Abstract

This document describes a simple key management approach for the PIM multicast routing protocol, observing the key arrangement for PIM defined in [Wei98] for PIM version 2.

1. Introduction

The PIM Working Group (IETF) has recently put forward a proposal [Wei98] on the arrangement of cryptographic keys within a given PIM domain, with the aim of deploying the keys for control-packet authentication in the PIM domain. Although the proposal suggests a key arrangement, it purposely does not specify how the keys are to be delivered to the relevant network entities in the PIM domain.

The current work covers a key distribution method based on the use of a combination of public key cryptography and private key cryptography. A Domain Key Distributor (DKD) entity is introduced which performs the initial dissemination of keys in the domain and the re-keying of the keys on a periodic basis.
The current work aims at a limited form of key management specifically for PIM, anticipating the development of a generic key management standard in the future. Although the solutions described here are PIM-specific, some of the methods can be developed further for a generic key management framework.

2. Background

Following [Wei98], when security is enabled, all PIM version 2 messages will carry an IPSEC [KS98a] authentication header (AH) [KA98c]. The authentication mechanism MUST support HMAC-MD5-96 [MG98a,Riv92] and HMAC-SHA-1-96 [MG98b] security transformations.

The PIM key arrangement of [Wei98] identifies the following entities in a PIM domain that require keys: the BootStrap Router (BSR), the Rendezvous Point (RP), the Designated Router (DR) and other PIM routers. All keys are relevant and recognized only within one PIM domain.

For clarity, here we denote the following symbols for the respective keys described in [Wei98]. Public key pairs are described using the symbol SK ("Secret Key") and PK ("Public Key"), whereas private (symmetric) keys are described using the symbol "K" with the appropriate notation. The keys are defined as follows:

- All PIM routers in the same domain still share a single secret (the "equal opportunity" key) that is used to compute digests for PIM messages. This key is denoted as Keq.

- All BSRs own an identical RSA [RSA93] key pair and uses the private key to sign an entire bootstrap message, whilst other routers only have the public key to verify the signature. This RSA secret-public key pair is denoted as (SKbsr, PKbsr). This allows only authorized candidate BSRs to become a bootstrap router.

- All RPs and BSRs share another symmetric key, known as the "RP-key" and denoted here as Krp. No other routers have this key. For candidate RP advertisement the digest is only calculated with the RP-key Krp (instead of the equal opportunity key Keq). This achieves the effect that only the authorized candidate RPs can advertise their candidacy to the BSR.

For convenience the keys of [Wei98], namely RSA key-pair (PKbsr, SKbsr), Keq and Krp, will be referred to subsequently as "Primary Keys" in order to distinguish them from other cryptographic keys.
3. Key-Management Keys: Assignments

In order to deliver the relevant primary keys specified in [Wei98], we define a further set of keys, which will be referred to as "Key Management Keys" (KM-Key). These keys are long-term keys whose use is primarily for delivering the Primary Keys. Thus unlike the Primary Keys which are used frequently on control-messages, the KM-keys need not be rekeyed very often.

- The Domain Key Distributor (DKD) is assigned the RSA key pair (PKdkd, SKdkd). Following the "semi-public" usage of public key technology, the key PKdkd is only known by PIM-routers within the domain. Only the DKD knows SKdkd.

- All Candidate-RPs (namely the routers from which the set of RPs will be selected) and the BSR are assigned the "semi-public" RSA public key PKrpbsr, whose matching secret half SKrpbsr is only known to the DKD. The DKD also knows PKrpbsr. Note that all RPs share the same PKrpbsr.

Note that all public key pairs defined in the current document is used in a "semi-public" or "closed" system, in which only certain designated entities know the public-key of other entities. All keys are uniquely functioning within only one PIM domain. Hence, implicit authentication is achieved through the limited sharing of the public-keys among certain entities, while confidentiality can established by the holder of the secret-key by enciphering messages to be sent using the secret-key.

The BSR key pair (SKbsr, PKbsr) will also be used on occasion for supporting key-management.

In the remainder of the current document, timestamping, nonces and other anti-replay mechanisms are assumed to be incorporated, both in the initial dissemination of keys and in their subsequent re-keying.

