Network Working Group
Request for Comments: 2154
Category: Experimental

S. Murphy
M. Badger
B. Wellington
Trusted Information Systems

June 1997

OSPF with Digital Signatures

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. This memo does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Abstract

This memo describes the extensions to OSPF required to add digital signature authentication to Link State data, and to provide a certification mechanism for router data. Added LSA processing and key management is detailed. A method for migration from, or co-existence with, standard OSPF V2 is described.

Table of Contents

1. Acknowledgements

The idea of signing routing information is not new. Foremost, of course, there is the design that Radia Perlman reported in her thesis [4] and in her book [5] for signing link state information and for distribution of the public keys used in the signing. IDPR [7] also recommends the use of public key based signatures of link state information. Kumar and Crowcroft [2] discuss the use of secret and public key authentication of inter-domain routing protocols. Finn [1] discusses the use of secret and public key authentication of several different routing protocols. The design reported here is closest to that reported in [4] and [7]. It should be noted that [4] also presents techniques for protecting the forwarding of data packets, a topic that is not considered here, as we consider it not within the scope of the OSPF working group.

The authors would also like to acknowledge many fruitful discussions with many members of the OSPF working group, particularly Fred Baker of Cisco Systems, Dennis Ferguson of MCI Telecommunications Corp., John Moy of Cascade Communications Corp., Curtis Villamizar of ANS, Inc., and Rob Coltun of FORE Systems.

2. Introduction

It is well recognized that there is a need for greater security in routing protocols. OSPF currently provides "simple password" authentication where the password travels "in the clear", and there is work in progress[11] to provide keyed MD5 authentication for OSPF protocol packets between neighbors. The simple password authentication is vulnerable because any listener can discover and use the password. Keyed MD5 authentication is very useful for protection of protocol packets passed between neighbors, but does not address authentication of routing data that is flooded from source to eventual destination, through routers which may themselves be faulty or subverted.

The basic idea of this proposal is to add digital signatures to OSPF LSA data, distribute certified router information and keys, and use a neighbor-to-neighbor authentication algorithm (like keyed MD5) to protect local protocol exchanges. The content of a Hello packet, Link State Request, Link State Update, or Database Description will be protected by the neighbor-to-neighbor algorithm. The LSAs that are being flooded inside the Link State Update packets are individually protected by a digital signature. Each LSA will be signed by the originator of that information and the signature will stay with the data in its travels via OSPF flooding. This will provide end-to-end integrity and authentication for LSA data. The digital signature attached to an LSA by the source router provides assurance that the data comes from the advertising router. It will also ensure that the data has not been modified by some other router in the course of flooding. In the case where incorrect routing data is originated by a faulty router, the signature will identify the source of the problem.

Digital signatures are implemented using public key cryptography. There are some good books on the subject of cryptography [6], but the high level view of how this design uses public key cryptography is as follows: Each router has a pair of keys, a public key and a private key. The private key is used to generate a unique signature of a block of data (in this case, the LSA). Each router signs its LSAs by first running a one-way hash algorithm (like MD5 or SHA) on the data, and then using its private key to sign the digest. The signature of an LSA is appended to the LSA. The public key can be used by any other router to verify the signature. The private key must be kept secret by one router and the public key must be distributed to all the routers that will receive link state information from the signer. The distribution is accomplished by creating a new LSA, the Public Key LSA (PKLSA), and distributing it via the standard OSPF flooding procedure. Flooding will ensure that a router public key is sent everywhere that the router's signed LSAs are sent.

Any router can send out a public key and claim to be a given router, so the public key itself provides no assurance of the actual identity of the sender. This assurance must be provided by a Trusted Entity. The Trusted Entity (TE) is a system that generates certificates for routers. A certificate is a packet of information about a router that identifies the router and supplies a public key. Certified router information will include the router id, its role, the address ranges that the router may advertise, a timestamp and the router's public key. The certificate is signed by the TE. Each router must be configured with a certificate and a TE public key to use in verifying other routers' certificates. A router PKLSA contains the certificate for that router. A router receiving a PKLSA verifies the certificate using the TE public key, and then verifies the whole LSA using the

router public key contained in the certificate. Successful verification provides assurance that the PKLSA is from the correct router, and that it has not been altered by any other router in the flood path.

OSPF with Digital Signatures is backward compatible with standard OSPF V2 in a limited way. Within an AS there may be "signed" areas and "unsigned" areas. The behavior of a mixed AS is discussed in section 5.

Digital signatures for OSPF LSAs can be implemented with the following major functions:

(1) Support for a digital signature algorithm

(2) Support for a signed version of all routing information LSAs

(3) Support for a new LSA: Router Public Key LSA (PKLSA)

(4) A mechanism for key certification and certificate distribution

(5) Extra configuration data (detail in section 7):

Trusted Entity (TE) information and key(s)
Router certification data and key
Area environment flag (signed/unsigned)
Timing intervals

An implementation of this design exists, based on the OSPF in Gated version 3.5Beta3. This implementation is available for use/experimentation. Please contact the authors for information.

3. LSA Processing

3.1. Signed LSA

A signed LSA contains the standard OSPF V2 header and data plus key identification information, a signature length and a signature. The top bit of the LS type field is set to indicate the presence of a signature. The signature covers the LSA header (starting with the options field), the LSA data, and the key identification information and the signature length that must be appended to the LSA data. There are two exceptions to this coverage: first, an LSA created with age=MaxAge has a signature that begins with the age field (see section on maxage); second, the LSA header checksum is set to zero for the generation of the signature. To assist in parsing the message, the key id information and the signature length fields are placed at the end of the LSA, following the signature. However, the

message must be signed and verified with these fields immediately appended to the LSA data. This can be accomplished either by doing the sign and verify "in parts" (allowed by RSAREF), or by storing the LSA data with appended fields and the LSA signature separately in the link state database (LSDB).

When a signed LSA is received, the signature can be verified using the public key of the advertising router contained in the advertising router's PKLSA. If the signature verifies, then the signed LSA is stored for use in routing calculations. If the signature verification fails, the LSA must be discarded. If the identified key is not available (in a PKLSA from the advertising router), then the signed LSA must be stored for a period of time defined by the configurable MAX_TRANSIT_DELAY interval. If the key arrives within this interval, the LSA will be processed then. If the key does not arrive within this interval, the LSA will be discarded. This delay period prevents loss of routing information due to LSAs arriving prior to their associated PKLSAs (which should not normally be the case, but could happen).

