Stealthy pre-attacks against random key pre-distribution security

Panagiotis Papadimitratos
School of Electrical Engineering
KTH, Royal Institute of Technology
Stockholm, Sweden
Email: papadim@kth.se

Jing Deng
Department of Computer Science
University of North Carolina at Greensboro
Greensboro, NC, USA
Email: jing.deng@uncg.edu

***Abstract*—Random key pre-distribution (RKPD) has been investigated for large wireless sensor networks, in order to achieve efficient security and robustness against limited node compromise. While it is possible that an adversary obtains a subset of the symmetric keys in use, it has been unclear how to use those to compromise specific secure links. We investigate how the adversary could do this practically. We term this the Stealthy Pre-Attack (SPA), because the adversarial nodes leverage benign behavior to guide their attack. The contribution of this paper is the identification of this adversarial behavior, the evaluation of its benefits for the attacker, which can then much more effectively compromise security, and the proposal of counter-measures to mitigate it.**

I. Introduction

There is a multitude of *random key pre-distribution (RKPD)* schemes in the literature: nodes equipped each with a set of symmetric keys, randomly chosen out of a pool, can communicate securely with those peers that they share at least one key with [1], [2], [3], [4]. The major benefits of RKPD are its simplicity and the efficiency of communication secured with symmetric keys. A major driving force has been wireless sensor networks [5], with their mid- to large-scale deployment and their resource-limited nodes. RKPD can also be useful in other distributed systems that require security but cannot practically use public key infrastructures; for example, mobile or on-line social networks, or embedded and Internet of Things systems.

To compromise the security of, e.g., the exchange of encrypted and authenticated messages between two benign nodes, the adversary should essentially have knowledge of the symmetric key(s) in use. The higher the number of keys compromised or disclosed to the adversary is, the more likely it is that secure communication *among benign and not compromised nodes* can be compromised.

However, with a large number of nodes and a large pool of symmetric keys, with few of them used by each node, it is far from trivial for the adversary to know which keys might be used by which pair of nodes. Essentially, the analysis of the vulnerability in the literature has abstracted away this aspect. From a practical point of view, however, an adversary would have to examine potentially any packet and try all keys it has on it. The likelihood to achieve this was calculated; but the effort the adversary needs to expend was not, thus the implication of a brute-force approach [6], [7], [8], [9], [10].

In this paper, we consider exactly a way for the adversary to guide such an attack and the use of its resources. This is especially important for the practicality of the attack and the validity of the threat. We term this the *Stealthy Pre-Attack, SPA*: The adversary can work its way over time in a stealthy manner, leveraging benign functionality, and identify in parts which keys are in use by which benign nodes. Afterwards, it can *focalize* its resources and actions (eavesdropping, interception of traffic, presence, equipment, etc) on those potential victims and try to compromise the security of their communication.

Achieving this is rather simple: The adversary has knowledge of a subset of all keys in use, e.g., by controlling some nodes. Then, it uses this knowledge to identify the benign nodes carrying these keys and, further, to compromise communication of those nodes only. Given such prior knowledge, it is clearly more likely to achieve that, compared to trying blindly against any secure communication link (note: not physical link but pairwise use of a symmetric key).

The identification of which keys are carried by which node comes "for free" for RKPD-based schemes. A "challenge-response" key-matching protocol is an integral part of all such schemes: Any two nodes can and need to run it in order to determine if they share at least one symmetric key. If they do, then they can use it to encrypt traffic, protect its integrity and authenticate each other, or use it to generate a session key to achieve these goals. For security reasons, the "challenge-response" key-matching protocol must, however, reveal only the common key(s) of the two participants and only to them; not any eavesdropper or any other network nodes. This is so for most of the schemes in the literature, but collecting the knowledge of which keys are available at each other node is, by default, meant to be possible.

As a result, after some time, the adversary will be able to single out a few benign nodes with a portion of their keys identified by the adversary. Focusing on those nodes, the adversary has higher probability of compromising their secure communication. Moreover, the adversary can mount its attack with less resources, intercepting or being in the vicinity of those nodes.