4. Initial Key Dissemination

In order to kick-start the key management process, two avenues are possible. The first is by manual configuration, while the second is based on a secure channel opened between the DKD and each PIM router.

A. Manual Configuration

Here all the following keys must be manually configured into the corresponding entities:

- the key PKdkd of the DKD is manually configured into all PIM-routers (including the BSR and RPs)
the pair (PKbsr, SKbsr) is manually configured into the BootStrap Router (BSR)

- the key PKrpbsr is manually configured into the BSR and the RPs.

B. Online Dynamic Configuration

This approach requires each router Ri to have a global public key pair (PKri, SKri), which may be manually configured or loaded manually, preferably using a secure method (eg. smartcard). Using its public key pair, a router in the domain establishes a Security Association (SA) and a secure channel with the DKD.

- The BSR can either generate the pair (PKbsr, SKbsr) and then upload a copy of PKbsr to the DKD through the secure channel, or the DKD can generate the pair (PKbsr, SKbsr) and download the pair to the BSR through the secure channel.

- The BSR must download a copy of the key PKrpbsr (generated by the DKD) from the DKD through the secure channel.

- All RPs must download a copy of the key PKrpbsr (generated by the DKD) through a secure channel which they individually establish with the DKD.

- All PIM routers must download a copy of the key PKdkd (generated by the DKD) through a secure channel which they individually establish with the DKD.

5. Dissemination of other Primary Keys

Through the initial key dissemination process (above), only the key pair (PKbsr, SKbsr) of the Primary Key have been disseminated to the BSR. What remains is the dissemination of the other Primary Keys to the other relevant entities. This is discussed in the following.

5.1 Disseminating PKbsr to All PIM-Routers

The DKD digitally-signs the public key PKbsr (using its secret-key SKdkd) and sends the result to all PIM-routers in the domain, either through unicast or through the distribution tree itself (eg. the "All-PIM-Routers" group). In effect, the DKD vouches for the BSR. Only PIM-routers (holding PKdkd) can verify the signature. If confidentiality is needed, then the DKD can encrypt PKbsr rather than digitally signing it.
5.2 Disseminating Krp shared by BSR and RPs

The DKD must encrypt Krp using the secret-key SKrpbsr (known only to the DKD) and send the resulting ciphertext to the BSR and all RPs. This can be done by multicasting to the special "All-PIM-Routers" group in the domain. Since only the BSR and RPs have the matching key PKrpbsr, only they will be able to decipher the message to obtain the key Krp. All other PIM-routers will drop this message (unreadable).

5.3 Disseminating Keq for all PIM-routers

The DKD encrypts the equal-opportunity-key Keq using its secret-key SKdkd and sends the ciphertext to all PIM-routers in the domain, either through unicast or through the distribution tree (e.g. the "All-PIM-Routers" group). Since only PIM-routers have the "semi-public" key PKdkd, only PIM routers will be able to decipher the ciphertext to obtain Keq. All other routers will drop this message (unreadable).

6. Re-Keying the Primary Keys: Strategy

The nature of public keys versus private keys requires their re-keying to be carried-out in differing manners. In the following, the re-keying of each of the Primary Keys is discussed, with some available options being described. In order to distinguish the old keys and the fresh keys, the numbers "1" is attached to the old keys and the number "2" is attached to the fresh keys.

6.1 Re-keying the BSR keys using installed keys:

The first step in the re-keying of the BSR public key pair consist of re-keying the BSR entity itself first. Two options are available depending on whether the BSR or the DKD generates the new key pair.

A. If the DKD generates the new key pair (SKbsr2, PKbsr2), then it must deliver the pair to the BSR. This can be done as follows:

   (1) The DKD encrypts (or digitally-signs) the new pair (SKbsr2, PKbsr2) under the DKD’s secret-key (SKdkd) resulting in a ciphertext Cbsr.

   (2) The DKD further encrypts the ciphertext Cbsr with the current BSR public-key PKbsr1 resulting in a ciphertext CCbsr. The DKD then unicasts the final ciphertext CCbsr (containing the new key) to the BSR. Only the BSR holding SKbsr1 will be able to decipher CCbsr.
B. If the BSR generates the new key pair (SKbsr2, PKbsr2), then it must deliver the public half PKbsr2 to the DKD. This can be done as follows:

(1) The BSR encrypts (or digitally-signs) the new key PKbsr2 under the BSR’s current secret-key (SKbsr1) resulting in a ciphertext Cdkd.