If the LSA is a Router Links LSA, the router's advertised links must be checked against the allowed address ranges stored in the PKLSA for the advertising router. All network links (link types 2 and 3) must have an IP address that fits in one of the ranges defined by the list of address ranges in the PKLSA (format 7.2). If there is a link that does not fit into one of these ranges, then an error must be logged and the LSA must be discarded. Careful subnetting and corresponding ranges can provide very tight control on what is advertised. A much less restrictive, but still useful, level of control can be obtained by defining allowed address ranges for an area, so that all routers in an area could be configured with the same set. To trivially satisfy this checking, one range with a zero address and mask can be defined that contains all IP addresses.

Link State Acknowledgements must be sent for all LSAs that are discarded due to verification failures, that are stored waiting for keys, and that are discarded because they are advertising a link that they are not allowed to advertise.

3.2. Router Public Key LSA (PKLSA)

A Router Public Key LSA (PKLSA) is sent in the same manner as all other LSAs. This LSA contains the router's public key and identifying information that has been certified by a Trusted Entity. The router public key is used to verify signatures produced by this router. There is only one PKLSA stored per router in the LSDB for an area, so the Router Id and LS type can be used to retrieve a given PKLSA. The Router Id is stored in the PKLSA Link State Id field to

use in retrieving the PKLSA. Identification information in the certified data (TE Id, Rtr Key Id) can be used to uniquely identify the current router key (section 7.2).

To assist in parsing the message, the router signature length and the certification length fields are at the end of the LSA, following the signature. The message must be signed and verified with these fields immediately appended to the LSA data. The router signature of the PKLSA is verified in the same manner as other signed LSAs. In addition, the certification must be verified using the referenced TE public key. If either verification fails, for any reason, the PKLSA is discarded.

A successfully verified PKLSA is stored for use in verifying signed LSAs from the advertising router. For every router that this router is in contact with, there may be one PKLSA stored at any given time. Each PKLSA is uniquely identified by the values (TE Id, Rtr Key Id) in the certified data (format in 7.2). When a PKLSA arrives for a given router, and there is already a PKLSA stored for that router, the PKLSA with the most recent "Create Time" is the one kept.

Whenever groups of LSAs are sent by a router (as when synchronizing databases or sending updates), the PKLSAs must be sent/requested before other LSAs to minimize the time spent processing LSAs that arrive prior to their associated keys. The PKLSA is sent at intervals like all other LSAs, and it is sent immediately if a router obtains a new key to distribute. A PKLSA is sent via OSPF flooding within an OSPF area. PKLSAs are not flooded outside an area with the exception of an Autonomous System Border Router's PKLSAs which must be flooded wherever AS external LSAs are flooded. The decision to flood or not flood can be implemented by checking the router role (Rtr, ABR, ASBR, ABR-ASBR) stored in the certified part of the PKLSA.

A router may flush its keys from routing tables by flooding a PKLSA for that key with age=MaxAge. This is called premature aging of the PKLSA. A key can also be removed from routing tables (superseded) by a PKLSA from the same router, containing a valid certificate for a new key with a more recent Create Time. If a key is superseded by a more recent key it is not necessary to flush the old key with a "MaxAge" PKLSA.

When a new key is received, the LSAs stored in the LSDB that are signed with the old key must be replaced within MAX_TRANSIT_DELAY. if the sending router is working properly. This is because a router distributing a new key sends all of its self-originated LSAs signed with the new key immediately after sending the new PKLSA. (See section 4.4 on Router Key Replacement). To ensure that data signed with an old (possibly subverted) key does not persist in the LSDB in

error, all LSAs signed with a flushed or superseded key are aged to within MAX_TRANSIT_DELAY of MaxAge. This should allow time for the new LSAs signed with the new key to arrive. If new LSAs do not arrive, or if the key has been flushed and not replaced, then the old LSA data will disappear from the LSDB in a timely fashion.

Link State Acknowledgements must be sent for PKLSAs that are discarded due to verification failures or because the PKLSA was less recent than the one already stored.

3.3. MaxAge Processing

The age field in the OSPF LSA header is used to keep track of how long a given LSA has been in the system. When the age field reaches MaxAge, a router stops using the LSA for routing, and it floods the MaxAge LSA to make sure that all routers stop using this LSA. In the normal course of the OSPF protocol, an LSA is always replaced by an updated version before the age reaches MaxAge, unless the advertising router fails, or changes in the AS have made the routing information in the LSA inaccurate. An LSA with age=MaxAge is either:

(1) being intentionally flushed from the AS by the advertising router because the information in it is no longer accurate, or

(2) an orphan LSA that has aged to MaxAge because its originating router has not refreshed it at the normal refresh intervals.

The age field cannot generally be included in the signature, because it must be updated by routers other than the originating router. For the same reason, the age field is not included in the checksum computation. The age field must be protected, because if a faulty router started to age out other router's LSAs, it would effectively deny service to those other routers.

To protect the age field, the signature must include the age field if and only if the originating router creates an LSA with age=MaxAge. Verification of the signature on a signed LSA must include the age field if and only if the age field value is MaxAge. In this manner, the originating router can flush an LSA, but other routers cannot. An LSA that ages to MaxAge in the LSDB of any router is still discarded by that router, but it is not synchronously flushed from the AS.

An LSA will be removed from a router's Link State Database in one of two ways: 1) the router receives a version of the LSA with the age field set to MaxAge and a valid signature that covers the age field, or 2) the LSA incrementally reaches MaxAge while it is stored by the router.

If a standard OSPF V2 router goes down, an LSA from that router will age in the LSDBs of each remaining router until it reaches MaxAge somewhere. As soon as it reaches MaxAge in some router's LSDB it is flooded, and this causes it to be flushed from the AS in a synchronized fashion. If router running OSPF with digital signatures goes down, its signed LSAs will be aged out by each remaining router individually. This will slow database convergence but the databases will still converge, and a fairly obvious security hole will be closed.

4. Key Management

4.1. Identifying Keys

4.1.1. Identifying Router Keys and PKLSAs

A router key is identified by the Router Id, and the identifiers associated with the particular key in its certificate: TE Id and Router Key Id. All three of these values are stored in a PKLSA (format in 7.1). The Router Id is the standard LSA header Advertising Router. The (TE Id, Rtr Key Id) are stored in the PKLSA certified data. The TE Id is a number assigned to a Trusted Entity that must uniquely identify one TE in the AS. The TE Id in a certificate identifies the TE that produced the certificate. The Rtr Key Id is associated with a key by the Trusted Entity that produced the certificate. The Trusted Entity must produce a stream of Rtr Key Ids for one router such that the router will not re-use a key id until all references to the last key having that id are gone from the AS. If a key is re-played, or re-used too soon, the Create Time in the key certification will determine which key is current. Rtr Key Ids do not have to be sequential.

