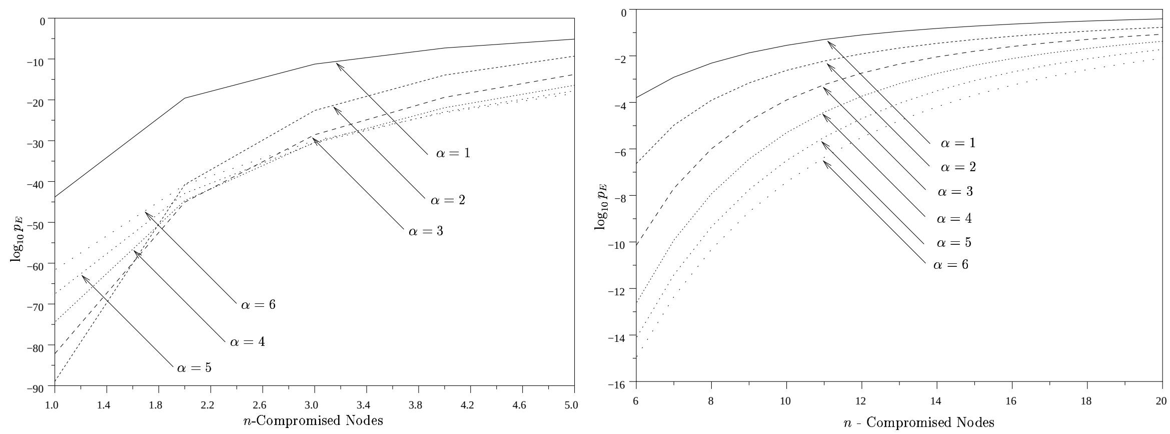


Figure 4: Probability of Eavesdropping for HARPS for $n = 1 \cdots 20$ and $r = 2$ for $\alpha = 1 \cdots 6$. The case of $\alpha = 1$ corresponds to LM.

is reduced merit for the case when the attacker has compromised fewer keys (as opposed to the case when α is small). Such a trade-off is not possible for the LM scheme.

Figure 4 compares HARPS and LM for the case of multicasting. Multicasting severely restricts the size of P (or α), as for effective multicasting the r communicating nodes should share many keys. The conflicting requirements of increasing α for higher merit when the number of compromised nodes increase and decreasing α for secure multicasting renders HARPS not very effective for large values of n (in other words the best operating point for HARPS in this case would collapse to that of LM).

Figure 4 depicts the performance of HARPS for $n = 1$ for $r = 2 \cdots 20$ for various values of α. Note that for this case HARPS (with an appropriately chosen α) performs significantly better than LM. As r increases we need to reduce the value of α to increase the merit (or reduce probability of eavesdropping).

Now let us consider the effect of k on the security of the system. For the LM scheme it is immediately obvious from Eq (47) that the probability of eavesdropping reduces exponentially with k. In [1] it was shown that a similar trend was observed for RPS too. Figure 4 depicts the exponential relationship between k and the probability of eavesdropping (or linear relationship between k and $\log_{10} p_E$) for HARPS, for various values of α, n, r for $L = 64$. The linear relationship implies that for all plots, changing the value of k to ψk would imply scaling the y-axis ($\log_{10} p_E$) by the same factor ψ.

5 Conclusions

This paper compared two key predistribution schemes employing only symmetric crypto primitives - RPS and LM - to a novel key predistribution scheme, HARPS, which is a generalization (and improvement) of both schemes. All three schemes rely on some degree of tamper resistance of the nodes to ensure that an attacker cannot "sniff" the embedded keys easily. A practical implementation of such schemes would therefore rely on some mechanism of periodic renewal of keys [1]. For instance, renewal of keys may be performed by interacting with the TA. For this purpose the session keys for the node-TA interactions, could be derived from *all* the k keys in the node's key ring. In addition, to provide forward secrecy, an additional key may be required. This can be a very highly protected "update" key shared between the TA and each node, or a "password" needed for node-TA interactions. The periodicity of the renewal should probably depend on the expected threat level. Even though, in analysis we assume that if one node is compromised, all keys buried in the node are compromised, in practice the nodes may be engineered in such a way that it would be extremely difficult to "sniff" *all* keys buried in a node. In other words, to expose k keys the attacker may have to compromise *many nodes*.