978-1-4577-2053-6/12/$31.00 ©2012 IEEE

The SPA attack, not previously considered by any of the RKPD schemes or their security analysis, is the main contribution of this paper. We analyze the basic form of the attack, outlined above, as well variants that can improve its effectiveness. Basically, all the adversary needs to do is compromise or "register" a number of nodes and interact actively with benign nodes. Having a number of nodes join the system, rather than compromising deployed nodes, as a means of eventual compromise has also been largely overlooked in the literature.

The SPA attack leverages benign behavior, beyond the compromise of nodes, and its mitigation can only be based on configuring the system in order to leave less space. As this depends on the exact use of the RKPD and the system operation, we outline a set of simple countermeasures that can reduce the effect of the attack. The effect of specific countermeasures is to be examined in follow-up work, taking into account specific applications. The contribution of this paper rests on the design of SPA attack, its evaluation, and the raising of awareness.

In the rest of the paper, we precise the system model (Sec. II) and define the adversary (Sec. III). We analyze the effect of the attack, both with analytic approximations and simulations, in Sec. IV. We then discuss countermeasures and their effectiveness (Sec. V), and conclude with a discussion of future work.

II. System Model

We consider a network with N nodes, with each node V_i equipped with a set, $\mathcal{K}_i$, of m symmetric keys. These keys are randomly drawn from a pool of N_p keys. Any V_i and V_j can communicate securely if there is an intersection of their $\mathcal{K}_i$ and $\mathcal{K}_j$: they use one of perhaps several common symmetric keys, namely some $K_{i,j}$. [1]

The basic system parameters m and N_p are chosen to regulate the probability of "security connectivity," that is, the likelihood that a pair of nodes, V_i and V_j, share at least one key ($K_{i,j} \neq \emptyset$). In principle, this probability should be sufficiently high, but the parameter values depend on the system, e.g., its scale, and the use of security. For example, for wireless sensor networks it may be desired that any two nodes share a key with probability 25%: in a deployment that results in 12 communication neighbors on the average, a node would be able to securely communicate with 3 of them on the average [1], [11]. We do not dwell on the exact use of the shared symmetric keys, that is, the security associations of any two nodes.

In order to establish secure communications, a node must first run the so-called key-matching protocol with any other node that it encounters (e.g., within range after deployment or encountered while moving, depending on which system uses RKPD). Therefore, legitimate nodes cannot in principle ignore seemingly legitimate key-matching requests; those are essential for the secure system operation.

[1]Without loss of generality, we assume throughout the paper that there is a single common key - unless noted otherwise.

The *key matching protocol (KMP)* is secretive, that is, nodes do not disclose the identity or other information on the keys that they hold and potentially share; certainly not the keys themselves. First, the initiator, say, V_i, generates m nonces and encrypts each of them with one of its keys, in $\mathcal{K}_i$. The result is sent to the target node, V_j, which tries to decrypt each of the encrypted nonces with keys in $\mathcal{K}_j$. If a matching key is found, the value of the nonce is disclosed. Then a response, e.g., nonce + 1, is encrypted and returned to the sender; for multiple common keys, all corresponding responses are sent back to V_i. Once the response is received, V_i knows the keys that it shares with V_j.[2]

The key pool server typically is an entity that randomly assigns m distinct keys from the N_p available, to any node requesting it. The exact type of operation depends on the system characteristics and requirements. We assume the aforementioned typical behavior and discuss alternatives later in the paper.

III. Adversary Model

The operation of RKPD schemes makes it trivial for any node to learn the subset of keys that it shares with any other node with the use of KMP. It only relies on its own knowledge ($\mathcal{K}_i$) and the outcome of KMP. With appropriately chosen parameters, m and N_p, each single node is unlikely to share more than a few keys with those nodes that it "KMP-ed."

However, this benign feature can be turned against the RKPD-based security. At first, an honest but curious node can run KMP aggressively, for example, at a pace exceeding that of other benign nodes. Nonetheless, there has been no typical threshold considered so far in the literature. Basically, curiosity is part of benign behavior for RKPD schemes. As a result, such an *honest but curious* node, V_{HC}, would be able to map, over time, many or potentially all of its keys, in $\mathcal{K}_{HC}$, to other nodes in the system.