4.1.2. Identifying TE Public Keys

Each TE public key has an associated TE Id, TE Key Id. The combination of (TE Id, TE Key Id) uniquely identifies one TE public key in the AS. The TE Id is a number assigned to a Trusted Entity

that uniquely identifies one TE in the AS. The TE Key Id must identify one particular key for a TE at any given time. The TE Key Id distinguishes between a new key and an old key for the same TE. The TE Key Id also differentiates between keys for different signature algorithms if one TE serves multiple algorithms. Each TE can have at most one current key per signature algorithm.

There can be multiple TE keys stored on each router. A TE public key is used to verify the certificates issued by other routers, and in an AS with several TEs, any given router may need several TE public keys. TE Key Ids do not have to be used sequentially, and they can be re-used. There is no timestamp for TE keys because these are not certified.

It is the responsibility of Configuration Management to ensure that TE Key Ids are not re-used before all references to a previously used key with the same (TE Id, TE Key Id) are gone from the AS, that a given (TE Id, TE Key Id) on one router identifies the same key as it does on any other router, and that the rules for TE Key Replacement (section 4.5) are followed.

4.1.3. Key to use for Signing

A router is configured with a pair of keys. The private key is protected from disclosure and is used for signing. The public key is flooded in a PKLSA and is used for verifying signatures. A router may have one key per area to use for signing at any given time. A router may use the same key for several or all areas.

4.1.4. Key to use for Verification

There are three uses of signature verification in this design:

(1) The signature in a signed LSA (format in 7.3) can be verified with the public key distributed by the advertising router in a Public Key LSA. A signed LSA contains the (TE Id, Rtr Key Id) of the key used to sign it. The signed LSA's Advertising Router Id is used to retrieve the router's PKLSA , and the (TE Id, Rtr Key Id) indicates if the router key in the PKLSA is the same as the one used to generate the signature.

(2) The router's signature in a PKLSA (format in 7.1) is verified with the public key contained in that PKLSA.

(3) The PKLSA contains data certified with a signature generated by a TE. The PKLSA certified data contains the (TE Id, TE Key Id) for the TE key that can be used to verify the certificate (format in 7.2). TE public keys must be configured on each router.

4.2. Trusted Entity (TE) Requirements

This design does not specify how the Trusted Entity (TE) must be implemented, where it must reside, or how it must communicate with routers. There are several very different possible approaches to the implementation of a Trusted Entity (e.g., an offline system with distribution of keys by floppy or secure e-mail, an online automated key distribution center, etc.) This design does mandate certain requirements for what a Trusted Entity must do. A Trusted Entity must generate a certificate for each signing router that contains individualized information about that router (format in 7.2) and is signed with the Trusted Entity private key. The Trusted Entity must have a unique TE Id for itself, it must create a Rtr Key Id for each router key that is unique for the given Router for this TE at this time, and it must timestamp certificates with a Create Time that is consistent for itself and for any other Trusted Entities operating in the AS. Note: routers do not have to be time-synched, but TEs do. Create Time is used by routers as a relative measure to determine which key is more recent.

The TE Public key, TE Id, TE Key Id and Signature Algorithm must be made available to each router processing certificates from this TE.

A TE can theoretically create certificates for more than one signature algorithm. The TE key and the router public key certified do not have to be of the same signature algorithm.

There can be more than one TE in an AS but the TE Id must identify a unique TE.

4.3. Scope for Keys and Signature Algorithms

The concept of "scope" relates to Router Keys, TE Keys, and Signature Algorithms.

(1) The scope of a PKLSA and therefore a router key, is defined to be the set of routers that will receive and store that PKLSA in the course of OSPF flooding. A router produces a PKLSA for each attached area. In a router with more than one area, the PKLSAs for each area may match, or each may contain a different key. The scope of PKLSA for an internal router is all the routers in that area. An ABR has multiple PKLSAs, each having a scope of

one attached area. The scope of an ASBR's PKLSA is the same as the scope of the ASBRs ASEs - all the routers in all the non-stub areas in the AS. An ASBR that is an ABR produces multiple PKLSAs that each have a scope of all the routers in all the non-stub areas in the AS. (This last case results in some situations that require special management - section 6)

(2) The scope of a TE key is defined to be the set of routers that are configured with this key. If a system is configured properly, then a TE public key will be configured on all the routers that will receive PKLSAs certified by that TE key. The minimum scope for a TE key is an area. If one router distributes a key certified with a given TE key, then all the routers in the area must be able toverify the certificate. A TE Key certifying an ASBRs key must have a scope of all non-stub areas in the AS. If the TE key is not on some router that receives PKLSAs certified by that TE key, then those PKLSAs and all the LSAs that require them will be discarded. A TE key gets to all the routers in its scope via out-of-band configuration.

(3) The scope of a signature algorithm is defined to be the set of routers that are capable of verifying the given algorithm's signatures. The minimum scope for a signature algorithm is an area. All routers in an area must be able to verify any signature algorithm used for signing by any router in the area. The algorithm used to certify an ASBRs key must have a scope of all non-stub areas in the AS if the ASEs are to be accessible everywhere (see section 6). If a signature algorithm is not available to verify an LSA, then the LSA must be discarded. If a signature algorithm is not available to verify the certification in a PKLSA, then the PKLSA must be discarded.

4.4. Router Key Replacement

Router keys should be changed periodically, and immediately if a key is found to be compromised. The regular period for changing a key is some locally determined function of the size of the key and the level of security needed.

Each router can have ONE valid key per area at any given time. Restricting the number of keys at a given time to one key per router per area allows key replacement to also serve the purpose of key revocation, without having a revocation list and without routers having synchronized time. Each key for the router/area revokes the last key, provided the "new" key has a more recent Create Time than the last key. The Create Time in each certificate is used to prevent an old key from being reused, but this Create Time is used only for comparing the relative ages of certificates, and does not require the

router to run a time synchronization protocol itself. An ABR can use the same key for all it's attached areas, or it can have a unique key for each area. This allows an AS to be managed by area with each area potentially having a different TE, signature algorithm, key size, and/or key.

When a new key replaces an old key, the router must quickly replace LSAs signed with the old key with LSAs signed with the new key. To change a router key the following steps must be followed:

(1) A valid certificate for the new key must be obtained for the router.

(2) The router builds and sends a new PKLSA with the new certificate.

(3) The router signs each self-originated LSA with the new key and sends them.