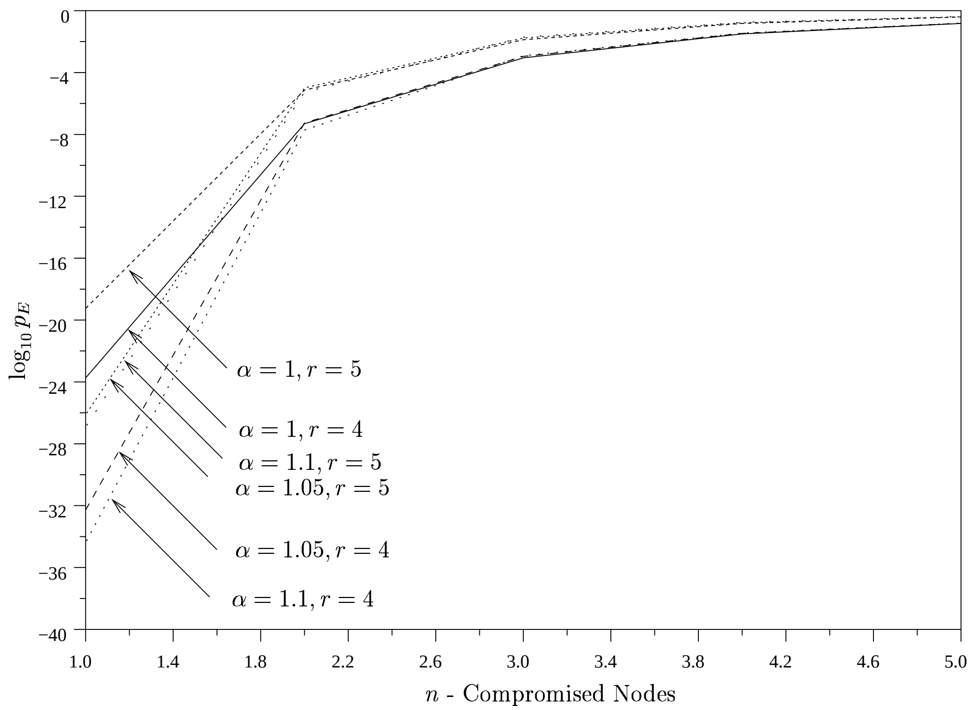


Figure 5: Probability of Eavesdropping for HARPS for $r = 4, 5$, $n = 1 \cdots 5$. $\alpha = 1$ corresponds to LM.

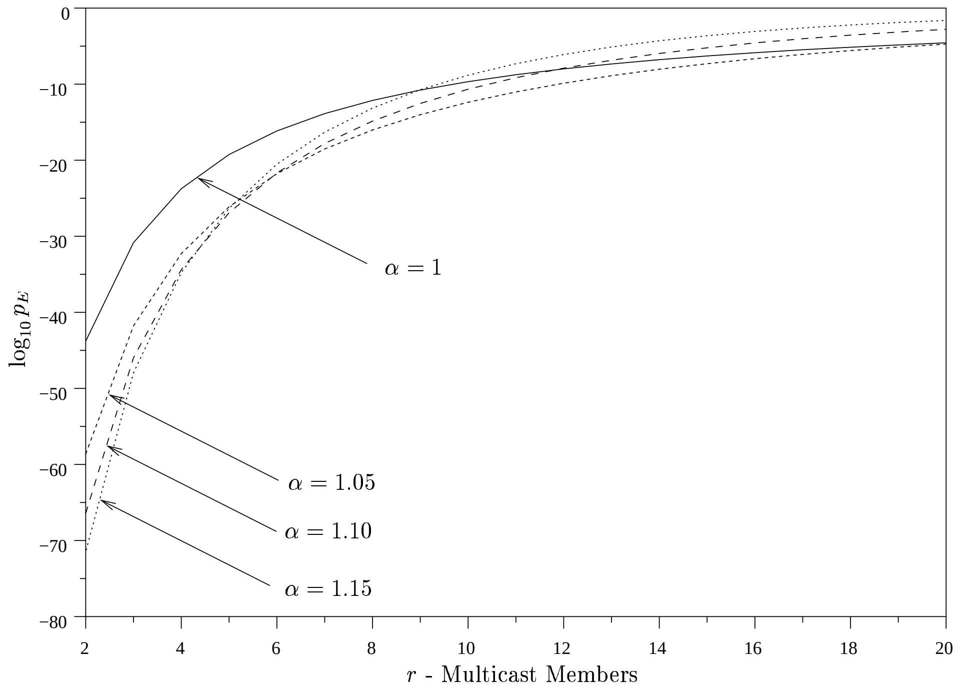


Figure 6: Probability of Eavesdropping for HARPS for $n = 1$ and $r = 2 \cdots 20$. $\alpha = 1$ corresponds to LM.

