[46] J. Steiner, *The Kerberos Network Authentication Service Overview*, **MIT Project Athena RFC, Draft 1**, April 1989.

[47] V. Voydock and S. Kent, *Security Mechanisms in High-level Network Protocols*, **ACM Computing Surveys**, June 1983.

[23] S. Kent, *Protocols and Techniques for Data Communication Networks*, **Englewood Cliffs, NJ:Prentice Hall, pp. 369-432**, 1980.

[24] M. Lepp and M. Steenstrup. *An Architecture for Inter-Domain Policy Routing* **BBN Report Number 7345**, July 1990, BBN Corporation, Cambridge, MA 02138.

[25] J. Linn, *Privacy Enhancement for Internet Electronic Mail. Part I: Message Encipherment and Authentication Procedures*, **RFC 1113, SRI Network Information Center**, January 1989.

[26] J. Linn, *Privacy Enhancement for Internet Electronic Mail. Part II: Algorithms, Modes and Identifiers*, **RFC 1115, SRI Network Information Center**, January 1989.

[27] J. Linn, *Practical Authentication for Distributed Computing*, **Proceedings of 1990 IEEE Symposium on Security and Privacy**, May 1990.

[28] K. Lougheed, Y. Rekhter, *Border Gateway Protocol*, **RFC 1105, SRI Network Information Center**, June 1989.

[29] J. McQuillan, *Adaptive Routing Algorithms for Distributed Computer Networks*, **BBN Report 2831**, May 1984.

[30] J. McQuillan, I. Richer, E. Rosen, *The New Routing Algorithm for the ARPANET*, **IEEE Transaction on Communications**, May 1980.

[31] J. Mogul, *Simple and Flexible Datagram Access Controls for Unix-based Gateways*, **Proceedings of Summer 1989 USENIX Technical Conference**, August 1987.

[32] J. Mogul, *Private Communication*, , June 1990.

[33] National Bureau of Standards, *Federal Information Processing Standards*, **National Bureau of Standards, Publication 46**, 1977.

[34] R. Needham and M. Schroeder, *Using encryption for authentication in large networks of computers*, **Communications of the ACM**, December 1978.

[35] D. Nessett, *Factors Affecting Distributed System Security*, **IEEE Transaction on Software Engineering, vol. SE-13**, 1987.

[36] International Standards Organization, *OSI Routeing Framework*, **ISO/TF 9575**, 1989.

[37] R. Perlman, *Network Layer Protocols with Byzantine Robustness*, **MIT LCS TR-429 (Ph.D. Dissertation)**, October 1988.

[38] J. Postel, *Internet Protocol*, **RFC 791, SRI Network Information Center**, September 1981.

[39] R. Rivest, A. Shamir and L. Adelman, *A Method for Obtaining Digital Signatures and Public Key Cryptosystems*, **Communications of the ACM**, February 1978.

[40] R. Rivest, *The MD4 Message Digest Algorithm*, **Proceedings of CRYPTO'90**, August 1990.

[41] E. Rosen, *Exterior Gateway Protocol (EGP)*, **RFC 827, SRI Network Information Center**, October 1982.

[42] J. Saltzer and M. Schroeder, *Protection of Information in Computer Systems*, **Proceedings of the IEEE**, September 1975.

[43] J. Saltzer, D. Reed, and D. Clark, *End-to-End Arguments in System Design*, **ACM Transactions on Computer Systems**, 1984, pp. 195-206.

[44] SDNS Protocol and Signaling Working Group, SP3 Subgroup, *SDNS Secure Data Network System Security Protocol 3 (SP3)*, **SDN.301 Revision 1.5 1989-05-15**

[45] M. Steenstrup et al, *Inter-Domain Policy Routing Protocol Specification and Usage: Version 1* **BBN Report Number 7346**, July 1990, BBN Corporation, Cambridge MA 02138.

References

[1] Advanced Micro Devices, *AMD MOS Microprocessors and Peripherals Data Book.*, **Advanced Micro Devices, Inc., Sunnyvale, CA.**, 1987

[2] ANSI, *Intermediate System to Intermediate System Inter-domain Routeing Information Exchange Protocol*, **Document Number X3S3.3/90-132**, June 1990.

[3] D. Balenson, *Private Communication*, , July 1990.

[4] W. Birnbaum, *SP3 Peer Identification*, **Proceedings of 1990 IEEE Symposium on Security and Privacy**, May 1990.

[5] *Blacker Front End Interface Control Document*, **DDN Protocol Handbook, Volume 1**, DDN Network Information Center, SRI International, 1985.

[6] H. Braun, *Models of Policy Routing*, **RFC 1104, SRI Network Information Center**, June 1989.

[7] L. Breslau and D. Estrin, *Design of Inter-Administrative Domain Routing Protocols*, **Proceedings of 1990 ACM Sigcomm Conference**, September 1990.

[8] CCITT, *The Directory Authentication Framework*, **CCITT Recommendation X.509**, 1988.

[9] D. Cheriton, *SIRPENT: A High-Performance Internetworking Approach*, **Proceedings of 1989 ACM SIGCOMM Symposium**, September 1989.

[10] D. Clark, *Policy Routing in Internet Protocols*, **RFC 1102, SRI Network Information Center**, May 1989.

[11] W. Diffie, *The First Ten Years of Public-Key Cryptography*, **Proceedings of the IEEE**, May 1988.

[12] D. Estrin and G. Tsudik, *Security Issues in Policy Routing*, **Proceedings of 1989 IEEE Symposium on Security and Privacy**, May 1989.

[13] D. Estrin, J. Mogul, G. Tsudik, *Visa Protocols for Controlling Inter-Organizational Datagram Flow*, **IEEE Journal on Selected Areas in Communications**, May 1989.

[14] D. Estrin, *Policy Requirements for Inter Administrative Domain Routing*, **RFC 1125, SRI Network Information Center**, December 1989.

[15] D. Estrin, G. Tsudik, *End-to-End Argument for Network-Layer Access Controls*, **University of Southern California Technical Report TR-90-13** , July 1990.

[16] European Computer Manufacturers Association, *Inter-Domain Intermediate Systems Routing*, **Technical Report ECMA/TC32-TG10/89/56**, May 1989.

[17] D. Feldmeier, *Improving Gateway Performance with a Routing-Table Cache*, **Proceedings of IEEE INFOCOM'88**, March 1988.

[18] M. Gasser, A. Goldstein, C. Kaufman, B. Lampson, *The Digital Distributed System Security Architecture*, **Proceedings of the 1989 National Computer Security Conference, Washington D.C.**, December 1989.

[19] S. Hares, D.Katz, *Administrative Domains and Routing Domains: A Model for Routing in the Internet*, **RFC 1136, SRI Network Information Center**, December 1989.

[20] R. Jain, *Characteristics of Destination Address Locality in Computer Networks: A comparison of Caching Schemes*, **Journal of Computer Networks and ISDN Systems**, May 1990.

[21] H. Kanakia, D. Cheriton, *The VMP Network Adapter Board NAB: High-Performance Network Communication for Multiprocessors*, **Proceedings of 1988 ACM SIGCOMM Symposium**, August 1988.

[22] P. Karger, *Authentication and Discretionary Control in Computer Networks*, **Computer Networks and ISDN Systems**, January 1986.

of the Open Routing Working Group, and anonymous reviewers for their useful comments on an earlier draft of this paper. We are especially grateful to Sharon Anderson who, tragically, is no longer with us. This research was supported by the NSF Presidential Young Investigator award and matching funds from ATT, GTE, and NCR.

behavior, integrating Policy Routing with end-to-end mechanisms, (e.g., *Visa*), and developing formal tools to verify the Policy Terms and Policy Routes computed by ADs.