When a PKLSA is received:

(1) If the PKLSA's age = MaxAge, remove the PKLSA from the LSDB and age LSAs signed with this key to be MaxAge - MAX_TRANSIT_DELAY, if they were not already older than this. This is a way to get rid of a key that should no longer be used.

(2) If the PKLSA is a refresh LSA for an existing key, update the LSDB.

(3) If the PKLSA contains a different key than the one currently stored for this router, compare the certificate Create Time. If the PKLSA key is less recent, discard it. If the PKLSA key is more recent, install it in the LSDB and remove the old key from the LSDB. If an old key was deleted from the LSDB, age LSAs signed with this key to be MaxAge - MAX_TRANSIT_DELAY, if they were not already older than this.

4.5. Trusted Entity Key Replacement

It is necessary to change a TE public key periodically. It is recommended that the TE public key be relatively large, so that it does not frequently require replacement. A router may store multiple TE public keys. Each key is uniquely identified by TE Id and TE Key Id. TE keys are used to verify certificates received from other routers in their PKLSAs. When a router sends a new certificate signed with a new TE Key, all the routers that receive the PKLSA containing the certificate must have that new TE Key in order to verify, store, and use that PKLSA. Management of TE public keys is done outside the OSPF protocol, and a method is suggested, but not

mandated by this design. Initially all routers must be configured with the TE Keys they will need to verify the certificates they will receive. To prevent use of a (possibly compromised) TE Key, that key must be replaced by a new (possibly null) TE Key having the same TE Id and signature algorithm. A compromised or faulty router can continue using certificates signed with the old TE key, but none of the properly configured routers will be able to verify them.

Changing a TE public key presents a design challenge. When a TE Public Key is changed, all the certificates depending on that key must also change. The router keys in the certificates may or may not be changed at the same time. When the TE key and certificates change, all PKLSAs depending on these must be reissued. In order to verify these new certificates, all routers receiving the new PKLSAs must have the new TE Public Key. So, the TE key replacement must be a synchronized event. Routers are not required to have synchronized clocks. The TE public key may well be distributed to the routers via an out-of-band mechanism (like a smart-card reader or other sneaker-net method). It is not reasonable to require that all the routers obtain the TE public key at the same time. There are probably several methods for meeting these requirements. The method tested in our implementation is as follows:

(1) Define a period of time needed to get the new TE key on all routers. This could be minutes, hours, even days depending on how the distribution is accomplished. This time period is a configuration value for each router (TE_KEY_DIST_INT) and must be the same for all routers sharing a TE.

(2) Install a new TE key and associated certificates (if there are any) on each router. Signal the router code when the new TE key is available to be accessed.

(3) The router sets a timer for the TE_KEY_DIST_INT. The router sets a flag indicating the presence of a new TE key.

(4) For each router, if the timer goes off:

```
Access the new TE key.
If there are new certificates, build and send a new PKLSA.
Age all PKLSAs in the LSDB certified by the old TE Key
        to MaxAge - MAX_TRANSIT_DELAY.
```

(5) For each router, if a PKLSA certified by a new TE key comes in before the timer goes off:

```
If the new TE key cannot be accessed, discard the PKLSA and
        log an ERROR.
Access the new TE key.
Process the received PKLSA.
If there are new certificates, build and send a new PKLSA.
Age all PKLSAs in the LSDB certified by the old TE key
        to MaxAge - MAX_TRANSIT_DELAY.
```

The effect of this method is that it takes a predetermined interval of time to change the TE public key. That interval is the amount of time from the installation of the new TE key on the FIRST router installed, until the time that router reads the key in. By the time the first router reads the key in, all other routers should have the new key. If some router does not get the new TE key in time, it will be unable to verify all the new PKLSAs that are received. It will log error messages and route data based on it's old database until those LSAs time out. The simple way to fix a router in this error condition is to load the new TE key and restart the router. If this error is expected to occur, and restarting the router is not acceptable, then some special purpose code will be needed to read in the TE key after it has been otherwise distributed, and do database synchronization to catch up with the other routers.

The group of routers that need the new TE key are all the routers in the scope of that Trusted Entity.

4.6. Flexible Cryptographic Environments

It is likely that an AS will have one cryptographic environment in use throughout the AS, with one trusted entity, one signature algorithm in use, and one key in use per router. To allow those cases where this is not true, multiple signature algorithms, multiple trusted entities, and multiple keys per router are allowed.

4.6.1. Multiple Signature Algorithms

It is possible to support multiple signature algorithms. Each router and TE key has a signature algorithm associated with it. All routers sending a key with a given algorithm must be capable of generating signatures of that kind, and all routers receiving keys with a given algorithm must be able to verify the signatures. If a router receives an LSA signed with a signature algorithm that it does not support, the LSA must be discarded. LSAs that cannot be verified by a router are not flooded by that router. When using multiple signature algorithms, the scope of each algorithm must be determined

(see section 4.3), and routers must be configured with support for these algorithms accordingly.

If an Area supports two signature algorithms and is to have full connectivity, some routers may sign with algorithm A and others with algorithm B, but all routers in the area must be able to verify signatures for A and B. In an AS that is divided into areas, it is possible for each area to have a different signature algorithm. The ABR connecting two areas would have to support both algorithms, but the internal routers in a given area would only have to know one algorithm.

ASBRs present a problem for this sort of division. ASEs flood throughout the non-stub areas of an AS. Any router that cannot verify an ASE will discard it without flooding. So, to have access to an ASE, a router, and all the routers in the flooding path, must support the algorithm used by the ASBR. One way around these difficulties is to have a lowest-common-denominator algorithm that is used for signing by all ASBRs and is supported for verification throughout the AS in addition to other algorithms used. Another approach is to place ASBRs on the backbone, and configure all areas using a signature algorithm different from the ASBR to have a default route to the backbone. A combined approach will allow an ASBR to be in a non-backbone area if it uses a signature algorithm supported on the backbone, and the areas using different signature algorithms are configured with a default to the backbone. There are special limitations in the case of a router that is an ABR and also an ASBR: see section 6.

There is currently only one signature algorithm (RSA_MD5) defined for use by this design. The RSA algorithm is defined in PKCS #1 [9] and the signature and key formats used by this design are defined in RFC2065 [10].

4.6.2. Multiple Trusted Entities

It is possible to have multiple Trusted Entities in an AS. Each TE has a unique TE identifier. Every router receiving PKLSAs certified by a given TE must have that TE's public key. If a router receives a PKLSA certified by a TE for which it does not have a public key, the PKLSA must be discarded. When using multiple TEs, the scope of each TE must be determined (see section 4.3), and routers in this scope must be configured with the TE key.

