PRELOADED KEY DISTRIBUTION SCHEMES FOR AD HOC NETWORKS

Mahalingam Ramkumar

Dept. of Computer Science and Engineering
Mississippi State University
Mississippi State, MS 39762

Nasir Memon

Dept. of Computer and Information Science
Polytechnic University
Brooklyn, NY 11201

ABSTRACT

We investigate the applicability of key pre-distribution schemes for securing ad hoc networks. While most key pre-distribution (KPD) schemes satisfy the fundamental requirements to serve as an enabler for ad hoc network security, there are some additional desired properties which would significantly enhance their appeal for practical deployments. This paper addresses many such desired properties like computational and storage efficiency, scalability, effect of partial guarantees of tamper resistance, key renewal, flexibility for post deployment extensions, and hierarchical deployments, *especially in the context of wireless ad hoc networks*. It is shown that HARPS, one such KPD scheme possesses many of these desirable properties.

1. INTRODUCTION

In many evolving applications like ad hoc networks of sensors or mobile nodes, involving autonomous and typically resource constrained devices, it is imperative to have an efficient means of developing trust between 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 distribution scheme (KDS). The motivation of this paper is three-fold:

1. identify issues involved in deployment of a suitable KDS for ad hoc networks;
2. compare the ability of existing KDSs to address the issues, and
3. provide a practical KDS for this purpose;

Any KDS, should satisfy the following fundamental properties in order to be useful in such applications - (1) low complexity, (2) scalability, and (3) ability to operate without a trusted authority (TA). The first requirement rules out KDSs using asymmetric cryptographic techniques. The second rules out the "basic" key distribution scheme, where $\binom{N}{2}$ keys are distributed among N nodes (and each node gets $N-1$ keys). The third rules out KDSs like Kerberos. This leaves us with key pre-distribution schemes.

Since Blom et al. [1] realized that it is possible to perform trade-offs between complexity and security, various KPD schemes have been proposed in literature. A KPD scheme consists of a TA and N nodes with unique IDs. The TA chooses P secrets. Each node is preloaded with k secrets (typically $k \leq P$). The preloaded secrets or the "key-ring" of each node is typically a function of the TA's P secrets and the ID of the node.

Key pre-distribution schemes are essentially trade-offs between security and complexity. Their reduced complexity permits even severely resource constrained devices to participate in the deployment. However, their limitation in security calls for a need to *control sizes of attacker coalitions*, perhaps by providing some assurance of *tamper-resistance* of the devices with preloaded secrets.

While the need for tamper resistance may seem an unreasonable assumption at first sight, it should be realized that the need for *autonomous* operation of the devices, implies that dependency on tamper resistance is *not optional*. Deployments based on PKI for instance, would still need mechanisms to protect the preloaded private keys. After all, every device that is deployed is expected to operate *without human intervention*. Even though many devices may have a "human controller" at hand, it is not practical for the person to store the key just in his / her head and supply it to the device when needed (for each instance of communication)! This realization, is already driving technology to improve tamper-resistance of devices. Nevertheless, it may not be wise to assume infallible tamper resistance. An attacker, with unlimited time and resources, may be able to circumvent any protection offered by tamper resistance.

Though almost all KPD schemes satisfy the 3 fundamental properties, it is also desirable for a KPD schemes, to possess the following additional properties for securing wireless ad hoc networks:

1. Obtaining session keys from preloaded secrets should not be computationally expensive.
2. The failure (or compromise of security) of the system should not occur catastrophically.
3. The number of preloaded keys (or the key-ring size k) in each node, should not be very high.
4. The KDS should be easily extensible to multicast communication scenarios.
5. The KDS should be renewable, to provide a limited time window for attackers to carry out attacks.
6. The KDS should be able to efficiently utilize the security provided by "partial" tamper resistance.
7. The KDS should allow for hierarchical deployments.