Perhaps of most immediate importance is to study techniques for streamlining packet processing in routers, in general, and with respect to validation issues in particular. Some of the more interesting techniques may be implemented at the expense of security. For example:

- Check the signature *after* a packet is forwarded and then treat the *next* packet in the session/from that source according to the result of the previous check. This technique has the attractive characteristic that an individual AD can implement this short cut unilaterally and thereby select its own point on the security performance curve. Cheriton[9] calls this approach *optimistic authentication.*
- A similar idea at a lower level is to compute and append the signature at the *end* of each packet so that it can be done in parallel with other packet processing tasks. This technique is described by Cheriton and Kanakia for transport protocols[21].
- Delivery of signed multicast messages raises new issues, in particular, multi-destination policy route setup and accounting. [25, 26].
- A higher level issue is to integrate the support of connection and transaction oriented traffic. Schemes such as IDPR (and secure versions of IDPR in particular) require significant state to be set up and maintained in the routers when a usable PR does not already exist. Such overhead makes sense for connections but may be excessive for transactions. It is worth investigating a scheme that would allow the source to flag transaction traffic and have policy routers handle the traffic differently (more efficiently) if their respective ADs are so willing.
- This brings us to the more general issue of managing the state in routers. Management issues include route setup and tear-down, cache management, and state reestablishment in case of a failure or cache overflow. For example, while the protocols allow for state reestablishment without breaking the higher level connection by repeating route setup, there may be techniques to obtain recently lost state information from neighboring ADs. Such protocols are worth investigating to support route state management. Moreover, such a requirement is not unique to PR, it is shared by other schemes involving setup such as resource reservation protocols.

Consideration of techniques such as these highlights the need for tools to systematically identify, evaluate, and express security risks. Without such tools it is difficult to identify the security risks that are introduced when we attempt to improve performance.

In conclusion, this paper presented the design of preventative security mechanisms and an analysis of their performance implications. We conducted our analysis in the context of a specific PR protocol in order to provide a concrete basis for the evaluation of performance and implementation overhead.

8 Acknowledgments

We would like to thank David Clark for providing early access to his ideas on the RFC 1102 Policy Routing proposal. We also thank Steve Bellovin, John Linn, Lee Breslau, Kim Korner, members

- N hash function computations and N public key signature verifications for verifying the setup packet signature *en route* (one for each AD_i).
- N conventional decryption operations by each AD_i to decrypt K_{dsig}. However, this can be done after the setup packet is forwarded to the next hop and so does not contribute directly to the setup latency.

The above is the worst case scenario. Some AD_is may not care to authenticate PRs upon setup. Furthermore, if PR authentication and integrity requirements (or lack thereof) are expressed in AD_i's PTs, AD_{src} can avoid unnecessary signature computation and reduce the setup overhead.

6.4 Other Per Packet Processing Costs

Additional (other than encryption) processing costs are incurred mainly by the added logic in routers for processing of PR-based packets, in particular, table lookups. Similar experiments show that time spent on lookups is far overshadowed by the encryption costs[13].

7 Conclusions and Future Work

Transit control mechanisms are needed by interconnected ADs to retain their autonomy in setting and enforcing policy while still achieving desired connectivity. This problem of interconnecting and navigating across Administrative Domains is of inherent interest to the security community because the policies in question concern control of resource access and usage. Moreover, the security of the transit control mechanisms themselves must be considered if they are to be applicable in sensitive environments. On the other hand, the security mechanisms, as usual, take a toll in overall system complexity and performance. The purpose of this paper was to explore the design of transit control mechanisms for sensitive environments and to investigate the performance overhead of the proposed mechanisms. After evaluating the application of stub-network access control techniques, we described and evaluated a secure protocol design for transit ADs.

Policy Routing security should be approached from an integrated perspective. It exists in the context of end system and network access control and the division of labor deserves consideration. In this paper we proposed that policy routing should be able to prevent unauthorized use of network resources, and control over routing of packet data, across AD boundaries. Network access controls are responsible for finer grain control (e.g., on a host, as opposed to AD basis) and end-systems for protection of non-network resources. Our proposed mechanisms were designed to support inter-operability across ADs with heterogeneous policies to the extent that their combined policies allow. Moreover, these *preventative* PR mechanisms can inter-operate with *detection*-based PR mechanisms.[17]

There remains much work to be done in several areas: simulating and experimenting with implementations of Policy Routing protocols, speeding up available signature mechanisms, experimenting with the data integrity alternatives proposed here for more detailed understanding of their

[17]In order to accomplish this, PR headers must indicate the type of data integrity, replay prevention mechanisms must be employed for packets belonging to a stream, and each AD's Policy Terms must contain details as to data integrity checking and replay prevention requirements.

6.1 Per Packet Integrity Costs

Per packet signature costs are largely dependent upon the particular variation of data integrity checking. These costs are summarized in Figure 3. N refers to the PR length including AD_{src} and AD_{dst}, and m refers to the index agreed to in the Source Patterned scheme.

If we consider, for example, a processing speed of 8 Mbits/sec for MD4, each hash function computation for a 1K byte packet will amount to $1ms$. Assuming DES encryption rate of 1 Mbit/sec, the overhead for encrypting the resultant 128-bit value will be $0.13ms$. The per packet overhead will amount to $2.26ms$ for either Endpoint-AD or Round Robin data integrity checking. The same packet will incur the overhead of $N * 2.26mss$ for the Full Transit variant (where N is the PR length). These numbers should be considered relative to the usual packet processing, transmission, and propagation delays encountered in transiting a wide area network. Replay prevention can be

Variation	Cost in # of encryptions
Endpoint	2
Full Transit	N
Designated Transit	2
Source Patterned	N/m
Round Robin	2

Figure 3: Encryption Operations.

used independent of the data authentication method. The cost of replay prevention amounts to one additional PR header field (32-64 bits depending on the timestamp granularity) and several instructions for implementing the protocol described in Section 4.6.6.

6.2 Costs Due to Increased Packet Length

Increased packet length is incurred by the PR header carried in every data packet. It is anticipated that the length of this header will be on the order of 32 bytes. Previous measurements of *Visa* Protocols[13] show that this overhead ranges from 20% for small (e.g., 16 bytes of user data) packets to less than 4% for larger (e.g., 1K byte) packets; the length of a *Visa* header is roughly the same as that of a PR header. Roughly, the header overhead amounts to about 1 ms per packet per hop.

6.3 Setup Overhead

PR setup is accomplished by composing and sending a packet containing the entire PR as described in Section 5.1. The costs include:

- N conventional encryption operations by AD_{src} to encrypt K_{dsig} for all intervening AD_is, if data integrity is checked en route.
- A hash function computation over the entire PR setup packet followed by a single public key signature computation of the 128-bit hash function value.

5.3 Applicability to other PR Approaches

In the previous sections, we concentrated on increasing the security of the IDPR proposal. It is also of interest to examine the applicability of the methods discussed to BGP.

- Distribution of Policy Terms
  Flooding of Policy Terms are a feature unique to link state protocols such as IDPR. However, in BGP, paths and distance metrics are exchanged among neighboring ADs. In order to make these exchanges secure, it is sufficient to use a conventional cryptosystem as the number of message recipients is always limited to the number of directly adjacent ADs.

- Route Setup and Packet Forwarding
  Route setup is not a feature of the BGP architecture (as described in [28, 16, 2]. This means that Policy Routes must be authorized/authenticated at packet forwarding time. However, nothing precludes PR setup from being implemented on a network that uses these protocols, perhaps in conjunction with QOS routing. In that case, the discussion in Sections 4.5, 4.6 and 5 will be applicable.

- Data Integrity Checking
  Of the five variants discussed, only Endpoint-AD and Transit-AD data integrity checking apply to BGP architecture. Other variants require coordination among ADs beyond that provided in BGP as it is currently specified.

- Preventing Replay of Data Packets
  Border routers in BGP do not maintain state with respect to current traffic as does IDPR. Therefore, replay prevention, as described in Section 4.6.6 is not applicable because t_{lower} is not maintained. Nevertheless, it is still possible to use timestamps to detect some old data packets.

6 Assessment and Cost of Security

Our purpose in this section is to investigate bounds on achievable data rates with the security schemes described above. Previous work in the area of performance and cost evaluation of secure protocols [13] identifies four important overhead contributors (listed in the order of magnitude):

- Per Packet Signature
- Increased Packet Length
- Setup Overhead
- Other Additional Per Packet Processing

Experimental results show that that overhead due to signatures constitutes the majority of the total overhead. In the remainder of this section we analyze each of the above contributing factors in several variations of the general scheme.

5.2 Conventional Encryption Version

The conventional encryption version of of PR setup protocol is illustrated below. We assume out of band distribution of K_i^j-s.

