An Efficient Signature Scheme for Route Aggregation

Suresh Chari[1] Tal Rabin[1]
Ronald Rivest[2]

[1]IBM Thomas J. Watson Research Center
Hawthorne, NY 10532.
Email: {schari,talr}@watson.ibm.com

[2]Laboratory for Computer Science,
Massachusetts Institute of Technology
Cambridge, MA 10532.
Email: rivest@lcs.mit.edu

Abstract

The strongest security guarantees for routing protocols can be obtained by the use of signatures on information in routing messages. The goal of digital signature mechanisms is to be unforgeable *i.e.* signatures of new messages cannot be computed from known signatures of other messages. This raises an interesting problem for hierarchical routing protocols such as OSPF where specially designated routers, have to aggregate information from one level of the hierarchy into the next. If digital signatures are used, these routers need to compute the signature of aggregated information, possessing only the signatures on component information.

In this paper, we address this problem by describing a new and efficient signature scheme to sign the nodes of a full binary tree with carefully defined algebraic properties: Given the signatures of the children of a node, it is easy to compute the signature of the node. While the scheme is not unforgeable in the traditional sense, we guarantee that no adversary will be able to forge the signature of a node without seeing or computing the signatures of all its children. We use this scheme to derive efficient solutions to the problem of signing aggregated routes in hierarchical routing protocols.

1 Introduction

The correctness of information exchanged by routing protocols such as the Border Gateway Protocol (BGP) [RL95] and the Open Shortest Path First (OSPF) [Moy98] is of paramount importance to the functioning of the global Internet, since they define how packets are routed to remote destinations. Due to its significance, substantial research effort has focused on ensuring correctness of this information via cryptographic mechanisms.

From a routing perspective, the Internet is divided into independent regions called autonomous systems (AS). Within a single AS, routing information is exchanged using *interior routing protocols*. The commonly used protocol is OSPF [Moy98], which is an hierarchical routing protocol: the AS is further architecturally divided into domains called areas each with a unique ID. Routers within an area propagate routing information using messages called link–state updates. These updates are sent by the originating router to each of its neighbors, which then forward the update and point of origination to their neighbors and so on. Thus each router in the area can collect information about the entire topology of the area independent of other routers. This information is then used by the router to build a shortest path tree to all destinations with itself as the root. Routers which straddle two or more areas are called *area–border routers*. These routers are responsible for propagating link–state updates from within an area to area–border routers of other areas and to similarly import routes from other areas. Figure 1 is an abstract depiction of an OSPF areas and routers. Area–border routers do not directly propagate all the updates from routers in the OSPF area: they perform aggregation functions and issue summaries of information. One of the aggregation functions performed is to advertise the shortest prefix describing the networks directly connected to routers in this area.

Perlman [Per88] identifies security exposures of abstract routing protocols and proposes solutions to mitigate these exposures. The strongest security requirement stated is protection against Byzantine failures where an arbitrary set of routers could fail and behave maliciously due to subversion or misconfiguration. Simply stated, we would like that any message claiming a route to a destination must be verifiably authorized by appropriate authorities. Subsequently, there have been a number of proposals to integrate signatures into actual routing protocol messages (see for example [MBW97]). The proposed countermeasures to secure link–state protocols is for area administrators to sign links between routers and networks and for router links to be signed by the routers themselves. This guarantees

