Automated Key Management for Router Updates

J. William Atwood
Department of Computer Science and Software Engineering
Concordia University
Montreal, Quebec Canada
E-mail: bill@cse.concordia.ca

***Abstract*—Security is assuming increasing importance in emerging networks. To ensure application security, the routing protocols are assumed to be trusted. Correct forwarding of packets requires the prior exchange of information among routers, using the appropriate routing protocol. Valid construction of the routes requires that the exchanged information be received from a legitimate neighbor, and that it not be altered en route, i.e., the inter-router communication has to be secure. This requires, in turn, an architecture for managing the necessary security associations and keying material within an administrative region. After reviewing current work and existing security tools, a proposal is made for a novel architecture to manage the router updates. The operations within this architecture are detailed, and its advantages are explained.**

***Index Terms*—Routing; Security; Key Management**

I. Introduction

The routing of packets through an internet involves two levels: *routing*, which is the exchange of information to permit determining the "best" path for a packet to take toward its destination, and *forwarding*, which is the act of sending a particular packet closer to its destination.

During the execution of the routing process, care must be taken to ensure that the routing information is exchanged with legitimate "neighbors" (however this is defined for a particular routing protocol), otherwise an incorrect forwarding path may be defined.

This implies the necessity for authentication and integrity protection for the exchanged routing packets, i.e., "securing" these packet exchanges. Unfortunately, for those Internet routing protocols for which security specifications are defined, most of the existing specifications are for manual keying procedures, which are inappropriate for a large internet.

In emerging networks, more and more emphasis will be placed on the security of the exchanges. Trust at the application level must be based on the existence of trust at the lower levels. If routing-level security is based on manual procedures, then the necessary trust cannot be maintained. Deployment of automated procedures for managing security in routing will be an essential part of all networks.

Within the Internet Engineering Task Force (IETF), the Area Directors responsible for routing and the Area Directors responsible for security have realized that a "road map" is necessary to ensure that security goals are (eventually) met while incremental progress is made in making the routing protocols more secure. They have started an effort within the IETF to design a framework for this roadmap. While the current document [1] has a single author, it represents a broad consensus within the two Directorates concerning the path to be followed.

This effort has two major steps:

1) enhancing the current authentication mechanism of the base routing protocol, to ensure that it employs modern cryptographic algorithms and methods for its basic operational model;
2) defining the use of a key management protocol (KMP) for creating and managing the session keys used in the message authentication and data integrity functions of the base routing protocols.

The second step provides the potential to replace error-prone, labor-intensive, and insecure operational procedures with an automated procedure, where keys can be effectively managed with less overhead than is presently required for manual keying.

This paper presents a possible structure for an automated key management system, which brings together existing "standard" security solutions; recently-proposed extensions to these standard procedures, which encompass the multicast case; and a novel architecture for the management of keying material.

In Section II, we outline current work that is relevant to the problem. In Section III, we discuss the existing security tools. Section IV presents our model of the expected communication among the routers. Section V outlines the architecture of the proposed solution, while Section VI discusses the operation of the system. Section VII gives our conclusion.

II. Current Work

In this section, we give examples of near-neighbor communication in routing protocols. We then discuss our assumptions and some requirements that a key management system must meet.

A. Near-neighbor communication for routers

Routers communicate with their "neighbors" to establish the best route for a packet to take. A "neighbor" is likely to be defined differently for different routing protocols. For a unicast routing protocol such as OSPF [2], a neighbor is typically directly connected. For an exterior routing protocol such as BGP [3], a neighbor may be in a different Autonomous System (AS). For a multicast routing protocol such as PIM-SM [4], a neighbor is always directly connected (either on a shared medium or at the other end of a point-to-point link).

Router-to-router communication may be one-to-one or one-to-many. Securing this communication is typically achieved by establishing an IPsec Security Association (SA). (See Section III for the details.) In the first case, unicast communication is used, with a one-to-one SA. In the second case, multicast communication *may* be used, along with a group SA. For some routing protocols, a router will always communicate with a near neighbor using multicast; for other protocols a router may send some messages directly to a specific neighbor, using a unicast SA, and others to the same peer as part of a multicast SA.