Some of the above requirements are obvious. The rest, we shall try to justify in this paper. Apart from a brief discussion of various KPDSs, in Section 2, KPD schemes are classified into two categories - deterministic and probabilistic. Notions of catastrophic and graceful failures are discussed. In Section 3 we outline and compare different properties of KPD schemes and address their suitability for the intended application. In Section 4 we conclude that HARPS [2] is very well suited for this securing ad hoc networks.

2. OVERVIEW OF KEY PRE-DISTRIBUTION SCHEMES

A (r, n, p)-KPD scheme (or an r-conference, n-secure KPD scheme) is a systematic method of allocation of k secrets to each of the N nodes in a network, in such a way that

1. any group of r nodes $\{g_1 \cdots g_r\} = \mathcal{G}$ can discover the shared secret $K_{\mathcal{G}}$ with probability p_d,
2. a coalition of n nodes $a_1 \cdots a_n \notin \mathcal{G}$ can discover the same secret $K_{\mathcal{G}}$ with probability p.

The probability p is referred to as the "eavesdropping" probability, and the probability $p_o = 1 - p_d$ as the "outage" probability. For a KPD scheme to be "secure" even when an attacker has exposed secrets from n nodes, p_o and p should be "very close" to 0. Obviously, if the group of r nodes cannot discover a shared secret (which happens with a probability $p_o = 1 - p_d$) 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. More specifically, $p = p_o$ for $n = 0$, and $p \geq p_o$ for $n > 0$. KPD schemes can be broadly classified into two broad categories. For *deterministic* KPD schemes, the eavesdropping probability p takes only binary values - 0 or 1. On the other hand, for *random* KPD schemes, p can assume continuous values.

In [4], Matsumoto et.al. presented a generalized model of deterministic KPD schemes, consisting of a TA and a collection of N nodes with unique IDs. The TA employs a r-symmetric function f (the coefficients of which are the system secrets, chosen by the TA). Each node employs a $r-1$ symmetric function, (say g_i for node with ID A_i). The functions satisfy the relationship

$$g_i(x_1, \ldots, x_{r-1}) = f(A_i, x_1, \ldots x_{r-1}) \quad (1)$$

The "symmetry" of the functions g and f manifest themselves as invariance to any permutation of the variables. For each node (say with ID A_i), the TA evaluates the function f by substituting the ID of that node, resulting in an expression in $r-1$ variables. The expression in $r-1$ variables is the the function g_i for that node. The coefficients of the function $g_i()$ are the *secrets* preloaded in the node with ID A_i. The group secret is obtained independently by each node. Node i, for instance obtains the secret by substituting the IDs of the other $r-1$ nodes in the group and evaluating $g_i()$. For instance, for two nodes A_i and A_j (or $r = 2$), the shared secret K_{ij} is obtained as $K_{ij} = g_i(A_j) = g_j(A_i) = f(A_i, A_j) = f(A_j, A_i)$.

2.1. Deterministic Schemes

In Blom's scheme [1], for $r = 2$ for n colluders, the function $f()$ is a symmetric n-degree polynomial in two variables in a prime field. The secrets provided to node A_i are the $(n+1)$ coefficients of a polynomial in one $(r-1)$ variable. The extension of Blom's scheme to multicast scenarios ($r > 2$) was proposed by Blundo et. al. [3], where $f()$ is a symmetric polynomial of degree n in r variables. In this case, each node needs $\binom{n+r-1}{r-1}$ secrets corresponding to the coefficients of symmetric polynomial of degree n in $r-1$ variables. Matsumoto et. al. [4] presented a linear symmetric scheme as a specific example of their generalized model Eq(1).

The major disadvantages of such KPD schemes is their *catastrophic onset* of failure, and their increased complexity due to the need for finite field arithmetic. Since then, many KPD schemes [5] - [8] based on *subset intersections* have been proposed which solve the problems of catastrophic failure and computational complexity. However, they introduce another disadvantage. The need to eliminate catastrophic failure results in a dependence of k on the network size N, which severely restricts the scalability of such schemes.