While this is not a threat if any individual node acts in that way (as long as the system is configured correctly), the threat arises if more than one V_{HC} nodes "pool" together their keys and their discovery of identities of other nodes that carry and can use those keys. Thus, we define as adversary one that can orchestrate a set, $\mathcal{V}_{adv}$, of such V_{HC} nodes. The objective of the adversary is to use those nodes $V_{adv} \in \mathcal{V}_{adv}$ and to perform an attack: identify a few of the more vulnerable nodes in the system, among those that share a portion of own keys with the V_{adv} nodes.

The adversary can use the V_{adv} under its control to probe a set of victim nodes, $\mathcal{V}_{victim}$, by running the KMP multiple times. It can use K (a part of or even all) of its nodes, each running KMP with each of the nodes in V_{victim}. At the end, it would be certain to know the intersection of all the keys its V_{adv} carry with those held by nodes in $\mathcal{V}_{victim}$. This repeated use of the KMP, which in principle cannot be distinguished from benign operation and node behavior, is termed the *Stealth Pre-Attack (SPA)*.

[2]The challenge-response procedure should be performed both ways, but we use this form to simplify the discussion.

The SPA is simple to mount; especially in an open system where new nodes are expected to join or in a system with mobile nodes where new secure communication links need to be established over time. In that case, the adversary simply needs to get as large a set of own nodes registered in the RKPD system. The more numerous such nodes are, the more keys the adversary has and the more it can deduce about keys in the key-sets of benign nodes. If β of the m keys of a node in $\mathcal{V}_{victim}$ are thus mapped, the adversary would be able to compromise up to β of the secure links established by the victim. Clearly, if the adversary has $K = |\mathcal{V}_{adv}|$ nodes and at most Km keys, it cannot map more than Km keys to benign nodes. Thus, it can at most compromise that many secure channels, or less, as this would depend on how many of these keys are used for benign node communication.

Alternatively, the adversary could compromise keys held by benign nodes and use those as well. In that case, we would consider those nodes (their key-sets) as part of the $\mathcal{V}_{adv}$. What, however, is of interest is not the compromised nodes in $\mathcal{V}_{adv}$, but rather the adversary's ability to compromise the secure communications of some targeted victims singled out through the SPA attack. Therefore, the overall objective of the SPA attack is to identify those nodes who share at least a β portion of their keys with nodes in $\mathcal{V}_{adv}$. Then, the adversary can attack secure links established by the victims using those keys.

Finally, we consider a variant of the adversary that seeks to maximize the number of keys it has. As RKPD implies that any two nodes may have some common keys, the introduction or the capturing of one more node does not imply m new keys for the adversary. To achieve that, the adversary could persistently bootstrap each new node until it gets a totally new or sufficiently disjoint set of keys [12]. We term this the *persistent key request* attack; this is partly a misnomer, as it is just a way to facilitate later the SPA attack.

IV. Attack Analysis and Evaluation

In this section, we approximate analytically the cost to match keys, the cost to request distinct keys from the key pool, and the chance of success of the focalized attack. Then, we present our performance evaluation through MATLAB.

A. Cost of Matching Keys

The analysis is based on partially overlapping key sets held by the attacker nodes. This is the default case, as all nodes in the RKPD-based system share one or more keys with a non-negligible probability. As a result, when the adversary either compromised some of the deployed nodes or it has its own nodes in the system, it would have collectively a set of non-disjoint key sets.

We derive first the cost, or time, to identify βm keys of one benign node among N nodes and we denote the cost as $T(N,1)$. Focusing on one of the N nodes, the cost essentially becomes the expected number of times to randomly choose m keys from the key pool and match βm of the keys held by the target. After choosing the first set of m keys from the key pool, or in other words, after having the first adversarial node running the KMP with the targeted node, the adversary should have identified δ_1 common keys:

$$\delta_1 = \sum_{i=0}^{m} i \Pr(\text{sharing } i \text{ keys}) = \sum_{i=0}^{m} i \frac{\binom{m}{i}\binom{N_p-m}{m-i}}{\binom{N_p}{m}} \quad (1)$$