Procedures for securing OSPFv3 packet exchanges are given in RFC 4552 [5]. This RFC states that 'it is not scalable and is practically infeasible to use different security associations for incound and outbound traffic to provide the required "one to many" security. Therefore the implementations must use manually configured keys with the same SA parameters for both inbound and outbound SAs.'

A similar set of procedures is specified for PIM-SM in [6]. However, the PIM-SM document recognizes the potential for automated key management in future specifications.

Note, however, that both of these documents specify how the communication is to be done, under the assumption that the keys and other parameters are in place. They are silent on the issue of how those keys and other parameters are to be installed and managed.

B. Trust structure

Before it is possible to establish the authenticity of a particular peer router, there must be some form of agreed mechanism for establishing that identity. We assume the existence of such a mechanism, but do not define its precise form. We note that this mechanism is not likely to be based on the IP address(es) of a router, because of the ease with which they may be spoofed.

We assume that all routers are part of a single administrative domain. The problem of securing multi-domain (i.e., inter-domain) routing is being actively investigated by the Secure Inter-Domain Routing (SIDR) working group of the IETF [7]. We observe that these inter-domain trust relationships are likely to be managed on a peer-to-peer basis. In this case, existing unicast security protocols are already perfectly adequate. Indeed, the primary goal of the SIDR working group is to specify *what* is being exchanged, rather than *how*. For single-domain (i.e., intra-domain) security, the one-to-one trust relationships can also be managed on a peer-to-peer basis, using existing unicast security protocols. However, the one-to-many trust relationships must be managed centrally, to ensure coordination over the entire administrative region. This is especially important since a single multicast address (for example, ALL_OSPF_ROUTERS) is typically specified for use by all routers implementing a particular protocol.

As will be seen, this central manager can be disabled for periods of time, without affecting the ability of the individual routers to continue their existing near-neighbor relationships. Thus, replication of the central manager is not a requirement, because continuous operation is not necessary. In addition, the scope of the central manager's operation is a single administrative region, so Internet-scope scalability is not required.

C. Communication constraints

Certain routing protocols operate "above" the unicast routing protocols, in the sense that they only begin to operate once unicast paths are available. As such, they can always assume that a path to the keyserver (or its replicate) exists.

The unicast routing protocols themselves, however, can make no such assumption. This implies that the initial boot-up, and reboots caused by power failures or scheduled maintenance, must be carefully considered.

D. Operational issues

According to Lebovitz [1], few operators have deployed routing security, and most of those who have report deploying one single manual key throughout their network. This clearly leaves them exposed to an attack by a terminated employee, or by someone to whom a terminated employee gave this single key. However, manually managing different keys is difficult, and changing those multiple keys periodically can be a significant overhead, which tends to overshadow the fact that increased security will result. This provides a strong motivation to develop automated procedures, as long as these procedures can be shown to be easy to manage. As Lebovitz [1] so clearly states, "Whatever mechanisms are specified need to be easier than the current methods to deploy, and should provide obvious operational efficiencies along with significantly better security and threat protection."

In addition, since operators will seldom have the personnel to spare for a massive conversion of their network, it is important to formulate a solution that can be incrementally deployed.

III. Existing Security Tools

In this section, we give a brief overview of IP Security and the Multicast Group Security Architecture. These concepts form the basis for our Automated Key Management proposal.

A. Internet Protocol security

The Security Architecture for the Internet Protocol is specified in RFC 4301 [8]. The IP Authentication Header [9] and IP Encapsulating Security Payload [10] documents define the security headers for IP packets. Other documents specify the Internet Key Exchange (IKEv2) Protocol [11], [12], and the cryptographic algorithm requirements for the use of the above protocols [13], [14]. The primary concept in IP Security (IPsec) is the Security Association (SA), which holds the information about a secure relationship between two end points. This information is maintained in three data structures: the Security Association Database (SAD), the Security Policy Database (SPD) and the Peer Authorization Database (PAD). For the unicast case, the information in the SAD is either manually inserted by an administrator, or is negotiated using a key management protocol such as IKEv2 when a new SA has to be established.