For any deterministic scheme designed for resistance to collusion of n nodes, if the number of colluders is $n_1 \leq n$, the probability of eavesdropping, $p = 0$; for $n_1 > n$ however, $p = 1$. Thus the failure of such systems occur *catastrophically*.

2.2. Probabilistic Schemes

More recently [2], [10]-[15], random KPD schemes have attracted the attention of many researchers. Unlike the categories of KPD schemes described above, which guarantee that the system is secure until a certain number (n) of nodes have been compromised, random KPD schemes typically offer a "probabilistic guarantee."

In the Leighton-Micali (LM) scheme [9] the TA chooses k "root" secrets $[M_1 \cdots M_k]$, and a cryptographically strong hash function $h(.)$. A one-way function (or a random number generator) $F_L()$ is employed to generate a stream of k uniformly distributed random numbers, $F_L(ID_A) = a_1 \cdots a_k$, $1 \leq a_i \leq L$, seeded by node ID. The preloaded keys in node A are $[M_1^{a_1} \cdots M_k^{a_k}]$, where M_i^j is obtained by repeatedly hashing M_i j times. The shared key K_{AB} between nodes A and B is then $K_{AB} = h(K_1 \cdots K_k)$, where $K_i = M_i^{\alpha_i}$, and $\alpha_i = \max(a_i, b_i)$. For the LM scheme the r-symmetric function is the evaluation of the *maximum* of the "hash-depths" for r nodes for each root key.

In the random preloaded subset (RPS) key distribution [15] the TA chooses an indexed set of P secrets $K_1 \cdots K_P]$. A one-way function (again seeded by the node ID) is used to obtain a *partial*[1] random permutation sequence $[A_1 \cdots A_k] = F_R(ID_A)$. Now $[A_1 \cdots A_k]$ are the indexes of the keys preloaded in node A. By exchanging IDs, two nodes can immediately determine the shared indexes, and use all shared keys to derive pair key.

Hashed random preloaded subset key distribution (HARPS) [2] is a generalization of RPS [15] and the LM [9] schemes. HARPS is defined by 3 parameters (P, k, L), and two public functions - $h()$, a cryptographic hash (one-way) function and $F_()$, a public key generation function. The TA generates P root keys $[M_1 \cdots M_P]$. The one way function $F_()$ (seeded by ID) is used to generate an *ordered pair* of length k. The first coordinate of the ordered pair is a partial random permutation sequence (like in RPS), and the second coordinate is the hash depth (uniformly distributed between 1 and L, like LM scheme). The first coordinate of each ordered pair determines the k root keys to be chosen. The second coordinate determines the number of times the corresponding root key should be hashed. RPS is a special case of HARPS where $L = 0$. Also the Leighton Micali scheme is a special case of HARPS when $P = k$. The security of latter three schemes (LM, RPS and HARPS) Unlike the first 3 KPDSs (Blom, Matsumoto and matrix), the security of LM, RPS and HARPS is governed by a non-zero[2] eavesdropping probability p.

At fist sight, permitting a finite eavesdropping probability may seem like a serious disadvantage. In practice, it is not. Even for a (deterministic) KPD scheme for which $p = 0$ for some n, the final shared secret is a usually a "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

[1] Say the first k numbers of a random permutation of $1 \cdots P, k < P$

[2] By careful choice of parameter k the probability p can be made vanishingly small

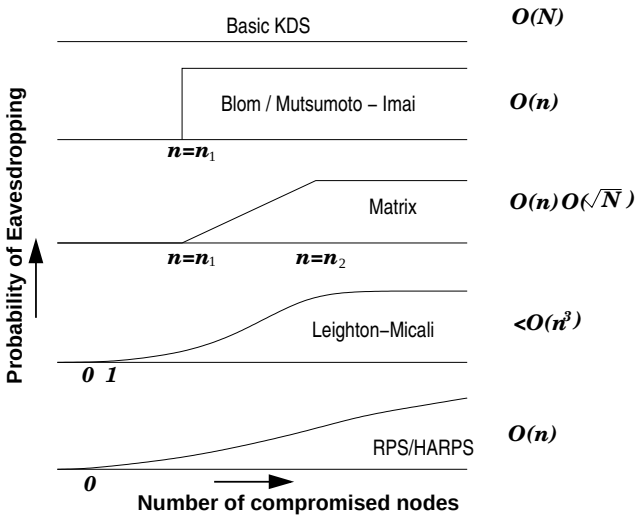

Figure 1: A **qualitative** representation of the progression of KPD schemes with relaxation of security constraints. N is the network size and n represents the size of the attacker's coalition.

