

KPI: A Security Infrastructure for Trusted Devices

Mahalingam Ramkumar

Department of Computer Science and Engineering
Mississippi State University, Mississippi State, MS 39762
Ph: 662-325-8435, Email: ramkumar@cse.msstate.edu

Nasir Memon

Department of Computer and Information Science
Polytechnic University, Brooklyn, NY 11201
Ph: 718-260-3970, Email: memon@poly.edu

I. INTRODUCTION

Deployments of perhaps billions of autonomous, heterogeneous wireless devices, some fixed and some mobile, manufactured by different vendors, with varying capabilities, and very different purposes, but with *one common feature* - that every device will have the ability to communicate with *any other* device - are expected to organize themselves into pervasive, highly interconnected, ad hoc networks. Such pervasive networks would serve as crucial infrastructures for our day to day computing / communication needs. Securing such deployments from malicious intents, aimed at sabotaging the infrastructure, is a very important requirement.

For example, nodes forming mobile ad hoc networks (MANETS) have to co-operatively build routing tables, and relay messages destined for other nodes. In such a scenario, malicious action by a single node could have a potentially disruptive effect over the entire network. An attacker “controlling” one or more nodes can inflict significant harm to *other* nodes. It is therefore vital that the nodes (or devices) people possess (or operate) “behave responsibly.” While it may not be possible to force the owners of the nodes to behave in a responsible fashion, it may be possible to force the *devices themselves* to do so. In other words, *it is the devices that are trusted - not the owners!*

This new paradigm shift (trusting devices instead of trusting the owners) is needed not just in applications that depend on mutual co-operation for functioning, but also under scenarios where

- 1) devices need to operate autonomously (there is no person around to supply the device with secrets when necessary), and
- 2) devices that need to operate in hostile environment (example, DRM applications, where the owner of a DVD player might be a potential pirate).

Two devices can trust each other if there exists some means of convincing each other that they “play by the rules,” or are “compliant” (to some pre-imposed rules). From a cryptographic perspective, two nodes can trust each other if they can establish an *authenticated* shared secret. This is facilitated by a key distribution scheme (KDS), which provides each node with one or more secrets. The KDS secrets are then used to *establish* (or discover) *shared* secrets. The fact that such a shared secret can be established simultaneously provides

mutual *authentication* (of the identities) of the parties involved - or the interacting parties establish a *security association* (SA).

The KDS secrets provided to a node could however, be used as a *hook* for compliance. In other words, only nodes (or devices) that have been *checked* for compliance would be provided with the necessary secrets. Thereafter, the ability of any two nodes to establish an SA, indirectly provides a means for *verification of compliance*.

Any security solution based on *trusted devices* therefore demands mechanisms for *read-proofing* the secrets stored in *tamper-resistant* devices [1]. In the absence of the assurance of read-proofness, secrets that serve as a hook for compliance could be *transferred* to non-compliant [2] devices. In the absence of the assurance of tamper-resistance, the components (or software) that ensure compliance of a device could be modified.

At a minimum, a deployment of trusted devices consists of a trusted authority (TA) who manufactures the devices, and the devices themselves. However, in practice, devices may be manufactured by different vendors (or different TAs). Therefore, the need for interoperability demands that the KDS should provide for establishment of security associations (authenticated shared secrets) even between devices manufactured by different vendors.

For long-lived security of the deployment of devices, the KDS secrets stored in a device (that guarantee compliance), should be *renewed* periodically. Further, the KDS should offer mechanisms for *revocation* of devices (revoked devices will not be able to take subsequent part in the deployment). Additionally, the KDS should also provide for *non-repudiation* of messages sent by devices. It would also be very useful if the underlying KDS provides solutions for *multicast security*.

A trusted device A , then consists of components that render the device compliant, and the set of secret(s) \mathbb{S}_A , all enclosed in a read-proof and tamper-resistant casing. For example, each device may have a general purpose processor. The software that runs on the processor determines the “rules” that the device honors. Only the processor in device A will have access to the secrets \mathbb{S}_A . The nature and number of secrets \mathbb{S}_A would depend on the underlying KDS used to secure the deployment.

II. KPI - KEY PRE-DISTRIBUTION INFRASTRUCTURE

For applications involving nodes forming ad hoc networks, privacy and practicality constraints dictate that interactions between any two nodes, for purposes of establishing security associations, should not need external mediators - thus ruling out Kerberos as a viable option. While PKI, based on asymmetric cryptography, supports ad hoc establishment of security associations, the computational demands placed by asymmetric cryptography may not be acceptable in all scenarios.

A third option is key pre-distribution (KPD) [3]. A KPD scheme consists of a trusted authority (TA), and N nodes with unique IDs (say $ID_1 \cdots ID_N$). The TA chooses P secrets \mathcal{R} . The node i is preloaded with preloaded secrets $\mathbb{S}_i = f(i, \mathcal{R})$ - the *key-ring* of node i . Two nodes and \mathbb{S}_j can discover a unique shared secret K_{ij} using a *public* operator $g()$ without further involvement of the TA.

$$K_{ij} = g(\mathbb{S}_i, ID_j) = g(\mathbb{S}_j, ID_i). \quad (1)$$

As $g()$ is public, it possible for two nodes, just by exchanging their IDs, to execute $g()$ and discover a unique shared secret. The nature of the functions $f()$ and $g()$ determine the actual KPD scheme.