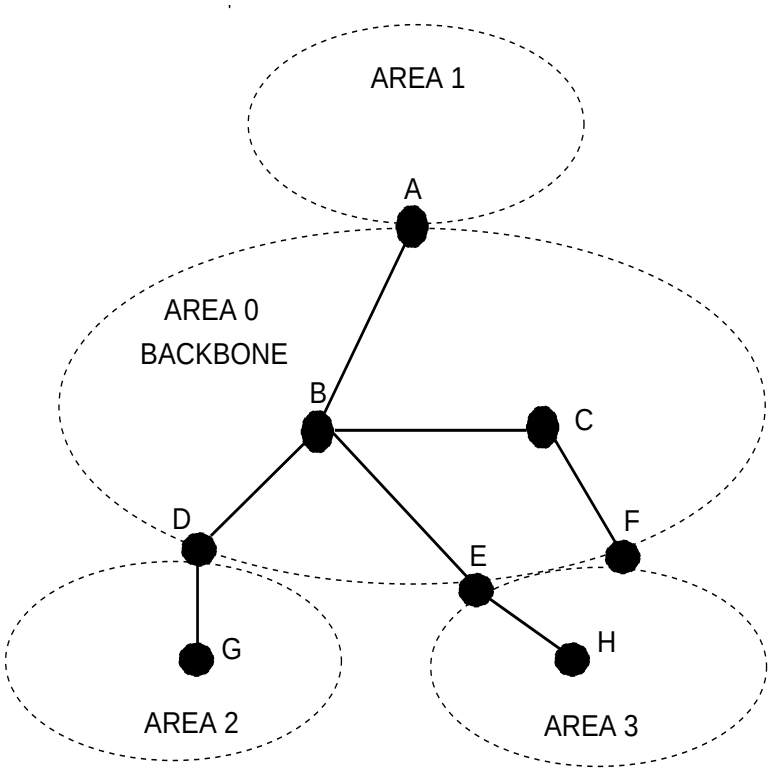

Figure 1: Areas and Routers in an OSPF Autonomous System.

that even under Byzantine failures no router can falsely claim reachability to networks which are not authorized.

Yet, signing messages and authorizations does not work well when area–border routers propagate this information to other areas since they aggregate routes within the area. If link–state updates claiming reachability to networks are signed in a standard manner (for example using the signature mechanisms defined in [IEE]), then the area–border router would need to concatenate the signatures to prove reachability to the aggregated route clearly losing the advantage of the aggregation. In addition, this also exposes the inner topology of the OSPF area to routers in other areas which runs counter to the design philosophy of hierarchical routing. This problem has been noted in the literature [MBW97] and left as an open problem and several system security style measures such as multiple area–border routers verifying each other's summaries are suggested.

In this paper, we provide a solution to this problem of securing reacha-

bility information in a manner which permits secure route aggregation. We introduce a *Prefix Aggregation Signature Scheme* which specifies first the method by which the reachability information is initially signed by entities such as the administrative authority of the area. Once these signatures have been put in place properly, possibly at the time of initial router configuration, the operation of aggregating them is a simple operation. We emphasize that our solution also provides great efficiency for the major operations of the initial signing of the information, the aggregation and the verification of a signature.

The idea of the solution is to build a full binary tree and associate with a subnet a node in the tree. The leaves of the tree correspond to individual hosts of the Internet and internal nodes correspond to subnets in an obvious manner. An attestation to the reachability of the subnet would be a signature on the node of the tree. The signature scheme is designed so that anyone can easily compute the signature of a node given signatures of both its children, enabling the area–border to independently compute the signature of the aggregated subnets.

By definition, the scheme is not unforgeable: Given signatures on the children of a particular node anyone can compute signature on the node itself. However, the scheme is secure in a strong sense: *No adversary can compute the signature on any node without the signatures on its children.* In the routing application, this means that no router can claim a verifiable route to any subnets without verifiable routes to its component subnets.

Our signature scheme, is in a sense, *homomorphic*: Given signatures on some messages, anyone can compute signatures on messages which satisfy a precise relation with the chosen messages. Recently, there has been interest in these types of signature schemes and a number of schemes, homomorphic under different relations, have been proposed [MR02, JMSW02] for special applications. We believe that such signature schemes which satisfy special algebraic properties tailored for specific applications will increasingly become popular. In the domain of routing protocols, our scheme is orthogonal to a number of proposals which optimize the use of traditional signatures by using other cryptographic techniques such as one–time signatures [HPT97] and signing streams of updates [Zha98].

The paper is organized as follows: Section 2 describes the construction of the scheme and the mechanics of its application to the problem of route aggregation. In Section 3 we prove that our scheme is secure. Section 4 describes related work in this area and we conclude in section 5.