4.6.3. Multiple Keys for One Router

An ABR may have one key for each attached area. These keys may differ in size, algorithm and/or certifying TE. Generally, each key will have a "scope" of the attached area, and there will be no conflict between keys.

There are special limitations in the case of a router that is an ABR and also an ASBR: see section 6.

5. Compatibility with Standard OSPF V2

OSPF with Digital Signatures is compatible with standard OSPF V2 in an autonomous system. Within an AS, there may be "signed" areas and "unsigned" areas. There will never be both signed and unsigned LSAs used in any one area. Each area will have an environment flag indicating whether it is "signed" or "unsigned". The environment flag is a per area configuration value for the router. The signed areas must contain all routers running OSPF with Digital Signatures, and the unsigned areas contain routers running standard OSPF V2 code (or OSPF with Digital Signatures with all areas set to be unsigned). An area border router connecting a signed to an unsigned area must be running OSPF with Digital Signatures with one area set to be unsigned.

In order to arrange this limited compatibility, a router running OSPF with Digital Signatures must be able to process both signed and unsigned LSAs. The only router that will actually be processing both kinds of LSAs is an Area Border Router connecting a signed area to an unsigned area. An ABR connecting a signed to an unsigned area will generate signed summaries for one area and unsigned summaries for the other. An ABR must not flood signed LSAs into unsigned areas. An ABR must not flood unsigned LSAs into signed areas. This will result in AS External LSAs being dropped if they reach an area that has a different environment from the one in which they were created. There are special limitations in the case of a router that is an ABR and also an ASBR: see section 6.

Complete connectivity is provided within the AS, because of the summarization provided by ABRs connecting signed and unsigned areas. There are limitations on connectivity to AS external routes in an AS with a mixture of signed and unsigned areas, depending on the location of AS border routers. An ASBR in a signed area will generate signed ASE LSAs. These LSAs will be flooded to every contiguously connected signed area. The connected signed areas are the "scope" of these ASEs. A host located in an area that is not in this scope, will not have connectivity to these external routes. An ASBR in an unsigned area will generate unsigned ASE LSAs. These LSAs

will have a scope of all the contiguously connected unsigned areas, and will be available to hosts in this scope. To arrange complete connectivity to an ASE route in an AS with signed and unsigned areas:

(1) Place the ASBR on the backbone.

(2) Signed Backbone: have some ABR in each unsigned area advertise a default route to the backbone.

(3) Unsigned Backbone: have some ABR in each signed area advertise a default route to the backbone.

Given this design for a mixed AS, routing is available throughout the AS, but the authentication and integrity provided by this design will be effective only for routes that are inside a signed area, or traverse only signed areas. There is no mechanism for a data packet to state a preference for signed routes. The basic rules of the OSPF protocol ensure that intra-area routes are preferred to inter-area routes, that routes within the AS are preferred to AS external routes, and that inter-area routes go from area1->backbone->area2. OSPF does not allow looping, or routes of the form area1->area2->area3. Because of these properties of OSFP routing, an AS can contain signed and unsigned areas, and achieve a predictable level of authentication.

6. Special Considerations/Restrictions for the ABR-ASBR

There are special restrictions and configuration considerations for a router running OSPF with Digital Signatures that is both an Area Border Router and an Autonomous System Border Router. An ASBR produces AS external LSAs that are flooded throughout the non-stub areas of the AS. An ABR that is generating digital signatures may be using a different key, certifying Trusted Entity, or signature algorithm for each of its attached areas, or it might be signing in some areas and not in others.

An ABR/ASBR with no restrictions on its configuration could produce multiple versions of an ASE that would all be flooded throughout the non-stub areas of the AS. The results of this production of multiple versions of LSAs would be detrimental to performance, and could produce unpredictable routing behavior.

The PKLSA of an ASBR is also flooded throughout the non-stub areas of the AS, and in the case of an ABR/ASBR there could be multiple, distinct PKLSAs for a given router, one per attached area, all being flooded throughout the AS. If two distinct PKLSAs from one ABR/ASBR router were present in one area, the key with the most recent create time would be stored, and all LSAs signed with a less recent key would be unverifiable.

The simplest way to deal with this problem, and the method recommended by this document, is the following:

If an ASBR must also be an ABR, then the security configuration (key, signature algorithm, certifying Trusted Entity, environment = signed/unsigned) for all attached areas must be the same. This way the PKLSA and the ASEs produced for each area match, and there is no proliferation of versions of LSAs.

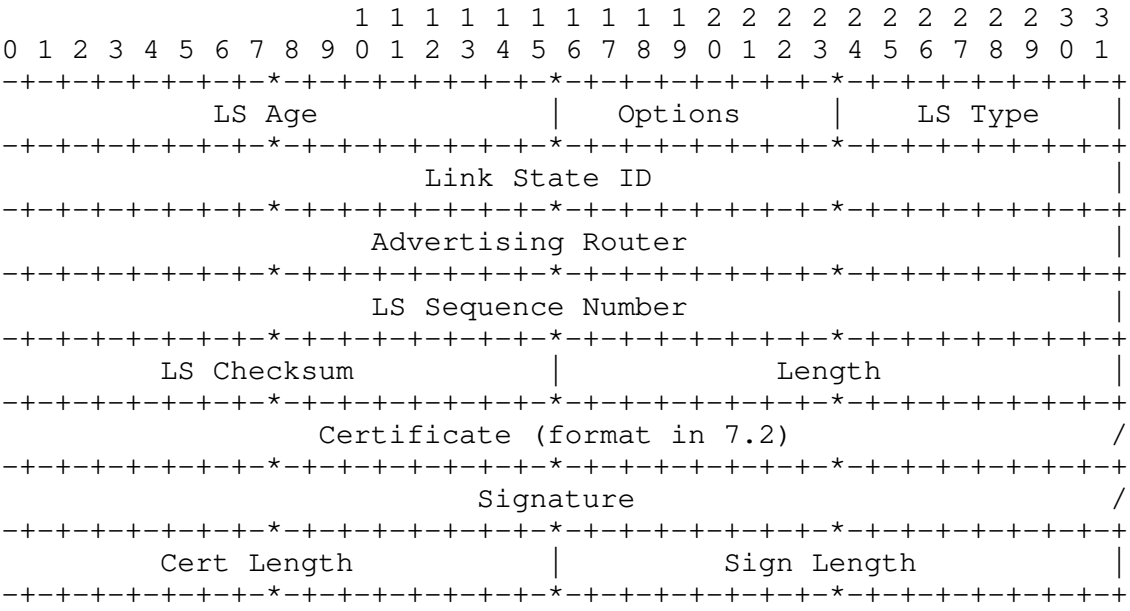
7. LSA formats

7.1. Router Public Key LSA (PKLSA)

This LSA is the vehicle for distribution of a router public key. The PKLSA is sent by one router, and stored by all the other routers in the flooding scope. The PKLSA contains the public key that other routers will use to verify the signatures created by this router. A Router PKLSA will be communicated in the usual database exchange and via flooding mechanisms. The regular period for sending this LSA is LSRefreshTime. The Router PKLSA will also be sent when there is a new key, or a key to be flushed from the system.