- Upon issuing a PR, AD_{src} computes N PR signatures (one for each AD_i) using pairwise keys:

$$SIG_i = E(F_{hash}(PR_{setup}))^{K_i^{src}} \tag{4}$$

- As before, AD_{src} generates a new key, K_{dsig}, which needs to be encrypted separately for each AD_i. The resulting PR setup packet is depicted in Figure 2b.
- Next, the PR setup packet is sent along the route. Each participating AD_i obtains K_{dsig} by decrypting $E(K_{dsig})^{K_i^{src}}$, recomputes and verifies SIG_i and validates the timestamp. Then, it authorizes the PR by checking it against local policy constraints.
- The remaining steps of the protocol are similar to the public key variant.

PR	
Timestamp	
SIG_{src}	
	$E(K_{dsig})^{EK_1}$
	$E(K_{dsig})^{EK_2}$
	.
	.
	.
	$E(K_{dsig})^{EK_N}$

(a) Public Key Version

PR	
Timestamp	
SIG_1	$E(K_{dsig})^{K_1^{src}}$
SIG_2	$E(K_{dsig})^{K_2^{src}}$
.	.
.	.
.	.
SIG_N	$E(K_{dsig})^{K_N^{src}}$

(b) Conventional Encryption Version

Figure 2: PR setup packets.

where $VALID_i$ is the information that AD_i can use to validate a PR (e.g., a list of applicable Policy Terms), and CU_i reflects the conditions of usage, such as UCI-s and/or a Charge Codes. Typically, a PR will contain no secret information, thus, it can travel in the clear. Note that a PR carries no information regarding individual host pairs. This is because all PRs are initially validated on a $[AD_{src}, AD_{dst}]$ basis. If a transit AD's policy terms are host-specific, that AD verifies host addresses at the time of packet forwarding.

5.1 Public Key Version

In this section we present a version of the secure PR setup protocol based on public key encryption. (For a more complete description, the reader is referred to [45]).

- The setup packet is protected by a single signature:

$$SIG_{src} = E(F_{hash}(PR_{setup}))^{DK_{src}} \tag{2}$$

  where DK_{src} is the private (signature) key of AD_{src}. Given a strong one-way hash function (such as MD4) and a strong encryption function (such as RSA), this signature is sufficient to maintain the integrity of the PR setup packet. Timeliness can be maintained by timestamping the setup packet.

- If data packet integrity is desired, the issuing AD_{src} needs to generate a new data signature key, K_{dsig}, for use in whatever per-packet data integrity variation is used. K_{dsig} must be communicated in secret to each AD_i. This requires that AD_{src} encrypt K_{dsig} N times, i.e., compute $E(K_{dsig})^{EK_i}$ for all AD_i. The resulting PR setup packet is depicted in Figure 2a.

- When the PR setup packet propagates along the route each AD_i obtains K_{dsig} by computing $E(K_{dsig})^{DK_i}$, recomputes and verifies SIG_{src}, and validates the timestamp (possibly, by comparing it to the timestamp of the previous setup packet, which it may choose to keep). It then proceeds to authorize the PR.

  At this point, each AD_i is assured that: (i) the PR is valid, i.e., does not violate local policy, (ii) the PR is authentic, i.e., issued by a recognized entity; and, (iii) the PR is *fresh*, i.e., issued recently. In other words, each AD_i is protected against attacks of **Type 1** and **Type 2**.

- In order to make use of an existing PR, the source must be able to supply information necessary to associate each data packet with a specific PR. This information is placed in the *PR header* mentioned in Section 4.6. If data packet authentication and integrity is desired, the source must be able to compute packet signatures with K_{dsig} in order to defend against **Type 3** attacks.

  When the source AD is ready to forward a data packet, P, it computes a packet signature:

$$DSIG = E(F_{hash}[PR_{header}||Packet])^{K}_{dsig} \tag{3}$$

  The details of this process depend on the variation of data integrity checking used (as described in Section 4.6).

control of transit traffic, we present a method for preventing replay.

There are two basic approaches for countering replay attacks: i) nonce identifiers, and ii) timestamps [16] The main disadvantage of using nonces is the difficulty in their verification. In particular, each relevant entity (each PG, in our case) needs to keep a complete history of past nonces which makes the verification inefficient. Timestamps are much better suited for this application. First, clocks need not be continuously synchronized between the source and the transit PGs. This is because a PR setup packet is timestamped; its timestamp can be used as a *lower-bound* for subsequent data packets in all intervening PGs. Furthermore, if intervening PGs maintain a more *current* lower-bound timestamp (t_{lower}), opportunities for replay can be reduced further.

Consider the following protocol:

1. When a PR is issued, PG_{src} timestamps the PR setup packet, and distributes the timestamp, t_{setup}, in a secure fashion to all intervening PGs in transit ADs. All transit PGs initialize their t_{lower} values for this PR to t_{setup}.
2. When a data packet is sent, the originating (first-hop) PG, timestamps its PR header. (Let t_{data} denote this value).
3. When this data packet reaches a transit PG, its PR header is examined and the t_{data} is compared to t_{lower}. Three outcomes are possible:
   (a) $t_{data} < t_{lower}$. The difference between the two values is examined. If it is small, i.e., less than some (locally defined) threshold, $delta_t$, the packet can be forwarded. Otherwise, the packet is discarded.
   (b) $t_{data} > t_{lower}$. In that case, the packet is forwarded and t_{lower} is set to t_{data}. Of course, a PG may get suspicious if the difference is too large.
   (c) $t_{data} = t_{lower}$. This can occur when two successive data packets belonging to the same PR stream carry the same timestamp. To distinguish between such packets, it would be necessary to keep additional information, e.g., a packet signature, for the last data packet processed. However, it is desirable for the clock rate to be at least as fast as the maximum packet rate. This would preclude duplicate timestamps on data packets.

This protocol prevents most, but not all, replay attacks. In order to prevent all replay attacks, $delta_t$ values must be set to zero in all transit PGs, which would essentially disallow any out-of-order data packets. This is a choice that will not be practical for environments where out-of-order packets are a frequent occurrence.

5 Protocol Description

We can now combine the PR setup and packet forwarding mechanisms into a single protocol. For the purpose of the following protocol description, a PR is composed of:

$$[AD_1, VALID_1], [AD_2, VALID_2], ..., [AD_N, VALID_N] \tag{1}$$

[16] It can be argued that timestamps of sufficient width and granularity are nonces as well. However, a *true* nonce is a randomly chosen number that is hard to predict.

of the PR's bandwidth can be abused. Moreover, the synchronization inherent to this protocol implies that care must be taken to recover from lost and out-of-order packets.

Alternatively, instead of the source choosing m (as in Patterned variation above), transit AD_is can choose their own m_is and may elect not to disclose them. Or, a transit AD_i could choose to authenticate each packet with probability p_i. In this scheme all data packets are signed at the source, but only $1/m$-th (or $p\%$) of the packets are checked per AD hop. This method has the advantage of being flexible and robust in that synchronization is not required.

4.6.5 Round Robin

This scheme achieves constant per packet overhead by using *round-robin* data authentication. Transit ADs take turn authenticating packets. In general, packet number K is authenticated by a PG in $AD_{[K mod M]}$ where M is the number of ADs in the PR. All data packets are signed at the source but only one check is done en route to the destination. Destination checking can be added for extra assurance at the cost of a single additional decryption by the destination. Moreover, unlike Source Patterned variant, lost and out-of-order packets can be accommodated easily. On the other hand, AD independence must be sacrificed due to the coordination required to set up the round-robin arrangement. While this approach benefits from fair sharing of encryption costs among transit ADs, it is only worth considering in cases when the number of transit ADs is large, i.e., the PR is long.

4.6.6 Preventing Replay of Data Packets

The final type of attack considered is the replay of data packets:

Type 4. *An intruder can replay previously recorded data packets which can lead to unjustified charging and/or denial of service.*

There are other, more serious threats posed by malicious replay. However, we are concerned primarily with protecting network-layer resources; other replay attacks are assumed to be handled by the end-points. Also, we only need to protect against replayed packets within the life-span of the associated PR. After a PR expires or is closed all packets carrying the expired PR identifier will not be processed.

Two sources of replay deserve equal consideration. The first is accidental replay due to a misbehaving machine stuttering and generating replayed packets. The second is malicious replay due to an intruder intentionally replaying prerecorded packets in order to deny resources (or inflate costs) to the rightful owner. Neither kind of replay can be handled on a purely end-to-end basis because by the time a duplicate packet is discovered, the resources are consumed and associated charges are incurred, e.g., the bill reflects the replayed packet and the rightful user of the afflicted charge code can no longer obtain service due to an overdrawn account.