2 New Signature Scheme

In this section we describe the details of our new signature scheme. The description of the signature scheme itself is very general. Later we also address technical and practical details on how such a scheme can be used to solve the problem of route aggregation in OSPF areas. As mentioned earlier we consider a full binary tree and associate in a straightforward manner a binary label to each node. Binary labels on the nodes of this tree correspond in an obvious way to addresses of hosts and subnets. This will be considered the lexicographic labeling of the tree on top of which we construct an aggregated (or multiplicative) labeling. With this new labeling, the aggregated signature scheme will be defined by applying a homomorphic signature mechanism, such as the RSA operation [RSA78], to the aggregated labeling.

First, the basic lexicographic binary labeling of the tree is defined as follows:

Definition 1 *Given a full binary tree define the lexicographic labeling of the nodes as the canonical binary labeling of the nodes. The root is labeled with the empty string and the right and left child of a node with label x are labeled $x0$ and $x1$ respectively.*

Given a tree with nodes labeled lexicographically we proceed to define an aggregated labeling $\mathcal{L}$ which will have the property that the labeling of a node is the product of the labeling of its two children.

Definition 2 *Fix a cryptographic hash function $H : \{0,1\}^* \mapsto Z_n^*$. Define the aggregated label of the root as a random element of Z_n^*. For other nodes $\mathcal{L}$ is defined recursively as follows:*

$$\mathcal{L}(x0) = H(x0)$$

$$\mathcal{L}(x1) = \frac{\mathcal{L}(x)}{\mathcal{L}(x0)} \bmod n$$

Given such an aggregated labeling we define our aggregated signature scheme as applying the RSA operation [RSA78] on this label.

Definition 3 Prefix Aggregation Signature scheme: *Let n, e and d be the modulus, public exponent and private exponent of an RSA cryptosystem. Given a binary tree define, let the hash function H, used to define the labeling $\mathcal{L}$, be a hash function onto Z_n^*. Define the signature on a node with lexicographic labeling x as $(\mathcal{L}(x))^d \bmod n$, i.e. $\sigma(x) = (\mathcal{L}(x))^d \bmod n$.*

Since the aggregate labeling $\mathcal{L}$ satisfies the property that

$$\mathcal{L}(x0)\mathcal{L}(x1) = \mathcal{L}(x)$$

and the RSA signature scheme satisfies the condition that $\sigma(x)\sigma(y) = \sigma(xy)$, we can immediately derive that the signature on node x can be computed given the signatures on nodes $x0$ and $x1$. This is done with a single modular multiplication operation which can be done fairly efficiently in practice.

2.1 Applicability to Signing OSPF Information

To secure claims of reachability to subnets in OSPF, we propose that any router claiming a link to a subnet must carry a signature of the area authority on this information. Through the standard link–state update mechanisms, the area border router accumulates the signatures of the authority on all subnets within the area. After verification of these signatures, it can advertised the reachability of the aggregated route and publish the computed signature of the authority on the aggregated routes. For each level of aggregation the area–border must perform a modular multiplication of the component signatures only knowing the public modulus used to sign the reachability to individual subnets. Since OSPF areas are defined loosely as components under the same area administrative authority, it is practical for a single authority (single RSA public–key pair) to sign the reachability statements for all the routers in the area. We believe that this can be easily deployed by including the signatures as an extra operation during the initial configuration of the routers.

There are several points to keep in mind in the application of such a signature scheme. Firstly, there is an added performance penalty in generating signatures: the authority has to compute the labels of the nodes before the private key operation. Even though the label is defined recursively, note that at most $\log(N)$ computations of the hash function H are needed to compute a label where N is the number of leaves. In the case of IP address subnets, this requires about 24 computations of H since, typically, subnets are defined with the mask of 255.255.255.0. Hash computations are typically one of the fastest cryptographic operations and we believe that this will pose no performance problems at all.

Note that, by design, the signatures can not be aggregated beyond the boundaries of the OSPF area since the signatures on reachability information are computed using different RSA parameters in different areas. This fits well with the modular decomposition of the OSPF autonomous system into areas which are loosely under the same administration.