The flooding scope of a PKLSA is the area, except in the case of ASBRs. The flooding scope of an ASBR's PKLSA is the same as that of the ASEs. The "role" of the router (RTR, ABR, ASBR, ABR-ASBR) is stored in the PKLSA inside the certificate, and can be checked during flooding.

ROUTER PUBLIC KEY LSA



LS AGE Defined in OSPF RFC [3].

OPTIONS Defined in OSPF RFC [3].

LS TYPE 16 for Router Public Key LSA.
First bit set to indicate a signed LSA.

LINK STATE ID Contains the Advertising Router Id (see next field).

ADVERTISING ROUTER Defined in OSPF RFC [3].

LS SEQUENCE NUMBER Defined in OSPF RFC [3].

LS CHECKSUM Defined in OSPF RFC [3].
Checksum does not cover the signature.

LENGTH Defined in OSPF RFC [3]. Length does include the Signature field, Cert Length and Sign Length.

CERTIFICATE Format in section 7.2.

SIGNATURE — The advertising router's signature of this LSA. This can be verified using the enclosed Router Public Key. The signature covers the LSA header and message starting with the LSA header options field and ending with the Trusted Entity certification field. For sign and verify, the last two fields (Cert Length and Sign Length) are appended immediately after the Certificate. When complete, the signature is inserted between the Certification and the Cert Length. There are two exceptions to this coverage:

1) If the LSA was generated with an age=MaxAge, then the signature begins with the age field (see section 3.3).

2) The checksum in the LSA Header is set to zero for the computation of the signature.

A pad is added to the end of the signature field to allow the next field to begin on a (4 byte) word boundary.

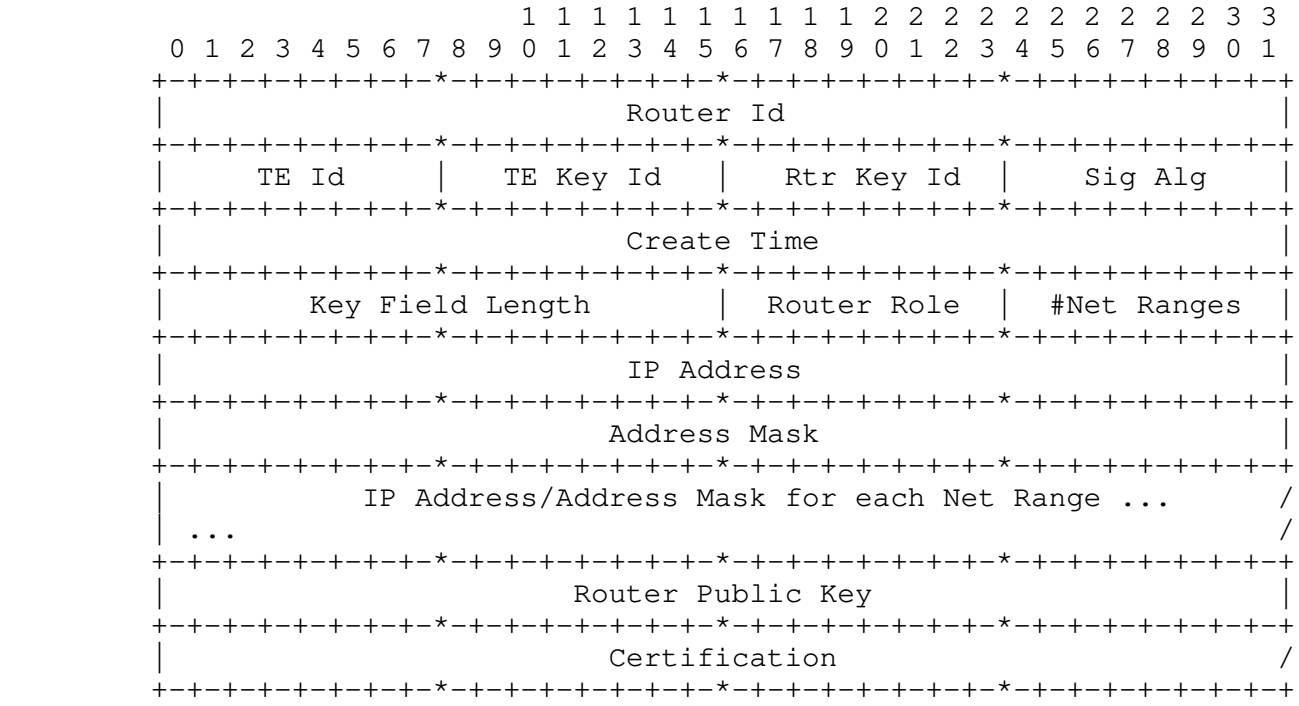
The format used for an RSA-MD5 signature is defined in section 4.1.2 of RFC2065 [10].

CERT LENGTH — The length in bytes of the Certification inside the Certificate.
Does not include pad that may follow Certification.

SIGN LENGTH — The length in bytes of the Signature.
Does not include pad that may follow Signature.

7.2. Router Public Key Certificate

A router public key certificate is a package of data signed by a Trusted Entity. This certificate is included in the router PKLSA and in the router configuration information. To change any of the values in the certificate, a new certificate must be obtained from a TE.



ROUTER ID Advertising Router.

TE ID TE Id must uniquely identify one TE in the AS. A number between 1-250. 0 reserved for null. 251-255 reserved for future needs.

TE KEY ID Must uniquely identify a particular key for a given TE at any given time. A TE Key Id may be re-used after all references to it are gone from the AS. A number between 1-250. 0 reserved for null. 251-255 reserved for future needs.

RTR KEY ID Must be unique for the TE and Router at any given time. The combination of (TE Id, Rtr Id, Rtr Key Id) uniquely identifies a particular router key at a given time. A Rtr Key Id may be re-used after all references to it are gone from the AS. Create Time resolves any conflict that could be caused by replaying old keys. A number between 1-250. 0 reserved for null. 251-255 reserved for future needs.