In some circumstances, the post facto approach of replay detection and cost recovery may be adequate. This includes auditing packet counts, setting a limit on the number of packets that can use a PR and other ad hoc methods. However, in sensitive environments, more aggressive prevention is required, albeit at some cost. Since our goal is to analyze the implications of secure

4.6.1 Full Transit

In network environments where data integrity and security concerns outweigh the overhead of extra processing, the data portion of every packet is subject to forgery and must be checked (for authenticity and integrity) at each hop on its way to the destination. Every data packet is signed at the source and checked at each AD hop en route. The protocol for this class of environment has the highest overhead, commensurate with security requirements. The per-packet processing in this scheme is similar to transit Visa.[15].

4.6.2 Endpoint

If authenticating data in each packet at every hop is prohibitively expensive, end-to-end data integrity similar to that in endpoint *Visa* protocols may be appropriate. Every data packet is signed at the source but is checked only at the destination. This approach has limitations, most notably the fact that an intruder located at some point along the route can modify data in each packet and the forgery will not be detected until the packet reaches the destination AD. This can result in unauthorized use of transit resources and inappropriate billing of the source. On the other hand, this approach benefits from lower per packet latency which is independent of the PR's length. This approach provides preventative control for stub ADs, but only detection-based control for transit ADs.

4.6.3 Designated Transit

If Endpoint exposes transit resources to excessive misuse, yet Full Transit is too expensive, the source AD can designate at PR setup time a specific transit AD to perform data integrity checks. Every data packet is still signed at the source, but only one transit and the destination check the signature. The positioning of the designated AD in the PR is important: having it too close to AD_{src} is almost equivalent to no checking at all, whereas having it too close to AD_{dst} is equivalent to the Endpoint variant. The designated AD can be reassigned from time to time in order to reduce the chance of its exploitation by an intruder.

4.6.4 Patterned

Instead of each transit AD having to authenticate each packet, it may suffice to authenticate every m-th packet. In the simplest version of this patterned authentication scheme, AD_{src} would choose m at random from a locally defined range of values and then specify m during route setup. In this scheme only $1/m$ data packets are signed at the source and the same $1/m$ packets are checked.

Transit ADs would either accept or reject the proposed m. If all ADs accept the proposed value for m, then every AD will check data integrity of every m-th packet. If any AD does not accept m (if it is considered too large or too small) then the source and all other ADs must choose a different m. In return for reduced overhead, if the value for m is discovered by an intruder then *(m-1)/m*

[15] However, the Full Transit approach avoids the per-session setup dialog associated with Visa.

internetwork). This makes conventional encryption a viable choice since a PR would only have to be signed N times.

The above discussion is a standard public key *verses* conventional encryption debate. Both methods have beneficial as well as burdensome features. In Section 5.1 we demonstrate a PR setup protocol based upon public key encryption. The conventional encryption variant is outlined in Section 5.2.

4.6 Packet Forwarding

After a PR has been set up, subsequent data packets can take advantage of the PR state in intermediate ADs. First, instead of a full PR, each data packet carries only an abbreviated version referred to as the *PR header*. Second, the state in all intervening PGs allows them to bypass expensive authorization checks on a per packet basis. A PR header only needs to contain the information necessary to identify the appropriate state in intervening PGs. Its exact contents are described in the next section.

Assuming appropriate security measures to prevent PR setup threats above, there remains the possibility of malicious attack at packet forwarding time:

Type 3. *An intruder located at some point along a PR can copy a valid PR header from a legitimate packet, attach its own data and send it along a PR, thus, obtaining service fraudulently.*

This attack can be remedied if each data packet (or a function thereof) is signed by the source. Depending on the type of encryption used, the overhead incurred by signing each data packet (or even a hash function thereof) can be prohibitively high. Per-packet overhead includes the signature computation at the source and its subsequent verification at each PG hop en route to the destination.

If only sender authenticity and data integrity is desired, then the tradeoff between conventional and public key signature of the hash function is dependent upon their relative speeds. In order to minimize per packet processing overhead, we will assume the use of conventional encryption for data packet signature computation. In this case, a secret key K_{dsig} is distributed during setup for use in data integrity verification. This key is shared by all ADs on the route. As a result, a simple group channel is established. However, it is possible for any AD along the route to masquerade as the source AD for the duration of that PR. Moreover, the secret key must be distributed without disclosure; requiring the public key scheme to encrypt the key multiple times, separately with the public key of each AD in the PR.

Even more so than in the case of PR setup, latency is a critical concern with respect to forwarding of data packets. For this reason, many ADs are likely to forego per-packet signature and verification of most traffic. We now discuss a spectrum of possibilities whereby ADs can trade the level of protection for the amount of overhead incurred. The section is concluded with a discussion of data packet replay in 4.6.6.

For the remainder of this paper we consider environments in which only integrity and authenticity of routing data is required and where public key encryption and the certifying authority mechanisms described in [27] are used for this particular function.

Although the distribution of PTs raises a number of interesting issues, the underlying concept of broadcasting signed messages is not unique to Policy Routing[14]. Other aspects of secure Policy Routing, such as PR setup and packet forwarding, require more careful protocol design because of the associated costs and threats.

4.5 Route Setup

The route setup phase requires that each intervening AD have means to forward subsequent data packets along a specific Policy Route. As described in Section 3.2.2, each AD along the route must be supplied with the next hop AD at PR setup time. The purpose of PR setup is to establish state in all intervening PGs so that Policy Routing decisions can be made in advance of the actual communication, and, subsequent data packets can carry a minimum of PR-related information, thus, reducing the overhead and latency. In addition, in a secure PR scheme, each AD must be able to authenticate and authorize a PR. Authentication means verifying that a PR was issued by a recognized entity, and was not tampered with. Authorization entails making sure that a PR conforms to local policy (the latter is beyond the scope of this paper).

More specifically, secure PR setup needs to address two types of threats:

Type 1. *Creating, or tampering with existing, PRs by an intruder.*
In order to defend against this threat, each PR must be traceable to the issuing AD. In other words, it must be signed with an unforgeable signature. For all intervening ADs, this would provide for non-repudiation of issuance and sender authenticity.

Type 2. *Replay of previously issued PR setup packets.*
This can be prevented if we include a timestamp within each PR setup. The signature of a PR setup packet becomes dependent on this timestamp, and, makes replay detection possible within the granularity of the timestamp and clock synchronizations. As the number of setups is relatively small (in relation to the number of data packets), relatively coarse timestamp granularity (e.g., $1ms$) should be adequate and is preferable to the management required to keep track of unique sequence numbers (nonces [34]).

As with PT update distribution, we are concerned with data integrity and authenticity of setup packets. However, unlike PT updates, PRs are set up frequently and increased latency is experienced directly by end users. Thus, we are far more concerned with the per-signature overhead for setup than we are for PT distribution. Consequently we will investigate the use of both conventional and public key encryption signature mechanisms.

As discussed earlier, conventional encryption, implies a significant key management burden, since an AD has to share a secret key with every other AD that it ever communicates with. Moreover, it entails computing a PR signature for every AD involved, whereas a single PR signature verifiable by all intervening ADs is sufficient in public key encryption. On the other hand, a typical PR traverses a relatively small number of ADs (N is much less than the diameter of the

[14]With the exception that routing updates are infrequent, thus, favoring the use of public key encryption.

Both public key and conventional encryption methods require key management. We will not discuss such issues in this paper because they are not unique to the problem of transit traffic control and have been discussed extensively elsewhere [25, 26]. Suffice to say, that for conventional encryption, key distribution centers (KDCs) are needed in each AD. Moreover, inter-AD key exchange must be supported. For public key encryption, a certification authority model such as that described in [18, 25] is appropriate.

4.4 Distribution of Policy Terms

In order to provide for secure distribution of Policy Term updates, each AD must be able to sign its own, and authenticate incoming PTs. Because of the Link State nature of the protocol, PT updates must be flooded to ADs throughout the internetwork so that all participant ADs can use them in their PR computation. Before using a new PT, each AD needs to verify the authenticity and integrity of its contents. This is difficult to achieve in a conventional encryption environment, as the number of potential recipients of a PT update can be quite large.[13] In general, conventional encryption is not well-suited for an environment such as Link State routing where routing updates are broadcasted to a large number of recipients.