secret out of a hat" (without the need to compromise any node). Thus permitting a small eavesdropping probability $p > 0$ 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).

Thus 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. Figure 1 depicts the *qualitative* progression of KPDs as a function of the probability of eavesdropping and the number of compromised nodes, under varying security assumptions.

3. PROPERTIES OF KPD SCHEMES

3.1. Network Size

Typically, the maximum network size that can be supported is only limited by the number of bits assigned for the node ID (every node needs a unique ID). For Blom's scheme [1], it is limited by the size of the prime field q in which the polynomial arithmetic is performed. For the $(r, n, p \neq 0)$ schemes (HARPS, LM, RPS), the limitation is the number of unique key-rings that can be obtained. If there are Q possible key-rings, then taking "birthday-paradox" into account, the maximum network size is also limited by $N_{max} = \sqrt{Q}$. As an example, for HARPS with $P = 2560, k = 256$ and $L = 64$, $N_{max} \approx \sqrt{\binom{P}{k} L^k} \approx 1.3 \times 10^{411} > 2^{1364}$. For RPS with $P = 2560, k = 256$, $N_{max} \approx \sqrt{\binom{P}{k}} \approx 8.4 \times 10^{179}$, and for the LM scheme (with $k = 256, L = 64$), $N_{max} \approx \sqrt{\binom{P}{k}} \approx 1.5 \times 10^{231}$. It is therefore safe to assume that the limiting factor for the network size would indeed be the number of bits assigned to represent the ID!

3.2. Efficiency

Both [1] and [4] need expensive finite field arithmetic to calculate group (or pair) secrets. For both schemes, $k \propto n$. Specifically, for $r = 2$, $k = n + 1$ for [1] and $k = hn$ for [4], where h is a "security factor". On the other hand, for the subset intersection schemes, $k \propto n\sqrt{N}$ (for [5]) to $k \propto n \log N$ (for other schemes). The dependence of k on the network size N severely restricts the scalability of the subset intersection schemes.

Table 1: Number of preloaded keys needed to maintain a probability of eavesdropping below 10^{-20}. For HARPS and LM $L = 64$.

n	LM	RPS	HARPS
5	960	620	431
10	3200	1250	825
20	12288	2400	1609

For the LM scheme [9], $k \approx O(n^2) - O(n^3)$. However for RPS and HARPS $k \propto n$, similar to the schemes that need expensive finite field arithmetic. In addition, HARPS has significantly lower probability of eavesdropping when compared to RPS for the same values of r, n, k.

One of the main advantages of the probabilistic schemes is the more graceful degradation of performance as more and more secrets are compromised. For all three schemes the probability of eavesdropping is exponentially related to k, the number of preloaded keys. Thus if k is doubled, the probability of eavesdropping gets squared.