The expected number of keys identified by the second attacker, running KMP with the target, is slightly smaller: the first attacker has already identified some. And there is a chance the second attacker shares and thus identifies the same keys. On the average, we have $\delta_2 \approx \delta_1 \cdot \frac{m-\delta_1}{m}$. Similarly, the expected gain (of identified keys) for the i-th attacker is:

$$\delta_i \approx \delta_1 \cdot \frac{m - \sum_{j=1}^{i-1} \delta_j}{m} \quad (2)$$

Therefore, $T(N,1)$ is:

$$T(N,1) \approx \arg\min_{\ell} \{\sum_{j=1}^{\ell} \delta_j \geq \beta m\} \cdot N \quad (3)$$

where N represents the chance of targeting a specific node in each round of the SPA. Then, the time to identify at least βm keys of a subset of αN nodes among all the N nodes in a network is:

$$T(N, \alpha N) \approx T(N,1) \cdot \alpha N \ . \quad (4)$$

B. Persistent Key Request

The effectiveness of the adversary is enhanced (that is, the likelihood of compromising the communication of victims increases) with the number of distinct keys held collectively by adversarial nodes. In order to maximize that, the adversary could tamper with the process of bootstrapping nodes: at the point of adding one node to the system, it can try repeating the key assignment in order to get the least overlap with the keys all adversarial nodes currently possess. Ideally, the adversary would seek to get another m totally distinct keys.[3]

To approximate how costly such an effort can be for the adversary, we consider a simple model: the newly joining node makes a request and receives a set of symmetric keys. If those overlap with the ones held by other nodes under the adversary's control, the node repeats the request. We derive the expected number of key requests for one adversary node, given that there has been τ sets of completely different keys assigned (to τ other nodes); $\tau = 0, 1, \cdots (K-1)$. We assume that $m\tau \leq N_p$. This is essentially a process of choosing m keys from an N_p key pool until all m keys are different to the $m\tau$ keys that were previously assigned to other nodes. This is a geometric distribution with success probability p_r:

[3]It is also possible to reuse the distinctive keys assigned in early rounds instead of throwing them away. Such an approach will significantly speed up the key request process.

$$p_r = \frac{\binom{N_p - m\tau}{m}}{\binom{N_p}{m}} \tag{5}$$

Hence the expected number of requests from this attacker until it receives a completely different set of keys is:

$$R(\tau) = 1/p_r = \frac{\binom{N_p}{m}}{\binom{N_p - m\tau}{m}} \tag{6}$$

The expected number of key requests due to all K adversarial nodes trying to obtain different set of keys is then:

$$R_K = \sum_{\tau=0}^{K-1} R(\tau) = \sum_{\tau=0}^{K-1} \frac{\binom{N_p}{m}}{\binom{N_p - m\tau}{m}} \tag{7}$$

C. Secure Communication Compromise Probability

The main benefit of the SPA attack can be demonstrated when a focalized attack on a particular node's neighborhood is launched. Without SPA, the adversary picks the target node randomly, with a relatively small chance of compromising secure communications established by the target node. After the SPA, the adversary picks a target node that is now known to share keys with the adversary. The chance of compromising the secure communications established by the target node with other benign nodes is much higher.

More specifically, when βm among m keys are known to the adversary, the chance that any secure communication link established by the target node is compromised by the adversary is β. The smallest value of meaningful β is $\beta = 1/m$. The chance of finding any node that shares at least one key with the adversary (holding Km keys) is:

$$P_x(\text{SPA}) = 1 - \frac{\binom{N_p - Km}{m}}{\binom{N_p}{m}} \tag{8}$$

where we assumed distinct keys for the K compromised nodes.