The alternative is to use public key encryption for the distribution of PTs. At first glance, this might appear problematic because current public key technology is still inferior in terms of performance. However, if we are concerned only with the integrity and authenticity of routing information then the signature mechanism described in Section 4.3 above can be used with little performance impact. Moreover, one of the central assumptions in the IDPR proposal is that policies change relatively slowly. Any added processing time associated with public key encryption is counter-balanced by the ubiquity and efficiency of being able to generate a single unforgeable packet signature which can be authenticated by any recipient. An example of a Link State routing protocol which uses public key encryption for routing update distribution is presented in [37].

Routing information distribution is one area of policy routing where confidentiality measures may be considered useful. However, it implies that the entire routing update must be encrypted **separately** for each anticipated destination AD. In general, this is impractical regardless of the type of encryption used. Since a link state update is flooded throughout the internetwork (in this case, to all ADs), the number of potential destinations can be quite large. In order to achieve confidentiality in this environment, the source of a link state update needs to encrypt the update N times (where N is the total number of ADs). Furthermore, the traffic due to update propagation will increase N-fold.

BGP has a similar requirement for authenticity and integrity (and, possibly, confidentiality) of routing information. However, because it is a distance vector protocol, routing updates are distributed to a much smaller number of destinations, i.e., only the direct neighbors of the originating router. At the same time, each message is considerably larger (equivalent of a complete routing table) than in the link state algorithm. Conventional encryption may be more appropriate in this case since the number of keys needed per policy router will be small, hence, easy to manage. Moreover, if confidentiality is desired, conventional encryption will save considerably on processing time.

[13] Sharing a single common key amongst all nodes affords little protection, while distributing pairwise keys to each AD-pair is impractical.

- N denotes the length of a PR, i.e., the number of ADs in a PR.
- AD_i $(0 < i < N)$ denotes the i-th AD in a PR, $AD_1 = AD_{src}$ and $AD_N = AD_{dst}$.
- K_i^j denotes a secret key shared by AD_i and AD_j.
- K_{dsig} denotes a secret key used for computing data signatures.
- EK_i, DK_i denote public (encryption) and private (decryption) keys of AD_i.
- $E(Data)^K$ and $D(Data)^K$ denote encryption and decryption, respectively, of $Data$ with key K.
- $F_{hash}(Data)$ is a one-way hash function of $Data$, e.g., MD4 [40].

4.3 Message Integrity and Authenticity Mechanisms

In the next three sections we consider separately security issues pertaining to different phases of Policy Routing: distribution of PTs, PR setup and packet forwarding. In each of these phases we are concerned primarily with data integrity and source authenticity. Confidentiality of user data is left to end-to-end mechanisms [47]. Confidentiality of routing control information is a less general requirement than integrity and authenticity, and is discussed briefly. Traffic analysis is not addressed here (for further discussion see [15]). Before proceeding, we discuss the use of signed one-way hash functions to achieve efficient integrity and authenticity measures. This approach is described in [8].

For each message we compute $F_{hash}(Data)$, where $Data$ includes the invariant portions of the packet header (i.e., those fields that do not change en route between source and destination) and the packet data. The resulting value is then signed. If a public key scheme is used then the value is signed in the private key of the originator. The resulting value is referred to as the packet signature. Anyone needing to verify the message integrity and sender authenticity computes $F_{hash}(Data)$, decrypts the packet signature with the public key of the sender, and compares the two values. If a conventional encryption scheme is used, $F_{hash}(Data)$ is encrypted and decrypted with the secret key to produce and verify the signature, respectively.

The scheme is efficient because only $F_{hash}(Data)$ needs to be encrypted, and computing the one-way hash function is faster than encrypting. As a result, the difference in processing overhead between public key and conventional encryption schemes is reduced.

Throughout the reminder of the paper we will refer to Rivest's Message Digest algorithm [40] as an example one-way hash function. In addition to being the fastest method available currently, it is also being used as a basic data integrity mechanism in several other contexts, most notably, Privacy-Enhanced Electronic Mail [25, 26]. Processing speeds for MD4 implementations range between 15 Mbits/sec on a DECStation 5000 and 6.4 Mbits/sec on a SUN SparcStation-1 [32]. For a 1 Kbyte packet, this translates into an overhead between 540 and 1,280 μs per packet. Unfortunately, existing (software) RSA implementations requires 2-3 seconds to sign a 128-bit MD4 result value.[12]

[12] It is possible to speed-up the public key encryption somewhat. One such method is specified in CCITT X.509 document [8]. It uses the RSA [39] encryption method in a manner such that message decryption (verification) is significantly faster than encryption. This is achieved by selecting a large (e.g., 512-bit) decryption exponent and a relatively small (e.g., 16-bit) encryption exponent.

4.1 Threats

The two basic threats to the security of PR protocols are falsification of routing information and falsification of data packets.

1. An intruder may distribute false routing information in order to (i) disrupt communication, e.g., create routing loops, or (ii) eavesdrop on communication, e.g., re-route traffic to a specific location. In BGP this can take the form of distributing falsified or prerecorded routing updates. In IDPR, falsified Policy Terms can be similarly distributed causing invalid PRs to be computed.

2. If routing protocols protect themselves by prescribing an authentication mechanism for validating routing information, an intruder can turn to falsifying control and/or data packets. This kind of attack can lead to unauthorized resource usage, unauthorized communication, and inappropriate accrual of charges. Three sub-threats are identified:

   (a) Falsification of control information
   In addition to the usual network layer information (source and destination addresses, data size, etc.), control information includes the route setup and packet forwarding parameters. In IDPR, an intruder can steal a PR header created by an authorized AD and substitute an invalid Charge Code. Similarly, in the extension of BGP proposed as an international standard, distribution list information can be modified.

   (b) Falsification of data
   In both protocols, the data portion of a packet can be modified or replaced by an intruder.

   (c) Replay
   Previously-recorded legitimate packets can be replayed by an intruder. Two sources of replay are of equal concern: accidental replay due to the *stuttering* of a misbehaving machine, and malicious replay due to a misbehaving person attempting a denial of service attack.

In the remainder of this paper we describe and analyze mechanisms to resist these threats. Hereafter, we use the IDPR proposal as the foundation for our protocol design. It provides the finest-grained control through the use of AD-level source routes. Moreover, this greater control provides a wider range of possible attacks. Hence, security issues that arise in this proposal form a superset of those in the BGP approach. Our approach is not inconsistent with the original proposal in IDPR. The *prevention*-based (as opposed to detection-based) measures proposed here can be included as optional features within the protocol.

In the remainder of this section, we detail the steps needed in a secure PR protocol and discuss issues dealing with data integrity checking and replay prevention.

4.2 Terminology

The following terminology is used throughout the remainder of the paper:

- $||$ is the concatenation operator, e.g., $X||Y$.

A path is established with the first packet carrying the full PR, i.e., the complete sequence of ADs in the route and applicable PT identifiers. PGs along the way make sure that the PR agrees with the local PTs (through use of templates, for example). The result is cached so that a specified **PR handle** can be used in the future to refer to the cached entry. Successive packets carry a PR handle, not a full PR. Many transport level sessions, and even host-pairs, may share a single PR if the policies enabling it are not end-system-specific; this reduces the average latency and router state overhead associated with inter-domain communication. PGs use these handles in the packets to check for cache entries. PGs also may relate return flow packets with forward flow. Given information about the next AD for a particular packet, each PG selects the next PG based on the information exchanged in a traditional up-down protocol.

3.3 Discussion

Policy routing allows ADs to interconnect to the global internet while still protecting network resources from general, unconstrained use. (We described earlier why such a function can not be left to end-systems.) However, policy routing mechanisms do not preclude the need for network access controls in the border routers of ADs that wish to control access to individual end-systems. A more extensive discussion of the interplay between end-system controls, network access controls, and policy routing can be found in [15].