The performance of RPS and HARPS depends on an "operating point," characterized by $\alpha = \frac{P}{k}$ [15], [2], the ratio of the total number of keys in the pool to the number of preloaded keys. In general, greater the "design" value of n, larger should be the value of α. Table 1 depicts the relationship between the number of preloaded keys, k and the number of compromised nodes, n, for a probability of eavesdropping less than 10^{-20}, obtained by numerical evaluation of the eavesdropping probabilities based on analytically derived formulas in [15] and [2]. Note that HARPS is able to achieve the performance target in terms of probability of eavesdropping with the least number of preloaded keys. Also note the almost linear dependence of k with n for HARPS and RPS. The results in the tables assume optimal choice of $\alpha = \frac{P}{k}$ for RPS and HARPS. While for LM, the outage probability $p_o = 1 - p_d = 0$, RPS and HARPS permit a small outage probability $p_o = (1 - p_d) > 0$ (which implies that $p = p_o > 0$ even for $n = 0$). It is primarily this freedom of permitting an outage probability that gives RPS and HARPS the ability to achieve $k = O(n)$. It is perhaps worth re-iterating that the eavesdropping probability *includes* the outage probability. The outage probability is the probability of eavesdropping when $n = 0$. For example, for a case where $k = 1500, P = 15000$, the outage probability, is about 5.5×10^{-62}. As another example, for the case where $k = 190, P = 342$, the probability of outage is about 1.5×10^{-55}. For a network consisting of say 100 billion nodes, there are 5×10^{20} possible interactions (each of the 100 billion nodes communicating with every other node - or 5×10^{20} possible pairs). With the latter case ($P = 342, k = 190$), the probability of finding a pair that does not share any key (or the probability of outage) is a minuscule 10^{-35} - that is about the same as the probability that some one can "guess" a 118-bit key in one try!

The major disadvantages of the deterministic schemes are the catastrophic failure of the system when the number of colluders increase beyond the "design" value, and the use of expensive finite field arithmetic. The probabilistic schemes on the other hand, apart from averting catastrophic failures, also eliminate the need expensive finite field arithmetic. Even though the possibility of

"outage" (in probabilistic schemes) may seem to be a serious disadvantage, the probability of such an occurrence can be made vanishingly small.

3.3. Tamper Resistance and Key Renewal

The security provided by any KPD scheme can be compromised by exposing secrets buried in nodes. The phrase "compromising a KPD scheme," may have different meanings, depending on the motivation of the attacker. An attacker with access to some exposed secrets $\mathbb{S}_A$, may be able to "masquerade" as some node i, for the purposes of his interactions with node j. He achieves this by "discovering" the shared secret K_{ij} between the two nodes (by employing his "knowledge" $\mathbb{S}_A$ - the attacker also simultaneously gains the ability to convince node i that he is node j). Some possible motivations (by no means an exhaustive list) then, of an attacker, would be to determine K_{ij} for the following cases

A1 a specific i, j;

A2 a specific i, when j is the TA;

A3 for all i when j is the TA.

Deterrence of the attacker from exposing secrets calls for some assurance of tamper-resistance of devices. Obviously, if tamper-resistance is perfect, KPD schemes are rendered secure. In practice, any form of tamper resistance can perhaps be broken by a motivated attacker with unlimited time and resources. However, it may be be reasonable to expect tamper resistance to provide some limited extent of guarantees.

A possible model for limited extent of assurances provided by tamper-resistance, is that tamper-resistance ensures that only a *fraction* of the secrets can be exposed by tampering with any node. The existence of this guarantee, affects different KPD schemes in different ways.

Consider a n-secure Blom's KPD, where $n = 20$. If the tamper-resistance property guarantees that only 10% of the keys buried in each node can be compromised, then an attacker needs to tamper with more than $10n = 200$ nodes to engineer a successful attack. On the other hand for a KPD based on subset intersection, with comparable complexity, the attacker may need to tamper with only 50 nodes for accomplishing attack A1 but probably 10000 nodes for accomplishing attack A3. For a random KPD, (with comparable complexity), an attacker may have to tamper with 120 nodes to accomplish the attack A1 with a probability of 10^{-20}, and probably 500 nodes to accomplish A1 with a probability of 0.5, and say 20,000 nodes to accomplish attack A2 with a probability of 0.5, and perhaps 25,000 nodes to accomplish attack A3 with a probability of 0.5.