One drawback which this scheme shares with all other signature schemes is that once a router has the signature of the authority on advertised routes, it can continue to do so even if there is no connectivity to the network. This is exacerbated in our scheme in two ways: Firstly, there is no way to indicate the time duration for which the attestation of the route is valid. In other signature schemes, the authority can also sign the time upto which the attestation is valid. Secondly, the area–border can claim reachability of the aggregated route even if any or all the component subnets are no longer reachable. Thus, revocation of routes is more difficult with this scheme. However, we wish to point out that, like all other signature based schemes, a router can at any time, only claim reachability to routes which it could validly claim at some point before. At no time can a router claim reachability to routes which it was not authorized to claim.

3 Proof of Security

To prove the security of our signature scheme we need to precisely capture what the security requirements are. Clearly, the Prefix Aggregated Signature scheme is not existentially unforgeable due to the requirements of the design. Intuitively, we want that an adversary attacking the scheme would not be able to generate a signature for a node, unless it has been given the signature of the nodes' two children, or the parent and the sibling's signatures[1]. Stating this some what graphically, in any triple of nodes of parent and its two children, we assume that if any two signatures are known then so is the third. Simply stated, given signatures on any two of these nodes anyone can compute the signature of the third and the security requirement states that the adversary can do nothing more.

Given signatures on some nodes of the tree, the following definition captures the set of the nodes whose signatures can be derived from the known signatures.

Definition 4 *Given a binary tree with a lexicographic labeling, and a set of labels I of nodes in the tree, define the aggregated closure, $\mathcal{C}_I$, of I as the smallest set that contains I and which is closed under the following operations:*

- *For every pair of labels $x0, x1 \in C_I$ add label x to C_I; that is, add the parent when the two children are in the set*

[1]Note, that in the application we would only consider a signature as a forgery if it was on a node whose ancestors have not been signed. But here we will prove a stronger result.

- *For every pair of labels $x, x0 \in C_I$ add label $x1$ to C_I;*
- *For every pair of labels $x, x1 \in C_I$ add label $x0$ to C_I;*

With this, we can precisely state the security guarantee offered by the scheme. Given a tree whose nodes are signed with our Aggregated Signature Scheme, we say that an adversary $\mathcal{A}$ has forged a signature in the scheme if given the signatures on a set, I, of q nodes, the adversary can produce a valid signature of a node not in $\mathcal{C}_I$, the aggregated closure of I.

Theorem 1 *The Aggregated Signature Scheme is unforgeable against an adaptive chosen message attack.*

Proof We show that if there is an adversary $\mathcal{A}$ who can break the Aggregated Signature Scheme, for a tree with N nodes and q signature queries, then we can build a forger $\mathcal{F}$ who can mount a selective attack on the underlying RSA signature scheme with probability of success $\frac{1}{cq}$, where c is a small constant.

The forger $\mathcal{F}$ builds a tree with an aggregated labeling, which it will transfer to $\mathcal{A}$. For some subset of the nodes of the tree $\mathcal{F}$ will not know the signature on the labels and for the rest of the nodes the aggregated labeling is defined so that $\mathcal{F}$ knows the signatures. By our assumption, $\mathcal{A}$ can ask adaptively to receive q signature on nodes of the tree. If $\mathcal{F}$ knows the answer to the query it will transfer the signature to the adversary, otherwise it aborts. If the queries were completed successfully and $\mathcal{A}$ produced a forgery then with some probability $\mathcal{F}$ will produce a selective forgery for the underlying signature scheme. $\mathcal{F}$ succeeds in forging if two things occur, first all of $\mathcal{A}$'s queries fall outside K, the set for which $\mathcal{F}$ does not know the signatures, and the second, if $\mathcal{A}$'s forgery is on an element in K. This means that the size of the set K must yield a probability of success which is polynomial in q. Thus, if we set the size of K to N/q, where q is the number of queries submitted by $\mathcal{A}$, we would get that the probability of $\mathcal{F}$ generating a forgery is:

$$(1 - 1/q)^q \times ((N/q)/N) = \frac{1}{cq}.$$

where c is a constant.