One essential difference between Visa and the policy routing approaches is the per session setup overhead. Transit Visa requires that a dialog transpire between the source and each transit-Visa networks' ACS, and that corresponding keys be distributed. Consequently, the initial setup delay grows in proportion to the number of transit networks. For short transactions such overhead is not acceptable. A PR-based approach such as that in BGP or IDPR avoids this setup dialog through background distribution of policy term information that is used in route computation. The work that the Policy Routing protocols do to distribute policy terms and compute authorized routes must be done at the time of the session setup in Visa. In particular, with Visa protocol this translates into a *reject* packet, ACS dialog, and visa-key grant message for each AD in the path. Moreover, this assumes that the source attempts communication over a path for which it has authorization. If there is a conflict with even one transit AD's policy, the process must begin again. Policy Routing incorporates policy into the route computation process in advance of the actual communication, thereby avoiding this problem.

On the other hand, the PR schemes, as described thus far, rely on *post-facto* detection of abuse, and, are in that sense less secure than network access control schemes, such as Visa protocols. The remainder of this paper addresses the integration of preventative mechanisms into policy routing to achieve secure control of transit traffic.

4 Security Issues in Transit Control

In this section we identify potential security threats faced by the two PR schemes described and detail the steps needed in a secure protocol.

1. Most policies can be classified and expressed in a standard notation.
2. Policies and inter-AD connectivity change relatively slowly.
3. End-point specific policies should be supported.

Two primary concepts in this proposal are *Policy Routes (PRs)* and *Policy Terms (PTs)*. A PR is a series of ADs. It is, essentially, an AD-level source route. In other words, there may be multiple physical realizations of a PR given multiple physical connections between ADs and multiple intra-AD routes. The actual selection of a particular physical path is done at packet forwarding time by each intervening AD, rather than by the source AD at route computation or route selection time. This *lazy evaluation* provides for a more adaptive protocol and unrestricted AD interconnection. Policy Terms (PTs) are the units of routing information exchanged by communicating ADs. Each PT represents a distinct policy of the AD that expressed it. The information distributed in a PT can be represented as:[10]

$$[(H_{end}, AD_{end}, AD_{ent}), (H_{end}, AD_{end}, AD_{exit}), UCI, Conditions]$$

The purpose of such a PT is to specify that packets from or to some end-point host, H_{end} (or a group of end-systems), in an end-point AD, AD_{end}, are allowed to enter the AD in question via some directly connected AD, AD_{ent}, and exit through another directly connected AD, AD_{exit}, on its way to or from a host, H_{end} (or a group of end-systems), in some end-point AD, AD_{end}. User Class Identifier (UCI) allows policies to distinguish between various user classes, e.g., Government, Research, Commercial, Contract. Conditions represent quality of service, billing, and other variables, and can reflect the agreements between neighboring ADs. Examples of the Policy Terms can be found in [14, 10].

Policies are expressed by source, destination, and transit ADs. The source AD may select all transit ADs while transit and destination ADs control which source and destination ADs can communicate via which directly connected ADs. ADs run link state routing algorithms to compute their respective tables of PRs. There may be multiple PRs listed for the same destination AD, each with a different set of conditions associated with its use (e.g., QoS, time-of-day, or UCI).

Note that ADs (with the exception of the source AD) do not exert control over the entire Policy Route. Referring back to our travel analogy, it is difficult to enforce policies that are based on information that is not verifiable at the point of reference. For example, it is difficult to enforce a policy that dictates non-admittance to anyone who has *ever* passed through country X, since it is very much dependent on X stamping passports reliably. In the environment of interconnected ADs, a transit AD can verify the previous and next hops because of its direct connections and the feasibility of employing pairwise authentication with the relatively small number of neighbors. Verifying other transit components of the PR is difficult, if not impossible.

Each AD has one or more Route Server (RS), an entity that collects Policy Routing information from other ADs, distributes local policy information to other ADs and computes, as well as issues, PRs to local end-systems. Actual policy enforcement is done at a Policy Gateway (PG) which, in addition to the usual task of forwarding packets, handles validation and verification of the PRs attached to the incoming packets.[11]

[10] This is a simplified version of the actual PT format used in the protocol. However, the differences are not relevant to this discussion.

[11] Policy Gateways (PGs) correspond to border routers in BGP. They perform level three routing functions.

ture. Neighboring nodes exchange reachability information for a specific destination in the form of distance metrics corresponding to each next hop. Nodes do not exchange information about subsequent hops to the destination. BGP augments this traditional approach by distributing full AD-level paths. In other words, for each destination advertised, nodes specify the AD-level path to that destination. As a result, BGP provides less information hiding in return for the ability to detect routing loops quickly. By using the full AD path to detect loops BGP avoids imposing topological restrictions on AD interconnection (such as those imposed by EGP). In addition, AD path information can be used as a policy criterion for route selection.

BGP allows for limited policy-based route selection. Each AD's BGP router can select its next hop based on the information provided in the full AD path, in addition to the distance metric. For example, AD_A can reject all routes through AD_B. On the other hand, each AD must apply the same route selection decision to all packet sources, including itself. For example, AD_A can not reject all routes through AD_B for itself without affecting its neighbors, and vice versa. Similarly, an AD can not apply one policy to one neighbor and a second policy to another neighbor. Since BGP *was not* intended to implement policies that discriminate between traffic end-points with arbitrary granularity, the approach achieves its goals [6].

Each BGP router can be configured according to its AD's local policy. Even though local policy is not distributed among ADs, it is represented in a universal *policy language*. A policy in this language is an expression:

[Network-list,AD-path]=preference

The semantics of a policy are as follows: if a routing update for a network in the *Network-list* is received via the *AD-path* and its *preference* metric is better than that of a path currently in use, then, this update must be redistributed to all ADs.

The proposed international standard augments the BGP protocol by including (among other features) distribution lists along with route information [2]. The list may be inclusive or exclusive and is propagated along with next hop and full-AD path information. Each border router along a path may further restrict a distribution list before advertising a route, i.e., ADs may be deleted from the inclusive list or added to exclusive list, but according to the protocol no router can relax or ignore the list.[8]

3.2.2 IDPR Policy Routing Proposal

An alternative architecture for policy routing has been developed to support a wider range of policies; with mechanisms that represent a more radical departure from existing routing techniques.[9] The Inter-Domain Policy Routing (IDPR) proposal allows stub and transit ADs to express and exchange packet routing and forwarding policies. The most distinguishing feature of this approach is the use of AD-level source routing. It uses a *Link State* algorithm [30] to compute source Policy Routes (PRs) at the granularity of ADs. Each AD expresses its policies in a *standard* form, called Policy Terms (PTs), and distributes them to other ADs. Each AD designates special Route Servers (RSs) to collect PTs and compute policy routes for constituent users. The basic assumptions of this model are:

[8] The proposed standard includes several other extensions which are not directly relevant to our discussion.

[9] This approach was first described by D. Clark in [10].

routing approaches in the next section we return to the problem of **secure** control of transit internetwork traffic.

3.2 Policy Routing

As described earlier, the central goal of transit traffic control is to allow ADs to independently express and enforce policies regarding transit traffic. Our discussion in the previous section demonstrates that transit control is intimately related to Inter-Administrative Domain Routing. We refer to inter-domain routing that incorporates policy constraints as policy routing (PR). Inter-domain routing constitutes the highest level of the OSI routing hierarchy as defined in [36].

In this section we summarize two approaches to policy routing. The first is the Border Gateway Protocol (BGP)[28] which is intended to support a limited notion of policy. The second is the Inter-Domain Policy Routing (IDPR) proposal designed to support more general policies[24, 45]. (For further comparison of inter-AD routing architectures see [7].)

Policy routing operates at the network layer. In both example architectures, only border routers and associated inter-domain route servers are directly affected by the presence of the inter-domain routing protocols. End-systems and interior routers can continue employing whatever internetworking protocols desired within their particular ADs. Border routers operate on behalf of the end-systems. For this reason, the term *source* hereafter refers to the border router in the AD of the source end-system.

In the remainder of this section we provide brief descriptions of the two inter-domain routing protocols as background to Section 4.

3.2.1 Border Gateway Protocol

BGP is a recently proposed addition to the Internet Protocol family[28]. It was designed to be a successor to the Exterior Gateway Protocol (EGP)[41] and a variant has been submitted as an international standard[2]. Its foremost goal is to provide efficient and robust Inter-AD routing with rapid convergence and loop detection for arbitrary internetwork topologies[7]. In addition, it provides policy-based distribution of routing information. It is aimed mainly at transit ADs and can inter-operate with other routing protocols.