Accomplishment of attack A2 (the ability to "fool" the TA), implies successful "synthesis" of a node by an attacker. Increased resistance of KPD schemes to node synthesis (or attack A2) can be used advantageously by *periodic renewal of keys*. For renewal, each node would authenticate itself to the TA using *all* its preloaded secrets, and receive a set of new keys. After key updates, the efforts of an attacker to gather secrets that made it possible for him to perform attack A1, are rendered useless. Obviously, KPD schemes based on finite field arithmetic cannot efficiently utilize the strength provided by a combination of limited tamper resistance and periodic renewal of keys.

While random KPD schemes have the advantage of much higher resistance to node synthesis compared to deterministic KPDs, HARPS in particular performs significantly better than other random KPD schemes in this respect. In [17] we have shown that under reasonable assumptions, an attacker may need to tamper with a few hundred thousand nodes to compromise HARPS. In this respect HARPS is more than an order of magnitude better than RPS or LM.

Another issue, that crops up when keys are periodically renewed, is the feasibility of communication between a node with updated keys and a node with not all keys current[3]. To facilitate this, each update could replace only a fraction of the keys, which is possible in all random KPD schemes and subset intersection schemes. Another possibility (in LM and HARPS) is that the updated secrets could be pre-images of old secrets, under a cryptographically strong one-way function. This would however, need a long one way hash chain [18], [19] of the secrets to be created before deployment.

3.4. Physical Layer Security

A major concern in security of wireless devices is their susceptibility to jamming. Susceptibility to jamming can be drastically reduced if nodes participating in message exchanges share a key, which can be used for CDMA or frequency hopping (FH). For nodes equipped with some form of KPD scheme, the nodes need to know only the respective IDs before they can establish a key, very little exchange needs to be done in an "open" channel (the IDs have to be exchanged in an open channel before the shared secret can be established). In practice, nodes could "tune" into an open channel for a very small fraction of time for the purpose of "welcoming" new neighbors. Thus KPD schemes are also ideally suited for securing the lower layers. However, for the "efficient" KPDs based on subset intersection schemes (for which $k \propto n \log N$) exchange of IDs is not enough to establish shared secrets. Their high efficiency (compared to the matrix scheme [5]) is a result of complex deterministic constructions for the purpose of allocation of keys to each node. Thus to determine the shared keys the nodes need to execute the complex construction algorithm, which may not be feasible. Therefore the other option is to exchange long messages explicitly specifying the indexes of the keys they have.

3.5. Extension to Multicast

Though multicast communication between r nodes can be achieved by $r - 1$ unicast transmissions, multicasting has two obvious advantages - efficient bandwidth usage, and authentication of multicast[4].

Although in theory all schemes are readily extensible to secure multicast or conference communications, not all *deployments* are. For instance a deployment of Blom's scheme for a specific r, n, cannot be used[5] for $r_1 > r$. On the other hand, the probabilistic schemes like LM, RPS and HARPS can be extended to operate in a multicast scenario (without changing any of the system secrets or the deployed secrets) albeit with a reduced margin of security (higher p). The reduced security of multicast group secrets however, is not a major disadvantage if the multicast secrets are used just for securing the physical layer. For the deterministic schemes, the number of preloaded secrets increase drastically with r - for

[3] This situation may arise if a node has not had the opportunity to access the TA for updates

[4] in some cases it may be necessary for *each* of the r nodes to know that *all* the r nodes received the message

[5] This would involve changing the r symmetric polynomial to a r_1 symmetric polynomial, $r_1 > r$, and therefore changing of all preloaded secrets

Table 2: Comparison of KPD Schemes.