The base IPsec documents permit an SA to have a multicast address as a destination address, but provide no mechanism in the SPD for the recording of policies relating to multicast communication. RFC 5374 [15] defines the Multicast Extensions to the Security Architecture for the Internet Protocol. It extends the SPD to form the Group Security Policy Database (GSPD), and provides new semantics for the operation of an SA. As with the definition of unicast IPsec, the key management is left to separate documents.

B. Multicast group security

The Multicast Group Security Architecture is specified in [16]. See Figure 1 for the details of the reference framework. This document provides semantics for a Group Security Association (GSA), which consists of a Registration SA, a Re-key SA, and a Data Security SA. The Group Security Architecture is targeted to large groups and very large groups. It is also more general than the IPsec architecture, in that it is applicable to network-level, transport-level and application-level groups. Specific instances of key management architectures for multicast groups are defined in the Group Domain of Interpretation (GDOI) specification [17], the Group Secure Association Key Management Protocol (GSAKMP) specification [18] and the Multimedia Internet Keying (MIKEY) specification [19].

IV. Communication Patterns

When a set of routers share information with their near neighbors using multicast, all routers send to the same multicast group address. However, for a multicast Security Association, RFC 4301 permits using the source address to select the appropriate SA for processing an arriving packet. As long as the source address used by a sending router is unique on a network segment, each receiving router can determine the source of incoming packets.

The packets are sent with the Time To Live (TTL) field set to 1, ensuring that they are not forwarded from one network segment to another (i.e., the packets are “link-local”).

Thus, the communication pattern consists of many independent “speakers” (one per router) each sending to the set of near-neighbors (i.e., those routers that are directly connected to the speaking router). For a specific example of these patterns in the case of PIM-SM, see [6]. This “group per speaker” model forms the basis for our proposed system architecture.

V. System Architecture

In this section, we discuss how the assumptions and requirements in Section II can be used in conjunction with the existing security tools (Section III) and the observed communications model (Section IV) to formulate a framework for automated key management, when the distribution mechanism is based on multicast delivery.

A. Controling adjacency

A key management system cannot accept the appearance of a new neighbor automatically, because the new neighbor could be an intruder. Therefore, the management of adjacency has to

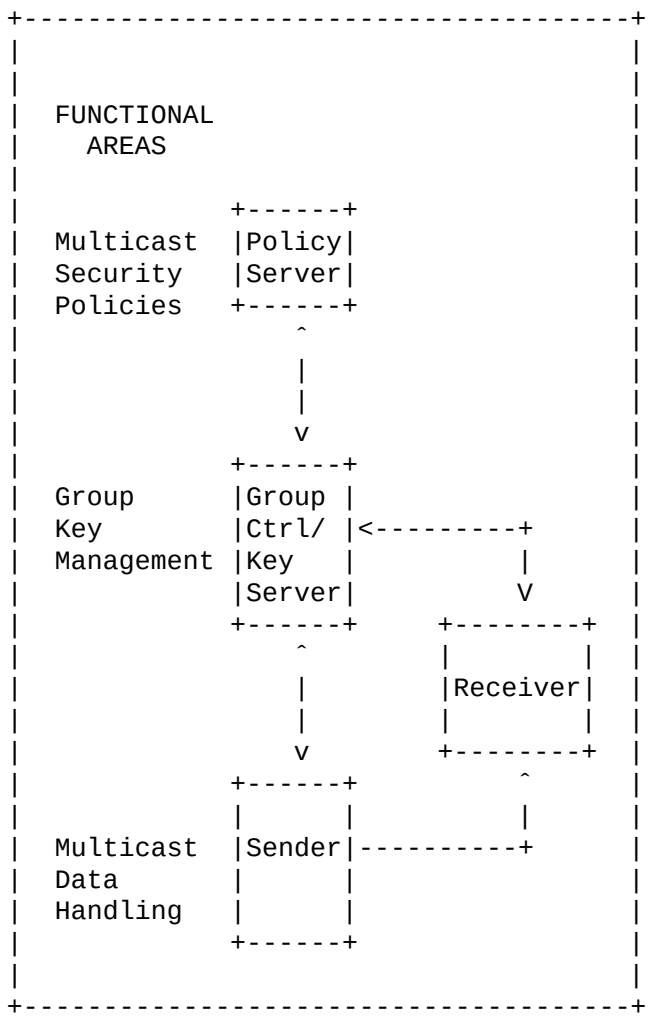

Fig. 1. Centralized Multicast Security Reference Framework [16]