In the rest of the section we describe the details of the proof. The forger $\mathcal{F}$ is given an RSA public key (n, e) and a message m for which it needs to generate the signature $\sigma = m^d \bmod n$. The Aggregated tree has N leaves. $\mathcal{F}$ will fix a set K of N/q nodes on which it will *not* know the signature.

The nodes in the set K satisfy the following properties: If a node x is in K then there exists a path from x to a leaf whose nodes are all in K. For every node $xb \in K$ exactly one of the nodes $x\bar{b}$ or x is in K. The construction of such a set is straightforward and we omit the details.

Now $\mathcal{F}$ will start the process of generating the Aggregated labeling for the tree. If the root is in K it will choose a random value r_ϵ and set $\mathcal{L}(\epsilon) = mr_\epsilon^e \bmod n$. Otherwise, it will choose a random s_ϵ (the "signature" on the node) and set $\mathcal{L}(\epsilon) = s_\epsilon^e \bmod n$. It will proceed to mark the tree in the following manner: given a node x and its aggregated label $\mathcal{L}(x)$

1. If the signature s_x on $\mathcal{L}(x)$ is known (i.e. $x \notin K$) then

   - If $x0, x1 \notin K$ then pick $s_{x0} \in_R Z_n^*$ as the signature on the label, set $\mathcal{L}(x0) = s_{x0}^e \bmod n$,
     $\mathcal{L}(x1) = \mathcal{L}(x)/\mathcal{L}(x0)$,
     and $s_{x1} = s_x/s_{x0}$
   - If $x0, x1 \in K$ then pick $r_{x0} \in_R Z_n^*$ and set $\mathcal{L}(x0) = mr_{x0}^e \bmod n$, and $\mathcal{L}(x1) = \mathcal{L}(x)/\mathcal{L}(x0)$

   Note that these are the only possible options, due to the fact that among a parent and its two children, either the signature is known for all nodes, or for at most one.

2. The node $x \in K$, i.e. only the aggregated label $\mathcal{L}(x)$ is known, furthermore it holds that $\mathcal{L}(x) = mr_x^e \bmod n$. We will show that for all $x \in K$ this representation of the aggregated label can be preserved. By the construction of K we have that either $x0$ or $x1$ are in K (but not both).

   - $x0 \in K$ then pick $s_{x1} \in_R Z_n^*$ which will be the signature for the node $x1$, set $\mathcal{L}(x1) = s_{x1}^e \bmod n$, $\mathcal{L}(x0) = \mathcal{L}(x)/\mathcal{L}(x1)$. Note that the aggregated label on $x0$ is $\mathcal{L}(x)/\mathcal{L}(x1) = mr_x^e/s_{x1}^e = m(r_x/s_{x1})^e$ and thus if we set $r_{x0} = r_x/s_{x1}$ we preserve the representation of the aggregated label.
   - $x1 \in K$ then do the same as above: pick $s_{x0} \in_R Z_n^*$ which will be the signature for the node $x0$, set $\mathcal{L}(x0) = s_{x0}^e \bmod n$, $\mathcal{L}(x1) = \mathcal{L}(x)/\mathcal{L}(x0)$ and $r_{x1} = r_x/s_{x0}$

We first need to show that the distribution of the aggregated labels on the tree is identical to a labeling which would have been generated in the proper manner. This is easily verified by seeing that each label whose value

was generated by either choosing an mr_x^e or s_x^e is random and properly distributed. And if an aggregated label was computed using the two other labels then it is also properly distributed.

Now we proceed to show how the forger will generate its forgery. Assume that $\mathcal{F}$ does not abort the interaction with $\mathcal{A}$, that is for all the queries that $\mathcal{A}$ asked $\mathcal{F}$ knows the signature (meaning that all the queried nodes are not in K). Furthermore, assume that the adversary returns a signature s on a label $\mathcal{L}(x)$ for $x \in K$. Recall that for each node $x \in K$ the forger knows a representation of the form $\mathcal{L}(x) = mr_x^e \bmod n$. As it has received the signature s we have that $s = \mathcal{L}(x)^d = (mr_x^e)^d = m^d r_x \bmod n$. The forger knows r_x and can extract $m^d \bmod n$ which is a signature on the message m and it outputs this value as the forgery.