SIG ALG — The signature algorithm for the Router Public Key. The signature algorithm encompasses the hash algorithm used as well. Currently defined value = RSA-MD5(1). Values 2-252 are available for future definition. Values 0 and 253-255 are reserved. The Sig Alg value is registered with IANA. Future signature algorithms will have to be defined or referenced in this document, and registered with IANA.

CREATE TIME — Timestamp set by the TE. An unsigned number of seconds since the start of January 1, 1970, GMT, ignoring leap seconds. Used to compare two certificates and determine which is more recent. Requires that time synchronization for TEs, but not for routers.

KEY FIELD LENGTH — The length in bytes of the Router Public Key. Does not include pad that may follow Router Public Key field.

ROUTER ROLE — Router (R=1), Area Border Router (ABR=2), Autonomous System Border Router (ASBR=4), ABR and ASBR (ABR-ASBR=6).

#NET RANGES — The number of network ranges that follow. A network range is defined to be an IP Address and an Address Mask. This list of ranges defines the addresses that the Router is permitted to advertise in its Router Links LSA. Valid values are 0-255. If there are 0 ranges the router cannot advertise anything. This is not generally useful. One range with address=0 and mask=0 will allow a router to advertise any address.

IP ADDRESS & ADDRESS MASK — Define a range of addresses that this router may advertise. Each is a 32 bit value. One range with address=0 and mask=0 will allow a router to advertise any address.

ROUTER PUBLIC KEY A key that can be used to verify the signatures produced by this router. The internal format for the Router Public Key is signature algorithm dependent.

A pad is added to the end of the Router Public Key field to allow the next field to begin on a (4 byte) word boundary.

The format used for an RSA-MD5 public key is defined in section 3.5 of RFC2065 [10].

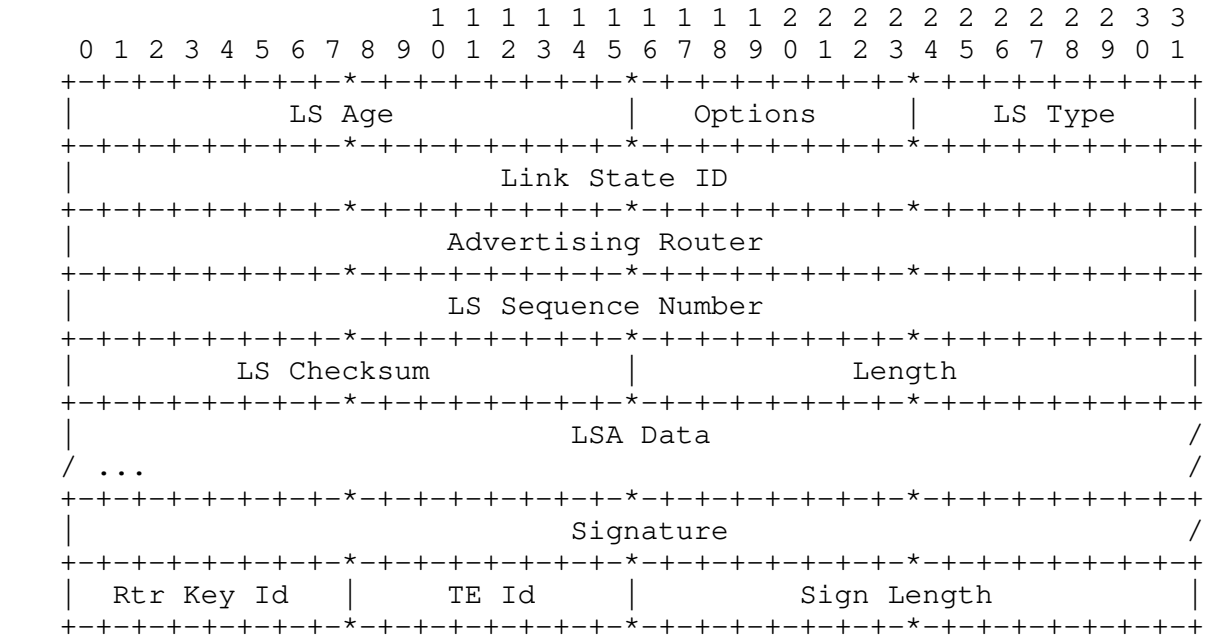
CERTIFICATION The Trusted Entity's signature of the certified data. This signature can be verified with the TE public key identified by TE Id and TE Key Id given in this packet. The length of the certification depends on the key size, and is stored in the PKLSA Cert Length field. A pad is added to the end of the Certification to allow the next field to begin on a (4 byte) word boundary.

The format used for an RSA-MD5 signature is defined in section 4.1.2 of RFC2065 [10].

7.3 Signed LSA

A signed LSA is an OSPF LSA with signature data and a digital signature attached. The first bit of the LSA Type field is set to indicate the presence of a signature. The signature follows the LSA Data. Signature length and id fields are positioned at the end of the signed LSA.

ANY SIGNED LSA



LS AGE Defined in OSPF RFC [3].

OPTIONS Defined in OSPF RFC [3].

LS TYPE Standard LSA Type with the first bit set to indicate the presence of security data and a signature. This creates a new signed LSA type for each existing type.

LINK STATE ID Defined in OSPF RFC [3].

ADVERTISING ROUTER Defined in OSPF RFC [3].

LS SEQUENCE NUMBER Defined in OSPF RFC [3].

LS CHECKSUM Defined in OSPF RFC [3].
Checksum does not cover the signature.

LENGTH Defined in OSPF RFC [3].
Length does include the Signature and security related fields at the end of the LSA.

SIGNATURE — The advertising router's signature of this LSA. The signature covers the LSA header and data starting with the LSA header options field and ending with the Trusted Entity certification field. For sign and verify, the last three fields (Rtr Key Id, TE Id, Sign Length) are appended to the Certificate. When complete, the signature is inserted between the Certification and the Rtr Key Id. There are two exceptions to this coverage:

1) If the LSA was generated with an age=MaxAge, then the signature begins with the age field (see section 3.3).

2) The checksum in the LSA Header is set to zero for the computation & verification of the signature.

A pad is added to the end of the signature to allow the next field to begin on a (4 byte) word boundary.

The format used for an RSA-MD5 signature is defined in section 4.1.2 of RFC2065 [10].

RTR KEY ID — Used to identify the router key used to sign this LSA. The combination of (TE Id, Rtr Id, Rtr Key Id) uniquely identifies a particular router key at a given time, and can be used to look up the PKLSA for the router key needed to verify this Signed LSA. A number between 1-250. 0 reserved for null. 251-255 reserved for future needs.