be under the control of the (human) system administrator(s). The adjacencies will be managed through some form of human-computer interface, and represented as policies in the Policy Server of the Multicast Security Reference Framework. When a request is made for information about a specific group (i.e., the group associated with a specific sending router), the policy information will be consulted to ensure that the adjacency constraints are respected.

B. Establishing trust

IETF key management procedures are specified as two-phase protocols. The phase 1 protocol is used to manage a unicast (peer-to-peer) relationship between the Key Server and an individual router. It provides peer authentication, confidentiality and message integrity for the Phase 2 protocol.

As mentioned in Section II-B, we assume that there is some form of agreed mechanism for establishing the identity of a router, independently of its IP address(es). This identity will provide the basis for the peer authentication. An example IETF

protocol that could be used is IKEv2 [11], which supports the use of various peer authentication methods.

The Phase 2 protocol will be responsible for actually distributing the keys and other parameters from the Key Server to the routers that form a near-neighbor group clustered on a particular speaking router. An example IETF protocol that is suitable for this phase is GDOI [17]. However, note that the "groups" being managed here are carrying "control information" for the correct operation of the multicast routing protocol, not "user data".

C. Communications constraints

If the assumption of connectivity to the key server (Section II-C) can be made (which is true in the PIM-SM case), then the Group Controller/Key Server (GC/KS) that is used for the management of the keys can be centrally located (and duplicated for reliability). In this case, the needs of the group structure can be met using the existing Multicast Group Security Architecture [16]. If the connectivity assumption cannot be made (i.e., in the case of adjacencies for a unicast router), then some form of "local" key server must be available for each group. Given that the listening routers are never more than one hop away from the speaking router, the speaking router is the obvious place to locate the "local" key server. This approach has the additional advantage that there is no need to duplicate the local key server for reliability, since if the local key server is down, it is very likely that the speaking router is also down.

We therefore introduce the idea of a *Domain Key Server* (DKS) and a *Local Key Server* (LKS). The DKS is the KS identified in [16]. Each LKS is co-located with the corresponding speaking router, and serves as a key server when the DKS is unreachable. See Figure 2 for the details.

This introduces three additional interfaces. Two of these are "on the wire", one between the DKS and the LKS, and the other between the LKS and the near neighbors. The third interface is within the speaking router, and would likely be implemented as an application programming interface.

The information exchanged on the lower two interfaces would be similar to what is passed within the existing Multicast Group Security Architecture. Therefore, GDOI can be used in this case. A new protocol must be developed for the DKS-LKS communication, to ensure security of the exchanges and consistency of the information presented.

D. Operational issues

The proposed infrastructure is managed centrally, so it is convenient for the system administrator(s). Given that the adjacency policy can be represented per interface, it is possible to deploy incrementally. Note that the implementation of secured communication is achieved entirely by manipulating the GSPD and the SAD at the level of IPsec. There is no need to alter any of the packet formats or procedures of the routing protocol itself. Thus, the protected routing software never sees the packets from an intruder.