The cost of identifying a target node in the SPA attack is simply the expected number of tries with the above successful probability, or a geometric distribution with success probability $1 - \frac{\binom{N_p - Km}{m}}{\binom{N_p}{m}}$. The cost can be computed as:

$$\frac{\binom{N_p}{m}}{\binom{N_p}{m} - \binom{N_p - Km}{m}} \tag{9}$$

On the other hand, when a target node is chosen randomly, the chance that any secure communication link established by the target node is compromised by the adversary is the same as that for choosing a key from the key pool that matches one in Km. Therefore, the chance is

$$P_x(\text{no SPA}) = Km/N_p \tag{10}$$

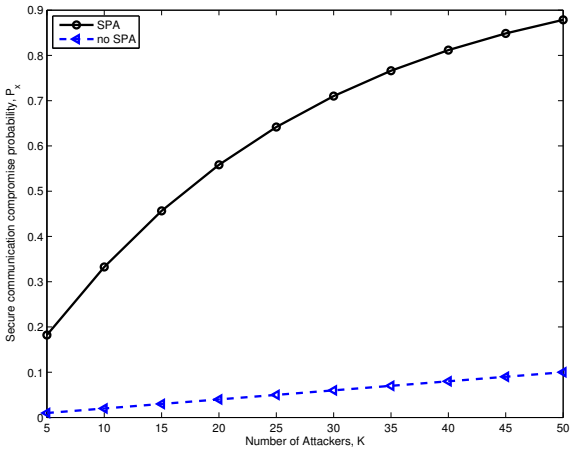

Figure 1. Probability to compromise secure communication, P_x. When SPA is used, the adversary can focus on the communications of a target node known to share keys with the K compromised nodes. Without SPA, the adversary can only randomly select a node with a much lower chance of success.

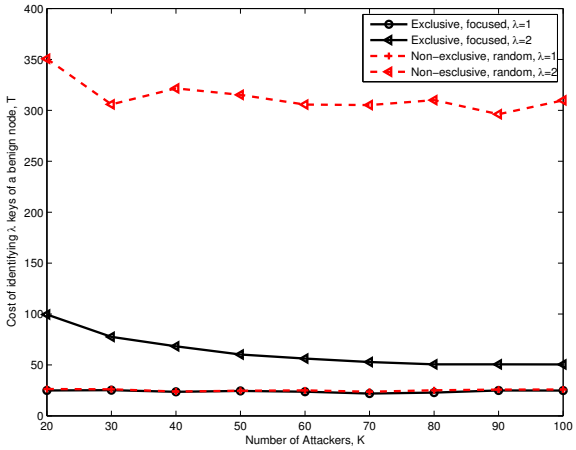

Figure 2. Cost of identifying λ keys of a benign node, T. For the "Exclusive, focused" attack, the adversary orchestrates K nodes with distinct key sets (e.g., thanks to persistent key request) and it runs KMP with each of its K nodes before moving on to another victim node. For the "non-exclusive, random" attack, nodes randomly chosen among the K adversarial nodes attack randomly chosen benign nodes. λ is the number of keys to identify per victim node in order to consider the SPA successful.

D. Performance Evaluation

Our performance evaluation is based on MATLAB simulations. In these simulations, each of the N benign nodes is randomly assigned m keys. The adversary controls K compromised nodes, each with m keys, and uses the SPA to identify keys of the benign nodes.

In Figure 1, we compare the secure link compromise probability with or without the SPA attack; the system parameters are $N_p = 10,000$, $m = 20$, $K = 10$ (Note: node density or wireless transmission range does not matter here). Since the attack after the SPA is focalized, the P_x value is much higher than the value for the random attacks. We can also see that P_x increases with K, the number of nodes controlled by the adversary.

In Figure 2, we demonstrate the cost, in number of rounds of KMP that need to be performed, for different SPA attack strategies. The number of benign nodes in the network is $N = 100$. The key pool size is $N_p = 10,000$ and $m = 20$. An interesting observation is that in both schemes, the costs to compromise $\lambda = 1$ keys of any node are similar. But when $\lambda = 2$, the "exclusive, focused" SPA attack is far more efficient (although more likely to be detected). Overall, as K increases, the cost decreases. This is because of the increased number of keys known by the adversary.

V. Countermeasures and Discussion

The basic element of the SPA attack is the seemingly benign use of the KMP process, so that the adversary can focalize its resources and attack. Moreover, we identified that adversaries could get more diverse key sets and, overall, more keys out of the pool.