The instance where the signature returned by $\mathcal{A}$ is on a node not in K will be added to the failed runs.

4 Related Work

The context for the work described in this paper is special signature schemes which are optimized for particular applications and in particular for securing routing protocols. For routing protocols a number of techniques such as one–time signatures [HPT97], stream signatures [Zha98] have been proposed as replacements for traditional digital signatures. These are primarily intended for purposes of efficiency and do not specifically address the problem of route aggregation by area–border routers. Thus our work addresses an aspect of efficiency which is orthogonal to that addressed by these schemes.

Traditionally, cryptographic signature schemes have focused on schemes which are existentially unforgeable as they offer strong security guarantees. In fact, multiplicative cryptosystems such as the RSA were considered to be weak due to many possible attacks on them [MvOV96]. Recently, however, there has been considerable interest in signature schemes which have special properties. One of the first proposals was the scheme of Micali and Rivest [MR02] for signing the edges of an undirected graph such that given the signatures on some edges anyone can compute the signature on any path in the transitive closure of the edges. This scheme could also have potential applications to securing routing protocols like BGP where routers advertise paths to remote destinations. However, like the scheme presented in this paper, the signature is done with a common RSA public–key pair. In the context of BGP, this limits the application since the routers (vertices) of the graph belong to different administrative domains. More recently

Johnson *et al.* [JMSW02] have studied special signature schemes which are homomorphic under different operations. They provide a signature scheme to sign arbitrary binary strings which is *redactable i.e.* given the signature of a string x anyone can compute the signature of strings which are a substring of x. They also discuss related questions regarding the feasibility of other homomorphic signature schemes.

5 Conclusion

In this paper we described a new signature scheme motivated by a practical problem arising from the security of routing protocol messages. Although the scheme is not *existentially unforgeable* it offers strong security guarantees while offering nice algebraic properties which make it applicable to the routing problem. While a few technical details need to be added to routing protocols we believe that our scheme can be used to prevent a number of attacks on these protocols such as subverted routers proclaiming false routes to destinations. In a broader context we believe that the research into special signatures optimized for particular applications promises to be an exciting area of research.

6 Acknowledgments

We wish to thank Charanjit Jutla and Hugo Krawczyk for fruitful discussions on various aspects of the signature scheme.

References

[HPT97] R. Hauser, A. Przygienda, and G. Tsudik. Reducing the cost of Security in Link–State Routing. In *Proceedings of the ISOC Symposium on Network and Distributed System Security*, February 1997.

[IEE] IEEE. IEEE P1363: Standard Specifications for Public Key–Cryptography. Documents available online at `http://grouper.ieee.org/groups/1363/index.html`.

[JMSW02] Robert Johnson, David Molnar, Dawn Song, and David Wagner. Homomorphic Signature Schemes. In *Proceedings of the RSA Security Conference Cryptographers Track*, February 2002.

[MBW97] S. L. Murphy, M.R. Badger, and B. Wellington. OSPF with Digital Signatures. Internet Engineering Task Force (IETF) Request for Comments (RFC) 2154, June 1997.

[Moy98] J. Moy. OSPF Version 2. Internet Engineering Task Force (IETF) Request for Comments (RFC) 2328, April 1998.

[MR02] Silvio Micali and Ronald Rivest. Transitive Signature Schemes. In *Proceedings of the RSA Security Conference Cryptographers Track*, February 2002.

[MvOV96] Alfred Menezes, Paul van Oorschot, and Scott Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.

[Per88] Radia Perlman. *Network Layer Protocols with Byzantine Robustness.* PhD thesis, Department of Electrical Engineering and Computer Sciences, Massachussets Institute of Technology, August 1988.

[RL95] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP–4). Internet Engineering Task Force (IETF) Request for Comments (RFC) 1771, March 1995.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public–key cryptosystems. In *Communications of the ACM*, volume 21, pages 120–126, 1978.

[Zha98] K. Zhang. Efficient Protocols for Signing Routing Messages. In *Proceedings of the ISOC Symposium on Network and Distributed System Security*, February 1998.