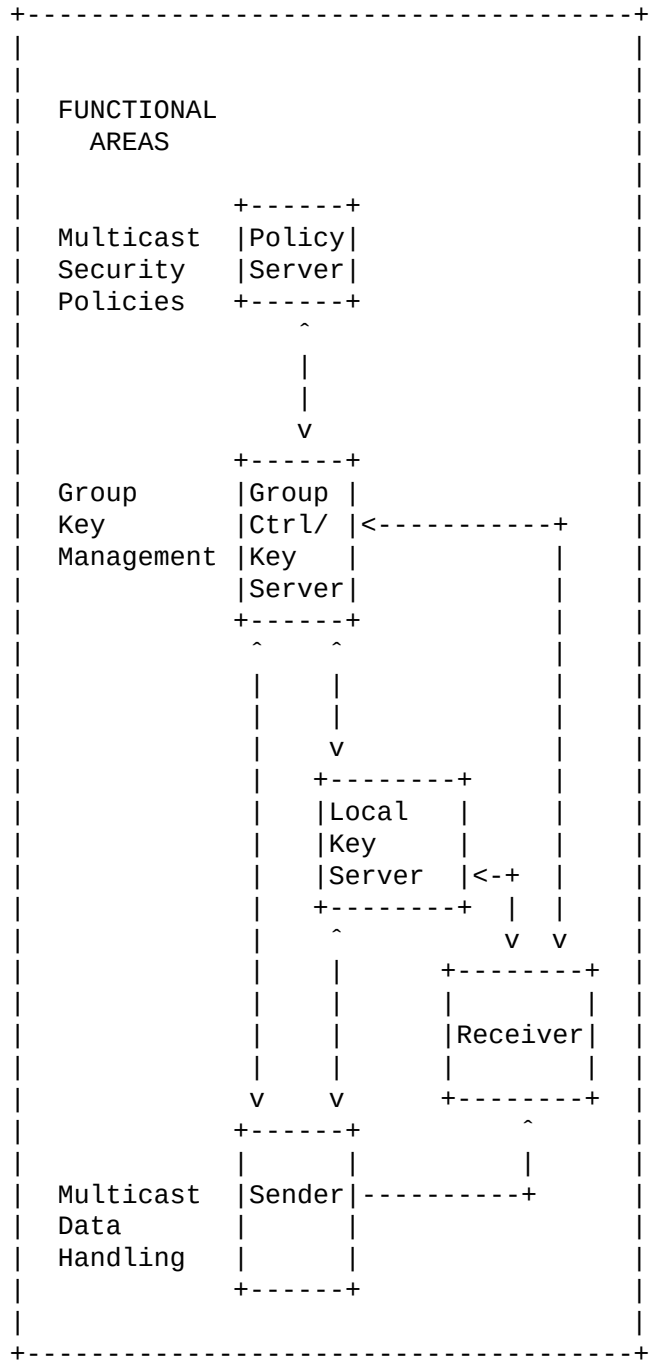

Fig. 2. Proposed Multicast Security Reference Framework

VI. Solution Operation

Similar to the operation of GDOI, it is expected that a system operating in conformance with the proposed architecture will have two phases. The first phase will provide a Security Association between the DKS and each router, to be used to pull and push keying material between the DKS and the individual routers. Using this SA, each router will then acquire the policies concerning those routers with which it is permitted to share information, and the parameters for that communication. In particular, it will acquire the material that permits it to identify a particular neighbor as being legitimate.

If the router becomes partitioned from the DKS, or during the initialization period after a power failure, a router can use its knowledge of those neighbors that were legitimate prior to the repartitioning or failure to execute the necessary phase 1 steps between itself and its neighbors. Once this has completed, it can restore and update the current set of Security Associations, based on information provided by the LKS instances on each of its neighbors, without participating in a storm of requests to the DKS. (Use of this feature clearly requires that the router have access to some form of information retention across the re-boot.) Note that this lowering of expected traffic during recovery from a major power outage represents a strong reason to make use of the LKS model, even for the case where the connectivity assumption can be made.

In a particular router, part of the retained policy material will specify how and when to attempt re-connection with the DKS, to learn about changes to the set of legitimate neighbors.

When deploying this architecture to a sub-region of an administrative region, the fact that security can be specified on a per-interface basis means that it is possible to prepare the adjacency information in the central group controller, and then instruct the routers to access the DKS for new parameters.