Property	Blom	SI Schemes	LM	RPS	HARPS
N - Network Size	prime field size	$\sqrt{\binom{P}{k}}$	$\sqrt{L^k}$	$\sqrt{\binom{P}{k}}$	$\sqrt{\binom{P}{k}L^k}$
k - key ring size	$O(n)$	$O(n\sqrt{N}) - O(n \log N)$	$\approx O(n^3)$	$O(n)$	$O(n)$
Failure Mode	Catastrophic	Graceful	Graceful	More Graceful	Most Graceful
Eavesdropping Probability	0/1	0 / 0-1	non-zero	non-zero	non-zero
Outage Probability	No	No	No	Yes	Yes
Computational Complexity	High	Low	Low	Low	Low
Extension to Hierarchical Deployment	Complex	Simple	Simple	Simple	Simple
Hierarchical Deployment mode	Tree	Tree	Vertical	Tree	Tree
Protection of Levels	Yes	No	Yes	No	Yes
Post Deployment Extensions	Not possible	Not possible	Possible	Possible	Possible
Extension to Broadcast Authentication	Not possible	Yes	Limited	Yes	Yes
System Renewal	No	Yes	Yes	Yes	Yes
Seamlessness of Key Renewal	No	Limited	Yes	Limited	Yes
Resistance to Node Synthesis	Low	High	High	Higher	Highest

example, $k = \binom{n+1}{r-1}$ for Blom's scheme. As no trade-off of security (p) vs k is possible (as p can assume only values 0 / 1), for practical deployments $r > 5$ may be infeasible. However, for the probabilistic schemes r could be made larger by sacrificing some security (increasing p).

3.6. Broadcast / Multicast Authentication

Most broadcast authentication techniques based on symmetric key cryptography use some form of key pre-distribution [16]. Even though some broadcast authentication techniques are based on finite field arithmetic, the preloaded keys for KPD schemes based on finite field arithmetic cannot be used for achieving broadcast authentication. However for subset intersection scheme, RPS and HARPS, the same keys used for establishing shared secrets can also be used for broadcast authentication. This can be achieved by transmission of key based message authentication codes corresponding to each preloaded key.

3.7. Hierarchical Deployments

The possibility of hierarchical deployments is a significant advantage for practical deployments, both in terms of ease of administration of the system, and localizing security breaches for damage control. While Blom's scheme can be deployed in a hierarchical fashion, it would be at the expense of substantial increase in complexity. For example, for a 2 level hierarchy of domains and users the system polynomial would be four variable instead of two variables (for $r = 2$).

The LM scheme offers a simple vertical hierarchy with no additional complexity. The vertical hierarchy is made feasible [9] by selecting non overlapping intervals of the hash depth for different levels. For example, for a m level hierarchy with a hash depth of L for every level, the hash depth of the m different levels would be $1 : L, L + 1 : 2L, \ldots, (m - 1)L + 1 : mL$. The highest (or most trusted) level would have the least hash depths. Even if keys at a lower level at completely compromised, it will not affect the security of the higher levels (lower hash depths).

RPS and HARPS can offer a richer tree structured hierarchy. This can be achieved as follows (Figure 1). At the top most level is the TA with P_0 keys. Nodes at the second level have P_1 keys each, where $P_1 \leq P_0$, picked from the P_0 keys of the TA. Each parent could act as a TA of its child nodes. The nodes at level 3 would likewise have P_2 keys, which are picked from the pool of P_1 keys belonging to *their* parent. HARPS provides an even better separation of the lower levels from parents, as we can choose different ranges of hash depths for each level, like the LM scheme. So child nodes pick a subset of the root keys of the parent, and have higher hash depths. Nodes in the same level will have the same hash depth range.

Figure 2 depicts the tree-hierarchy possible in RPS / HARPS In such a scheme, the parent node is capable of eavesdropping on all conversations *involving* its children. This implies that node A in Figure 2 can eavesdrop on a conversation between a and b or between a and c. Both nodes A and D would be able to eavesdrop on communications between a and c. However, A can not eavesdrop on communications between c and d. For hierarchical deployments of both RPS and HARPS, it is possible to either limit communication to only between siblings ($a - b$, $c - d$), or allow communication between arbitrary nodes irrespective of their position in the tree hierarchy - for example "cousins" ($a - c$, $b - d$), or even across levels and direct ancestors ($1 - c$, $A - 4$). For interaction between siblings, the siblings need not even know the ID of the parent. But for non-sibling interactions, the complete hierarchical ID of both should be known to the interacting nodes. For example, for a communication to be possible between nodes 1 and d, the address (or hierarchical ID) of node 1 is $A - a - 1$, and that of node d is $D - d$. Without the complete ID, the key intersections (for RPS and HARPS) and hash depths (for HARPS) cannot be calculated. However, communication between child nodes of different parents is bound to be less secure than communications within children of the same parent[6].