First, the rate of the KMP protocol should be throttled back, kept as low as possible, to prevent the adversary from mapping benign nodes to a fraction of their keys. The ability to achieve this depends on the utility of the KMP and the functionality of the system. If, for example, the system does not change frequently, then the KMP is not often needed. After an initialization phase, nodes could decline engaging to KMP more than a protocol-selectable threshold. On the contrary, if the KMP is needed for reconfiguration, e.g., a change of secure communication path, then, the appropriate threshold must be chosen, in order to not harm the system functionality.

Second, the bootstrapping phase can also be regulated relatively easily: newly joining nodes may be simply rejected once and for all after a protocol-selectable number of (e.g., one or two) failures to conclude their key assignment.

Moreover, the addition of new nodes, many of which can deliberately be introduced by the adversary, can also be regulated. However, this also depends on the system application. If RKPD is used in an open system, with frequent reconfigurations (joins and leaves), other access control methods should be used to prevent the massive joins of adversarial nodes.

Finally, the effectiveness of the SPA could be reduced by a simple modification. Asking any node to respond with a message encrypted with (one of) the m keys it has, makes it easy for the adversary to check this response: it simply decrypts each of those m parts of the response with all (up to) the Km keys it has. To counter this, the KMP could be designed to require the initiator to provide its own encrypted nonces upfront and the responder to return a message on confirming a match (or no match if this is so).[4] This implies that the attacker would have to present m keys (encrypted nonces) at a time, and that the responder would reveal less information. This would increase the delay for the adversary to identify which keys are held by which nodes.

[4]Although out of scope for this paper, this latter approach may raise Denial of Service (DoS) attack concerns.

VI. Conclusion

In this paper, we focused on the Stealthy Pre-Attack (SPA). This is a refinement over brute force attacks, and an easy way for the adversary to work its way towards nodes whose communication it is more likely to compromise. An adversary could abuse the widely used key matching protocol to identify a portion of the keys carried by some benign nodes. The adversary would then be likely to compromise the secure communication links established by those targeted nodes. Our analysis and simulation showed that the SPA attack can be powerful against dynamic networks as well as networks that allow incremental node deployment.

Acknowledgment

This work was supported in part by an ACCESS Linnaeus Centre mobility grant.

References

[1] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *Proc. of the 9th ACM conference on Computer and communications security*, pages 41–47, Washington, DC, USA, November 18-22 2002.

[2] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proc. of IEEE Symposium on Security and Privacy*, pages 197–213, Berkeley, California, May 11-14 2003.

[3] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *Proc. of the ACM Conference on Computer and Communications Security (CCS '03)*, pages 42–51, Washington, DC, USA, October 27-31 2003.

[4] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *Proc. of the ACM Conference on Computer and Communications Security (CCS '03)*, pages 52–61, Washington, DC, USA, October 27-31 2003.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE communications Magazine*, pages 102–114, August 2002.

[6] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. Technical Report 02-207, Carnegie Mellon University, 2002.

[7] T. Moore. A collusion attack on pairwise key predistribution schemes for distributed sensor networks. In *Proc. of the 3rd Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, Pisa, Italy, March 2006.

[8] F. Kausar et al. A key distribution scheme preventing collusion attacks in ubiquitous heterogeneous sensor networks. In M. S. Denko et al., editors, *Emerging Directions in Embedded and Ubiquitous Computing*, volume 4809 of *Lecture Notes in Computer Science*, pages 745–757. Springer Berlin / Heidelberg, 2007.

[9] D. Tran Thanh and J.I. Agbinya. Combating key-swapping collusion attack on random pairwise key pre-distribution schemes for wireless sensor networks. *Security and Communication Networks*, 4(2):109–121, 2011.

[10] T. Bonaci, L. Bushnell, and R. Poovendran. Node capture attacks in wireless sensor networks: A system theoretic approach. In *Proc. of the 49th Conf. on Decision and Control (CDC '10)*, pages 6765–6772, December 2010.

[11] D. Liu. Protecting neighbor discovery against node compromises in sensor networks. *International Conference on Distributed Computing Systems*, pages 579–588, 2009.

[12] A. Seshadri, M. Luk, and A. Perrig. Sake: Software attestation for key establishment in sensor networks. *Ad Hoc Networks*, 9(6):1059 – 1067, 2011.