BGP is designed under the following assumptions:

1. Policies can be expressed using information about the full AD path that packets will travel to a destination.
2. Transit policies apply uniformly to all end-points.

BGP uses hop-by-hop routing and a distance vector algorithm for the next hop selection[29]. One common benefit of traditional distance vector algorithms is the ability to hide network struc-

[7] BGP and EGP use the term Autonomous System and Routing Domain. We use the term Administrative Domain. They are not completely equivalent but for the sake of this discussion can be interchanged. See [28, 14] for further discussion.

host proceeds to use these keys to generate stamps which it places on each outbound datagram. Thus, each datagram carries two stamps: one that allows it to exit the local AD, and, the other, that allows it to enter the destination AD. Border routers of end-point ADs proceed to verify the validity of their respective visas and pass datagrams until a visa expires or is otherwise revoked. If the communication is two-way, visa-keys are issued to the destination host as well, and the authorization procedure is symmetric. For further details of the Visa protocol see [13].

The process of establishing authorization in Visa-controlled, transit ADs is essentially the same as for stub ADs in the Visa protocol. The major difference is that the source host must obtain a transit visa for each transit AD that requires one, in addition to obtaining a pair of exit/entry visas from AD_{src} and AD_{dst}. In the worst case, each transit AD's ACS will conduct an authorization and authentication procedure before issuing a transit visa, and, each packet will have to carry a separate visa for each intervening AD. Of course, transit ADs may choose to issue visas automatically, or not require any visas at all where transit traffic is concerned. Furthermore, ADs could program their ACSs to obtain and issue transit visa-keys in advance of the actual communication. This would reduce the setup delay at connection establishment time. On the other hand, such mechanisms increase the problems associated with visas expiring before, or while, they are in use.[5].

The use of Visa for transit control may be appropriate if: (i) transit policies are as diverse as stub network policies, and (ii) policies change frequently. The former limits the practicality of expressing policies in a simple universal syntax. The second assumption also makes it impractical to distribute policies in the way that we distribute connectivity information, for the fear of using stale route information or incurring excessive overhead due to frequent information updates. These assumptions result in several interesting features of transit Visa. First, organizational policies are embodied in ACSs and are not propagated outside; hence, a wide range of policy statements can be accommodated. Moreover, very little coordination among ADs (beyond that in Visa protocols for stub networks) is required to implement this protocol.

Although the extension of the Visa concept to transit control is rather straightforward, the approach does not scale well to an internetwork where many ADs, both stub and transit, want to control traffic flows. For example, visa acquisition and route setup must be repeated (or adapted) each time an involved visa-router goes down. Moreover, a source AD has no way of determining if it will be issued a visa without incurring the overhead of contacting the particular ACS in question.

The essence of the problem is that transit control is related to packet routing. Therefore, controls for transit should be incorporated into the route calculation itself, not only into the packet forwarding function.

Other network access control schemes such as SP3[44] and router packet filters [31] face the same limitation when it comes to controlling transit traffic, i.e., these schemes enforce controls on packet forwarding and do not provide information to the route computation[6].

In summary, Visa protocols and other network access control mechanisms are best suited for stub network control. Transit network control for large internetworks is more efficiently achieved by integrating policy considerations into the route computation process. After discussing policy

[5] More aggressive policies could also be implemented such as applying for group visas in advance of use to accommodate a collection of end-systems that have a need for communication. However, these may imply significantly more trust among the ADs and requires more careful consideration.

[6] SP3's access control policy, in particular, is endpoint-oriented. It is concerned mainly with determining whether or not two peers may communicate and what type of information they may exchange.

protocol as opposed to the network or link protocol. However, we argue that in the sense of the *end-to-end* argument, the network resources are *endpoints* to the extent they require protection in their own right. From this perspective, it is imperative to address the protection of network resource in addition to end-system protection.

If control is left to the end-systems, valuable stub-AD network resources may be consumed by unauthorized traffic. Rejecting packets at the end-system is *too late* from the perspective of network resource usage. Moreover, unrestricted network access increases the vulnerability of ADs to denial of service attacks in the form of packet storms. Finally, some ADs must control which routes are used to and from their internal end-systems; due to cost or security characteristics, for example. Because routing is a network level function, these controls must also involve network level entities and can not be left to transport session endpoints.

In summary, to the extent network resources require protection, the highest relevant endpoint is the network router and associated routing protocols. In this sense, the *end-to-end* argument supports implementing these controls at the network layer. For a more extensive discussion of this argument the reader is referred to [15].

3 Controlling Transit Internetwork Traffic

There are two basic approaches to controlling transit traffic. In the first part of this section, we discuss an approach based on the extension of traditional network access control mechanisms and identify its limitations. Subsequently, we consider alternative approaches based on integrating controls into internetwork routing.

3.1 Extending Network Access Controls

One potential method of enforcing transit policy enforcement is to extend existing stub network access control mechanisms to the generalized internetwork model. In this section, we discuss an extension of Visa protocols[13] that incorporates support for transit policy enforcement. Other network access control schemes are discussed briefly at the end of this section.

Visa protocols were developed originally to provide datagram-level control at AD boundaries.[4] Conceptually, a secret key is used by the communicating host to compute an unforgeable stamp. This stamp is placed in the datagram's header to assure the border-router that the transmission of this datagram across AD boundaries is authorized. The stamp is called a *visa*, by analogy with the stamp on a passport that allows a tourist to cross international borders. A unique visa is bound to each datagram in order to guarantee the authenticity and the integrity of the data.

A host on a visa-controlled network that wants to communicate across the AD boundary is initially required to engage in a high-level authorization/authentication dialog with an Access Control Server (ACS) on both local and destination ADs. The need for, and particulars of, this dialog are determined independently by the administration of each AD involved. When and if the communication is approved, the respective ACSs issue *visa-keys* to the requesting host. The

[4] In a connection oriented network a similar approach can be applied to stamp packets. However, the establishment of the visa key can be part of the connection setup and several of the datagram related design issues do not apply.

2.3 Internetwork Topology

Some routing protocols place restrictions on internet scale and topology, e.g., EGP[41]. Any inter-AD routing protocol should have the potential of supporting very large scale internetworking. We anticipate on the order of 10^5 ADs.[2] In an internet of such enormous size it would be unwise to design a protocol that relied on topological restrictions; enforcement would be near impossible. Consequently, one of our design goals is to allow for maximum degree of flexibility in regard to the configuration of the internetwork. The protocols discussed below do not place restrictions on the internetwork topology at the granularity of ADs.

Figure 1: Example of AD interconnection.

Figure 1 depicts an example of the AD interconnection. It resembles a traditional hierarchy of long haul, regional and campus (stub) networks. However, there are exceptions to the hierarchy in the form of lateral links. These exceptions to the otherwise regular topology are not dispensable and must be supported, perhaps at the expense of optimality. Absence of restrictions on AD interconnection allows us to accommodate this, or any other, topology.[3]

2.4 Network layer mechanisms

Many discussions of *network security* are actually discussions of end-system protection in a network environment, e.g., [46, 35, 23, 22, 13, 47]. One of our assumptions in the design of transit network controls is that both stub and transit ADs have valuable network resources that are themselves the subject of policy [15].

Initially, this may appear to be in violation of a well-known design principle, the *end-to-end* argument[43]. In the case of security, the argument suggests that end-system resources are best protected by the end-systems themselves, e.g., security services should be provided in the transport

[2] Although the majority will be stub ADs, our model assumes a large number of transit and hybrid ADs as well.

[3] Although the architecture works for arbitrary topologies, some topologies entail greater overhead in terms of route computation. For further discussion of topology see [14].

how the AD's packets travel to their destinations.

In some respects, the requirements for transit policy enforcement are simpler than those for stub policy enforcement. However, several factors complicate the implementation of the latter. First, in an internetwork, a packet may travel through a number of transit ADs on its way to the destination. Consequently, applicable policies from all transit ADs must be considered when a packet is being sent; whereas for control of stub resources, only the policies of the two end-point ADs need to be taken into account. In addition, transit control has to be reconciled with topology changes (routers or links going down). If in the middle of a connection any component of the route becomes disabled, entirely different policies may come into effect. Also, when a transit AD decides to account and/or bill for resource usage, coordination is required to pass charges back to the end points. Moreover, stub route selection criteria must be integrated with transit control policies to determine the appropriate routes. These factors add to the complexity of potential enforcement mechanisms.