However, as the keys stored in different devices are *not* independent, an attacker, by exposing secrets from a finite number of devices, may be able to compromise secrets of other devices, or even compromise all the secrets \mathcal{R} . There is thus a concept of n -secure KPDs. Typically, the efficiency of a KPD scheme is measured as a ratio of n vs the *key-ring size* required in each device.

The KPI (or key pre-distribution infrastructure) [4] consists of a KPD scheme at its core, and security *policies* and *protocols* to render the deployment inter-operable and secure.

We propose the use of HARPS (hashed random preloaded subsets) [5] as the underlying KPD for the KPI, The security policy for the envisaged KPI is an *extension* of the “resurrecting duckling” policy in by Stajano et al [6] - [7]. The extension of the security policy is based on a delay based circuit authentication technique proposed by Gassend et al [8], which permits *remote* resurrection of the duckling - or in other words, safe renewal of the preloaded secrets *without physical contact* between a device and the TA [9].

The tree-hierarchical deployment of KPI starts with a root node at the root of the tree. Each child node could further act as TAs (vendors) for their child nodes (devices manufactured by the vendors). Each node, in accordance with HARPS, is preloaded with a subset of secrets belonging to its parent. However, the preloaded secrets are repeatedly hashed a variable number of times.

The tree hierarchical nature of the deployment permits devices manufactured by different vendors to establish security associations. Further, the preloaded HARPS secrets, apart from being used for establishing pairwise security associations can also be used for

- 1) Discovery of conference secrets
- 2) Broadcast authentication [10] - or non-repudiation of the source, and
- 3) Broadcast encryption [11]

In particular, HARPS permits even peer nodes (or devices) to perform authenticated broadcasts and broadcast encryption. Broadcast authentication by the TA can be used for broadcasting revocation lists similar to PKI. An even more efficient mechanism of revocation is rendered possible through broadcast encryption by the TA. The TA could broadcast revocation secrets that would not be decipherable by revoked nodes. Note that if broadcast authentication is used for revocation, the nodes would need to store list of revoked devices. However, if broadcast encryption is used, nodes need to store only the latest revocation secret (which is not available to the revoked nodes).

A combination of different security primitives could also be used to realize more complex security associations like establishment and maintenance of communities of interests (or multicast groups), and also provides a security framework for peer-to-peer publish-subscribe [12] systems.

A unique feature of broadcast authentication using HARPS is that it caters for a novel cryptographic paradigm of “targeted signatures” [13]. While a typical signature schemes do not differentiate, or do not have the ability to differentiate, between intended and non-intended recipients of a broadcast, for most practical applications, most messages do in fact have intended and non-intended recipients. HARPS enables signatures can be *targeted* to one or more verifiers.

REFERENCES

- [1] R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali, T. Rabin, “Tamper Proof Security: Theoretical Foundations for Security Against Hardware Tampering.” Theory of Cryptography Conference, Cambridge, MA, February 2004.
- [2] J. Lotspiech, S. Nussler, F. Pestonoi, “Anonymous Trust: Digital Rights Management using Broadcast Encryption,” Proceedings of the IEEE, **92** (6), pp 898–909, 2004.
- [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] M. Ramkumar, N. Memon, “A Hierarchical Random Key Pre-distribution Scheme for a Low Complexity Security Infrastructure,” submitted to the IEEE Information Assurance Workshop, 2005.
- [5] M. Ramkumar, N. Memon, “An Efficient Random Key Pre-distribution Scheme for MANET Security,” to appear, IEEE Journal on Selected Areas of Communication, March 2005.
- [6] F. Stajano, R. Anderson. “The Resurrecting Duckling: Security Issues in Ad-Hoc Wireless Networks.” In “Security Protocols, 7th International Workshop Proceedings”, Lecture Notes in Computer Science. Springer-Verlag, 1999.1
- [7] F. Stajano, “The Resurrecting Duckling - what next?,” available at <http://www-lce.eng.cam.ac.uk/fms27/duckling/duckling-what-next.html>.
- [8] B. Gassend, D. Clarke, M. van Dijk, S. Devadas, “Delay-based Circuit Authentication and Applications,” Proceedings of the 2003 ACM symposium on Applied Computing, Melbourne, Florida, pp 294 – 301, 2003.
- [9] M. Ramkumar, “On Key Renewal in Trusted Devices,” submitted to ICDCS 2005.
- [10] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas, “Multicast Security: A Taxonomy and Some Efficient Constructions,” INFOCOMM’99, 1999.
- [11] A. Fiat, M. Naor, “Broadcast Encryption,” Lecture Notes in Computer Science, *Advances in Cryptology*, Springer-Verlag, **773**, pp 480–491, 1994.
- [12] C. Wang, A. Carzaniga, D. Evans, and A. Wolf, “Security Issues and Requirements in Internet-scale Publish-subscribe Systems.” In HICSS’02, January, 2002.
- [13] M. Ramkumar, “Targeted Signatures: Broadcast Authentication with Hashed Random Preloaded Subsets,” submitted to the IEEE Symposium on Security and Privacy 2005.