TE ID — The id of the Trusted Entity that produced the certificate. TE Id must uniquely identify one TE in the AS. A number between 1-250. 0 reserved for null. 251-255 reserved for future needs.

SIGN LENGTH — The length in bytes of the Signature. Does not include pad that may follow Signature.

8. Configuration Information

Trusted Entity Information Set: (one per Trusted Entity used by this router)

- Trusted Entity ID - TE Id
  Identifies the Trusted Entity within the AS (defined in 7.2).
- Trusted Entity Key Id - TE Key Id
  Identifies the particular key for this Trusted Entity (defined in 7.2).
- Trusted Entity Public Key
  A public key for this Trusted Entity.
  The format used for an RSA-MD5 public key is defined in section 3.5 of RFC2065 [10].
- Signature Algorithm < and optional parameters >
  The signature algorithm for the public key (defined in 7.2).

Router Information Set: (at least one for the router)

- Router Private Key
  The router's private key that goes with the public key in the certificate following. The format used for the private key depends on the crypto package used by your implementation. This key is not transmitted as part of this design. Our implementation uses the private key format compatible with RSAREF [9].
- Router Certificate (format in 7.2).

Timing Intervals:

- Trusted Entity Key Distribution Interval (TE_KEY_DIST_INT)
  The period of time, in seconds, needed to get a TE public key installed on all the routers in the TE's scope.
- Maximum Transit Delay (MAX_TRANSIT_DELAY)
  The maximum period of time, in seconds, that it should take for an LSA to reach all the routers in the AS.

Router Information per attached Area:

- Environment flag
  Signed=1, Unsigned=0

9. Remaining Vulnerabilities

Note that with this mechanism, one router can still distribute incorrect data in the information for which it itself is responsible. Consequently, an autonomous system employing digital signatures with this mechanism will not be completely invulnerable to routing

disruptions from a single router. For example, the area border routers and autonomous system border routers will still be able to inject incorrect routing information. Also, any single internal router can be incorrect in the routing information it originates about its own links.

9.1. Area Border Routers

Even with the design proposed here, the area border routers can inject incorrect routing information into their attached areas about the backbone and the other areas in Summary LSA's. They can also inject incorrect routing information into the backbone about their attached area.

Because all the area border routers in one area work from the same database of LSA's received in their common area, it would be possible for the area border routers to corroborate each other. Any area border router for an area could double check the Summary LSA's received over the backbone from other ABR's from the area, and could double check the Summary LSA's flooded through the area from the other area border routers. The other routers in the area or backbone should be warned of a failure of this check. The warning could be a signed message from the area border router detecting the failure, flooded in the usual mechanism.

Another possibility would be that the area border routers in an area could originate multiple sets of Summary LSA's -- one for itself containing its own information and one for each of the area border routers in the area containing the information each of them should originate. Each router in the area or backbone could then determine for itself whether the area border routers agreed. This distribution of information but coordination of processing is in keeping with the paradigm of link state protocols, where information and processing is duplicated in each router.

Both alternatives mean much additional processing and additional message transmission, over and above the additional processing required for signature generation and verification. Because the vulnerability is isolated to a few points in each area, because the source of incorrect information is detectable (in those situations where the incorrect information is spotted) and because the protection is costly, we have not added this protection to this design.

9.2. Internal Routers

The internal routers can be incorrect about information they themselves originate.

A router could announce an incorrect metric for a valid link. There is no way to guard against this, but the damage would be small and localized even if the router is announcing that the link is up when it is down or vice versa.

A router could announce a connection that does not in fact exist. If a router announces a non-existent connection to a transit network, the OSPF Dijkstra computation will not consider the connection without a similar announcement from another router at the other "end". Therefore, no damage would result (above network impact to transmit and store the incorrect information) without the cooperation of another router. A router could also announce a connection to a stub network or a host route that does not exist. The Dijkstra computation can not perform the same check for a similar announcement from the other "end", because no other end exists. This is a vulnerability.

A faulty router announcing a nonexistent connection to a stub network or host could result in the faulty router receiving IP packets bound for that network or host. Unless the faulty router then forwarded the packets to the correct destination by source routing, the failure of packet delivery could expose the incorrect routing. To exploit the vulnerability deliberately, the faulty router would have to be able to handle and pass on the received traffic for the incorrectly announced destination. Furthermore, if the incorrect routing were discovered, the signatures on the routing information would identify the faulty router as the source of the incorrect information. Finally, this design checks router advertisements against allowed address ranges certified by a trusted entity. A faulty router could announce nonexistent host or stub network routes, but only to addresses within its allowed ranges.

9.3. Autonomous System Border Routers

The autonomous system boundary routers can produce incorrect routing information in the external routes information they originate. There is no way to double check or corroborate this information, as there is with area border routers. No authority within an autonomous system exists to authorize the networks an autonomous system boundary router could announce, as is the case for the internal networks an internal router could announce. Consequently, the autonomous system boundary routers remain a unprotected vulnerability. With this in mind, special care should be taken to protect the autonomous system boundary routers with other means.

10. Security Considerations

This entire memo is about security considerations.

11. References

[1] Finn, Gregory G., "Reducing the Vulnerability of Dynamic Computer Networks," ISI Research Report ISI/RR-88-201, University of Southern California Information Sciences Institute, Marina del Rey, California, June 1988.

[2] Kumar,B and Crowcroft,J., "Integrating Security in Inter-Domain Routing Protocols", Computer Communications Review, Vol 23, No. 5, October 1993.

[3] Moy, J., "OSPF Version 2," RFC 1583, Proteon, Inc., March 1994.

[4] Perlman, R., "Network Layer Protocols with Byzantine Robustness", Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT, August 1988.

[5] Perlman, R., "Interconnections: Bridges and Routers", Addison-Wesley, Reading, Mass., 1992.

[6] Schneier, B., "Applied Cryptography: Protocols, Algorithms, and Source Code in C," John Wiley & Sons, Inc., New York, 1994.

[7] Steenstrup, M., "Inter-Domain Policy Routing Protocol Specification: Version 1", RFC 1479, BBN Systems and Technologies, July 1993.

[9] PKCS #1: RSA Encryption Standard, RSA Data Security, Inc., June 1991, Version 1.4.

[10] Eastlake D. & Kaufman C., "Domain Name System Security Extensions", RFC2065, January 1997.

[11] Moy J., "OSPF Version 2", Cascade Communications Corp, Work In Progress.

12. Authors' Addresses

Sandra Murphy murphy@tis.com
Madelyn Badger mrb@tis.com
Brian Wellington bwelling@tis.com

Trusted Information Systems
3060 Washington Road
Glenwood, MD 21738