Based in part on the difference in policies, and in part on the functionality required in any routing (i.e., transit) mechanism, transit and stub AD *mechanisms* also differ. By analogy with international travel, in most countries transit travelers are set apart from other visitors. They are issued special *transit* visas and are restricted in movement and length of stay. We discuss transit mechanisms further in later sections.

2.2.2 Policy Attributes

Network usage policies can be based upon a number of attributes:

- **Endpoint** policies place restrictions on the source and/or destination of traffic.
- **Path** policies place restrictions on other ADs of the path in addition to the source and destination ADs.
- **Security** attributes express requirements for authentication, data integrity, replay detection and privacy.
- **Temporal parameters** include restrictions on usage based on time of day, day of the week or other time-related parameters.
- **Quality of Service** policies discriminate according to the service parameters (e.g., delay, throughput) made available to different users.
- **Accounting/Billing** policies express conditions related to charging and accounting.

A typical policy statement can be based upon several policy attributes. For example, the policy statement:

transit voice traffic from AD_a is accepted between 2 and 6 am with a per packet charge

applies to transit traffic and combines application protocol, temporal and accounting/billing attributes. Further examples of policy types can be found in [14].

2.1 Administrative Domains

An internetwork is composed of a number of Administrative Domains, or ADs. An AD is defined as a collection of network resources under control of a single administrative entity [19]. It is important to distinguish between two dominant types of ADs: *stub* and *transit*. *Stub* ADs are interested mainly in communication with other stub ADs, i.e., providing communication for their constituent end-systems. A campus network is an example of a stub AD. *Transit* ADs are involved in providing transit service for traffic between stub ADs. NSFNET is an example of a transit AD. Finally, there are also hybrid ADs that combine transit service with end-point communication (e.g. USC ISI).

2.2 Policies

As frequently happens with a new concept, an analogy can lead to better understanding of the problem at hand. One interesting analogy is to view ADs as sovereign countries, each with a specific set of foreign policy statements regarding interaction with foreign entities (other ADs). For example, a country may have policies restricting foreign visitors to specific areas or restricting travel privileges of the local populace when visiting foreign countries. Countries may also have specific policies pertaining to transit travelers, e.g., restricting entry on the basis of the traveler's itinerary. Security policies regarding international travel can express policy as to passport and visa requirements, length of stay, etc. Accounting or billing policies may concern, for example, visa fees or departure taxes.

ADs can express similar policies regarding communication with external entities, e.g., restrict internal systems available for external access or restrict external systems available for internal access. Transit traffic may or may not be allowed, or it may be restricted to specific source and/or destination ADs or end-systems. Policies can also embody security requirements, e.g., authentication and authorization for inter-AD traffic, as well as accounting and billing conditions [14].

Network level policies are primarily concerned with unauthorized access to network resources, denial of service, and inappropriate accrual of communication-related charges. These threats can all come about through attacks on the authenticity and the integrity of internetwork packet traffic. Some concerns are of greater importance to stub networks and others, to transit networks.

2.2.1 Stub and Transit Policies

Due largely to the nature of service provided, stub and transit ADs tend to express different policies. Policies expressed by stub ADs, for the most part, serve to protect internal resources from external access, while those expressed by transit ADs tend to be cost-related. Another way of making this distinction is to observe that transit ADs, by virtue of providing transit service, are inherently more *open* than their stub counterparts. Furthermore, subversion of transit AD's policies will, in the worst case, result in denial of service, whereas subversion of stub network polices can potentially disrupt the end-systems themselves. Another reason for separating the respective policies is the difference in accounting and billing requirements. Stub ADs are more likely to bundle communication costs into billing for end services, if any such billing occurs. Transit ADs are more likely to charge for the communication itself. Finally, stub AD policies include route selection criteria, which dictate

1 Introduction

A collection of autonomous networks can be interconnected to form a single internetwork using current internetworking protocols (e.g., IP [38]). Although the interconnection promotes transparent flow of information across autonomous network boundaries, participant organizations would like to maintain their autonomy by independently expressing and enforcing network usage policies. In particular, the enforcement of access control policies with respect to the organizational network resources becomes of primary concern.

Network access control methods that restrict the information flow between end-systems on individual networks have been demonstrated [13, 31, 5]. Moreover, end-systems employ a range of operating system and application level mechanisms that restrict access to data and resources. However, controlling access to **transit** network resources (such as routers and links) requires additional protocol support because of the need to coordinate routing decisions among all intervening networks. Organizations cannot simply enforce policy restrictions on a unilateral basis at packet forwarding time. Internetwork **routing** decisions must be made according to policy-related parameters such as access rights and cost, in addition to the traditional parameters of connectivity and delay [10, 6]. Consequently, policies pertaining to network resources must either be implicit in the topology of an internetwork, or advertised to the anticipated resource users. Only then can entities throughout the internetwork determine the logical, *policy-based* connectivity of an internetwork and compute valid routes.

This paper addresses the design of protocols for **secure** control of transit traffic on an internetwork. Transit control protocols can be designed with varying levels of security. In some environments, relatively vulnerable protocols may be used in conjunction with *post facto* detection mechanisms. Most of the work in policy-based protocol development is being conducted with such environments in mind [10, 6, 16, 28]. This paper addresses environments where post facto detection is not adequate or possible in a timely manner. In particular, we address the design and costs (i.e., performance and manageability) of incorporating more defensive, *preventative* security measures into the protocols for controlling internetwork traffic.[1]

This paper is organized as follows. As further introduction, Section 2 discusses the interconnection of autonomous networks. Section 3 begins by considering the extension of network access control methods to control transit internetwork traffic. In response to some fundamental deficiencies, the remainder of Section 3 describes the role of the so-called Policy Routing protocols in providing transit control. Next, Section 4 outlines the security concerns particular to the Policy Routing protocols. The discussion of security is continued in the remaining sections on secure protocol design (Section 5) and cost assessment (Section 6). Section 7 summarizes our discussion and proposes areas for future work.

2 Interconnection of Autonomous Networks

In order to provide appropriate background for the subsequent discussion, this section defines our terminology and assumptions regarding internetwork environments, policies, and protocol design principles.

[1]Portions of this paper first appeared in [12].

Secure Control of Transit Internetwork Traffic*†

Deborah Estrin Gene Tsudik

Computer Science Department
University of Southern California
Los Angeles, California 90089-0782
estrin@usc.edu tsudik@jerico.usc.edu

December 12, 1990

Abstract

When independent administrative domains (ADs) interconnect their networks, usage control mechanisms are needed to preserve the autonomy of each AD. Neither traditional network access control methods nor current internetwork routing protocols are well-suited to the enforcement of network usage policies. Consequently, new *policy sensitive* inter-domain routing protocols are currently under development. While these protocols are designed to enforce network policies, they raise new security-related concerns.

This paper explores the design of transit control mechanisms for sensitive environments and investigates the performance overhead of the proposed mechanisms. After evaluating the application of stub-network access control and policy routing techniques, we describe and evaluate a secure policy routing protocol for transit ADs.

Categories and Subject Descriptors: C.2.0 [**Computer-Communication Networks**] : General - *Security and Protection*; C.2.2 [**Computer-Communication Networks**]: Network Protocols - *Protocol Architecture*; D.4.6 [**Operating Systems**]: Security and Protection - *Access Controls, Authentication, Cryptographic Protocols*; D.4.7 [**Operating Systems**]: Organization and Design - *Distributed Systems*; D.4.8 [**Operating Systems**]: Performance; K.4.3 [**Computers and Society**]: Organizational Impacts;

General Terms: Design, Management, Performance, Security;

Additional Key Words and Phrases: Enforcement of Network Policies, Control of Internetwork Traffic, Network Resource Management, Network Security Threats, Data Integrity, Design of Secure Protocols, Cost of Secure Protocols;

*This research was supported in part by the National Science Foundation Presidential Young Investigator award and matching funds from ATT,GTE, and NCR.

†An earlier version of this paper was presented at the DIMACS Workshop on Connections between Distributed Computing and Cryptography, October 1989, Princeton, NJ.

Secure Control of Transit Internetwork Traffic

TR-90-14

Deborah Estrin Gene Tsudik

Computer Science Department
University of Southern California
Los Angeles, CA 90089