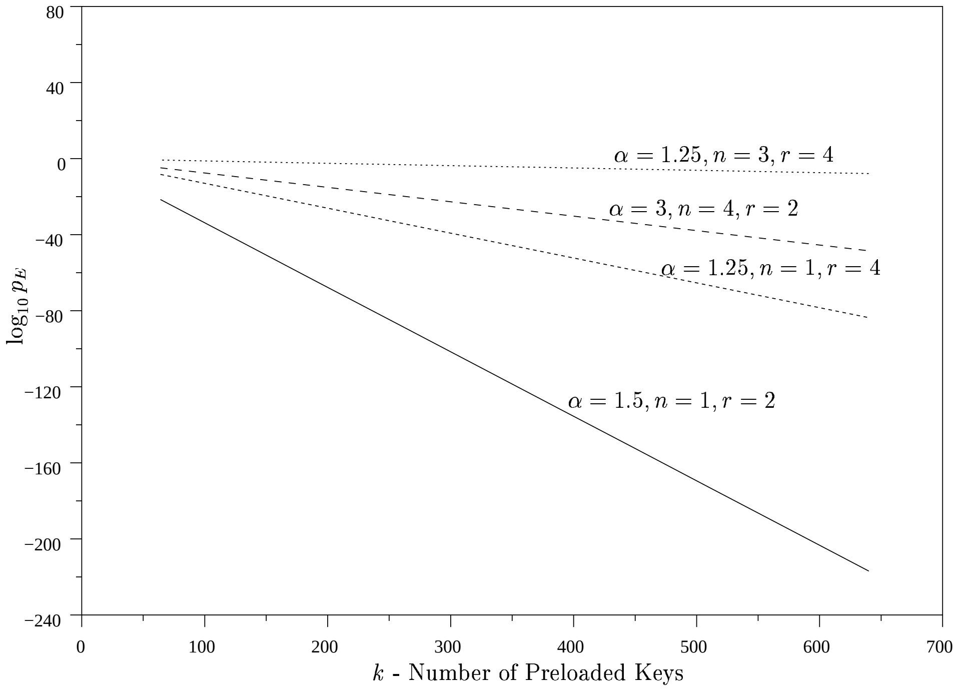


Figure 7: Linear relationship between k and $\log_{10} p_E$ in HARPS for various values of α, n, r.

The main advantage of HARPS over LM is the flexibility provided to choose the parameters of the system depending on the expected threat level and the nature of the deployment. For example, if the threat of compromising nodes is high, and the periodicity of renewal relatively low, then a higher value of $\alpha = \frac{P}{k}$ should be chosen. For all schemes, the security offered can be increased exponentially by increasing the number of preloaded keys k in each node.

The merit of HARPS was found to be considerably higher than that of RPS or LM. This implies that for the same level of security, HARPS can be implemented with less resources (in terms of number of preloaded keys). For example, if a probability of eavesdropping of the order of 10^{-20} is deemed satisfactory, then for a scenario of 5 compromised nodes ($n = 5$), for unicast communications ($r = 2$), LM needs $k = 960$ and RPS needs $k = 620$. The same security can be achieved by using HARPS with $k = 272$. Similarly for the case of $n = 10$ and $n = 20$ respectively, the required values of k are (3200, 1250, 528) and (12288, 2400, 1024) respectively (for LM, RPS and HARPS respectively).

References

[1] M. Ramkumar, N. Memon, R. Simha,"Pre-Loaded Key Based Multicast and Broadcast Authentication in Mobile Ad-Hoc Networks," to appear in 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] T. Matsumoto, M.E.Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, **IT-22**(6), Dec. 1976, pp.644-654.

[5] 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.

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

[7] H. Chan, A. Perrig, D. Song, “Random Key Predistribution Schemes for Sensor Networks,” IEEE Symposium on Security and Privacy, Berkeley, California, May 2003.