(2) The BSR further encrypts the ciphertext Cdkd with the current DKD public-key PKdkd resulting in a ciphertext CCdkd. The DKD then unicasts the final ciphertext CCdkd (containing the new key) to the DKD. Only the DKD holding SKdkd will be able to decipher CCdkd.

6.2 Re-keying the BSR keys using a separate secure channel:

An alternative approach is for a separate secure channel to be created between the DKD and the BSR through the existing security associations.

The BSR can either generate the new pair (PKbsr2, SKbsr2) and then upload a copy of PKbsr2 to the DKD through the secure channel, or the DKD can generate the pair (PKbsr2, SKbsr2) and download the pair to the BSR through the secure channel.

6.3 Re-keying Krp:

The key Krp shared between the BSR and the RPs can be re-keyed using the previous method (Section 5.2) or using the existing key Krp1. Here we assume that the DKD generates the new Krp2 for equality between the BSR and RPs, and for simplicity.

The method to re-key using the existing Krp1 is as follows:

(1) The DKD encrypts (or digitally-signs) the new key Krp2 under the DKD’s secret-key SKdkd resulting in a ciphertext Crp.

(2) The DKD encrypts the ciphertext Crp with the existing key Krp1 resulting in a ciphertext CCrp. The DKD then unicasts CCrp to the BSR and each RP, or the DKD multicasts CCrp to the special "All-PIM-Routers" group in the domain. Only the BSR and the RPs will be able to decipher CCrp since only they hold Krp1. All other PIM-routers will drop this message (unreadable)
6.4 Re-keying Keq in all PIM-routers:

Since all PIM-routers hold Keq1 and hold the DKD's "semi-public" key PKdkd, the best way to replace Keq1 with a new key Keq2 is using the previous method (Section 5.3). That is, the DKD encrypts the new Keq2 using its secret-key SKdkd and multicasts the ciphertext to all PIM-routers in the domain. Only PIM-routers holding PKdkd will be able to obtain Keq2. All other routers will drop this message (unreadable).

Here, for simplicity, we assume that the DKD generates the new Keq2. A combined method can also be adopted, employing the DKD’s key and the existing Keq1:

1. The DKD digitally-signs the new key Keq2 using the DKD’s secret-key SKdkd resulting in a digest.

2. The DKD encrypts the pair (Keq2, digest) under the existing key Keq1 resulting in a ciphertext CCEq. The DKD then multicasts CCEq to all PIM-routers in the domain. Only PIM-routers holding Keq1 will be able to decipher CCEq to obtain Keq2. All other routers will drop this message (unreadable).

7. Interdomain control-message authentication

The current key management approach is limited to a single PIM-domain due to the use of public keys in a "closed" fashion (ie. "semi-public"), where the public half of the keys are made known only to a limited number of entities.

Currently, in order to allow some control-messages from a PIM-entity in a PIM-domain D1 to cross the domain boundary to another PIM-entity in a different PIM-domain D2, the DKDs in the respective domains must be assigned global public keys in the traditional (fully-public) manner of use of public keys. Through these global public keys the DKDs can negotiate an intermediate-key used by a PIM-entity in domain D1 to communicate (with authentication and/or confidentiality) with another PIM-entity in another domain D2. In other words, the DKD in a domain vouches for all the PIM-entities in its domain.

An improvement over this approach would be for each PIM-entity to be assigned a long-term global public key pair. This would allow router-to-router authentic and/or confidential communications across domain boundaries, without the intervention of the DKDs of the
respective domains. Furthermore, it would allow the DKD of a domain to create a secure channel with individual routers in its domain in order to carry-out key management and other network management functions securely.

8. References


9. Author’s Addresses

Thomas Hardjono
Email: thardjono@baynetworks.com

Brad Cain
Email: bcain@baynetworks.com

Nortel Networks
600 Technology Park Drive
Billerica, MA 01821, USA
Tel: +1-978-288-4538
Thomas Hardjono
email1: thardjono@yahoo.com
email2: hardjono@nortelnetworks.com
Tel: +1-978-288-4538