VII. Conclusion and Future Work

There is a growing need to verify the authenticity and the integrity of the information on which routing decisions are made. While there are standard solutions for group communications that are suitable for large groups, they are not well-suited for managing the large number of small groups that are found in a typical routing domain.

We have proposed a novel variation on the standard architecture, which addresses the requirements for managing the keying material needed to ensure the security of the routing control infrastructure. This proposal permits restricting near-neighbor communication to legitimate neighbors. It is centrally managed, but completely automatic in its operation, so it has the potential to be convenient for system administrators, while offering considerably enhanced security. The design is capable of being incrementally deployed, and is tolerant of interruptions in the operation of the central manager.

Our next step will be to design, and then formally model, the new protocol that is required for the DKS-LKS interface. This formal model will be based on an existing formal model of GDOI [20].

Acknowledgments

J. W. Atwood acknowledges the support of the Natural Sciences and Engineering Research Council of Canada, through its Discovery Grants Program.

References

[1] G. Lebovitz, "Roadmap for cryptographic authentication of routing protocol packets on the wire," Internet-draft, Internet Engineering Task Force, Mar. 2009.

[2] R. Coltun, D. Ferguson, and J. Moy, "OSPF for IPv6," RFC 2740, Internet Engineering Task Force, Dec. 1999.

[3] "A border gateway protocol 4 (BGP-4)," RFC 4271, Internet Engineering Task Force, Jan. 2006.

[4] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "Protocol independent multicast-sparse mode (PIM-SM): protocol specification (revised)," Request for Comments 4601, Internet Engineering Task Force, Aug. 2006.

[5] M. Gupta and N. Melam, "Authentication/confidentiality for OSPFv3," RFC 4552, Internet Engineering Task Force, June 2006.

[6] W. Atwood, S. Islam, and M. Siami, "Authentication and confidentiality in PIM-SM link-local messages," Internet-draft, Internet Engineering Task Force, Apr. 2009.

[7] "Secure inter-domain routing (SIDR) working group," http://www.ietf.org/html.charters/sidr-charter.html (accessed: 2009 July 10).

[8] S. Kent and K. Seo, "Security architecture for the internet protocol," RFC 4301, Internet Engineering Task Force, Dec. 2005.

[9] S. Kent, "IP authentication header," RFC 4302, Internet Engineering Task Force, Dec. 2005.

[10] S. Kent, "IP encapsulating security payload (ESP)," RFC 4303, Internet Engineering Task Force, Dec. 2005.

[11] "Internet key exchange (IKEv2) protocol," RFC 4306, Internet Engineering Task Force, Dec. 2005.

[12] D. Black and D. McGrew, "Using authenticated encryption algorithms with the encrypted payload of the internet key exchange version 2 (IKEv2) protocol," RFC 5282, Internet Engineering Task Force, Aug. 2008.

[13] V. Manral, "Cryptographic algorithm implementation requirements for encapsulating security payload (ESP) and authentication header (AH)," RFC 4835, Internet Engineering Task Force, Apr. 2007.

[14] J. Schiller, "Cryptographic algorithms for use in the internet key exchange version 2 (IKEv2)," RFC 4307, Internet Engineering Task Force, Dec. 2005.

[15] B. Weis, G. Gross, and D. Ignjatic, "Multicast extensions to the security architecture for the internet protocol," RFC 5374, Internet Engineering Task Force, Nov. 2008.

[16] T. Hardjono and B. Weis, "The multicast group security architecture," RFC 3740, Internet Engineering Task Force, Mar. 2004.

[17] Mark Baugher, Brian Weis, Thomas Hardjono, and Hugh Harney, "The group domain of interpretation," RFC 3547, Internet Engineering Task Force, July 2003.

[18] H. Harney, A. Colegrove, and G. Gross, "GSAKMP: Group secure association key management protocol," RFC 4535, Internet Engineering Task Force, June 2006.

[19] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, "MIKEY: multimedia internet KEYing," RFC 3830, Internet Engineering Task Force, Aug. 2004.

[20] Salekul Islam and J. William Atwood, "Sender access and data distribution control for inter-domain multicast groups," *submitted to Computer Networks*, June 2009.