4. CONCLUSIONS

Key pre-distribution schemes enables simple and effective means of building trust relationships. The single most disadvantage of key pre-distribution schemes is their vulnerability to the ability of an adversary to tamper with and expose hidden secrets from many nodes. However a combination of limited extent of tamper resistance and periodic renewal of keys can render KPD schemes reasonably secure.

[6]Siblings are bound to share more keys than cousins

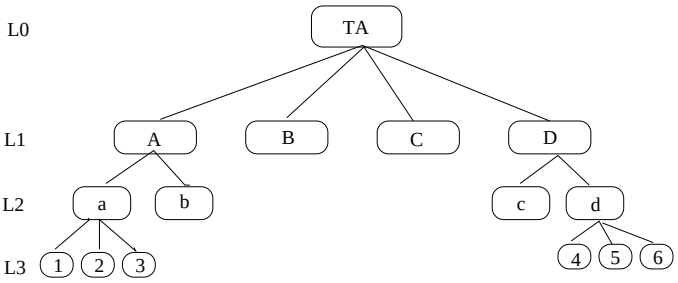

Figure 2: Tree hierarchy in RPS and HARPS.

The paper addressed many desirable properties that a key distribution system should possess to aid practical deployments, like hierarchical deployments, ease of post deployment extensions to the system, key renewal, extensions to multicast and broadcast authentication, and complexity. The many proposed methods were classified into two categories - deterministic methods like [1], [4] and [3], and methods with probabilistic figures of merit like LM [9], RPS [15] and HARPS [2]. It was seen that the latter methods offer more flexibility for deployment. Among the probabilistic methods, HARPS, [2], a generalization of RPS and LM has significantly better performance, and deployment flexibility.

Apart from being highly efficient, HARPS also has the following very desirable properties:

1. Employs only symmetric cryptographic primitives like hash functions.
2. Nodes need to exchange only their IDs in order to discover the shared secret.
3. No practical restrictions on the network size N.
4. Slower degradation of performance of a system for larger values of n - the size of attacker coalition.
5. Offers a high degree of resistance to "node synthesis."
6. Provides a "seamless" mechanism for key updates.
7. Offers a highly desirable tree-hierarchical deployment with very little or no increase in complexity. Further, it guarantees that secrets exposed in lower levels of the hierarchy have no impact on the security of the higher levels.
8. Offers seamless extensions to multicast secrets and multicast / broadcast authentication

Thus the nice efficiency together with the rich deployment flexibility it offers should make HARPS a very desirable key predistribution scheme for securing ad hoc networks. A comparison of various KPDSs are tabulated for quick reference in Table 2.

5. REFERENCES

[1] 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.

[2] M.Ramkumar, N. Memon, "An Efficient Key Predistribution Scheme for MANET Security," submitted to the IEEE Journal on Selected Areas of Communication.

[3] 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.

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

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

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

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

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

[10] 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.

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

[12] 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.

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

[14] 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.

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

[16] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas, "Multicast Security: A Taxonomy and Some Efficient Constructions," IEEE Infocom, Long Beach, CA, USA, Oct. 2001.

[17] M. Ramkumar, N. Memon, "On the Security of Random Key Pre-distribution Schemes," 5th Annual IEEE Information Workshop, West Point, NY, June 2004.

[18] L. Lamport, "Password Authentication with Insecure Communication," Communications of the ACM, 24(11):770-772, November 1981.

[19] A. Perrig, R. Canetti, D. Song, D. Tygar, "Efficient and Secure Source Authentication for Multicast," in Network and Distributed System Security Symposium, NDSS '01, Feb. 2001.