Identity-Based Registry for Secure Interdomain Routing

E-yong Kim
School of Computer Science and Engineering
Seoul National University
Seoul 151-744, Korea
eykim@theory.snu.ac.kr

Li Xiao
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
lixiao@cs.uiuc.edu

Klara Nahrstedt
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
klara@cs.uiuc.edu

Kunsoo Park
School of Computer Science and Engineering
Seoul National University
Seoul 151-744, Korea
kpark@theory.snu.ac.kr

ABSTRACT

The current Internet has no secure way to validate the correctness of the routing information. We suggest a mechanism that supports secure validation of routing information in the interdomain routing protocol of the Internet. Our mechanism focuses on alleviating obstacles which previously prevent the complete and correct construction of the Internet routing information. In particular, we propose an *identity-based Registry with Authorized and Verifiable Search* (RAVS) so that routing information can be constructed securely. We construct an efficient RAVS scheme and prove its securities in the random oracle model. By our scheme, the routing information can be securely stored and tested without revealing contents of both the registry and the search query. Furthermore, our registry is verifiable and its correctness is guaranteed. Only the legal autonomous system (AS) can construct the valid registry and the single compromised AS can be detected. Our experiment shows that our RAVS scheme can be implemented efficiently and the incurred overhead, in terms of time and space, is acceptable in practice.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*security and protection*; E.3 [**Data Encryption**]: Public key cryptosystems

General Terms

Security, Theory

Keywords

Identity-based registry, authorized search, verifiable search, interdomain routing, BGP

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ASIACCS '06, March 21-24, 2006, Taipei, Taiwan.
Copyright 2006 ACM 1-59593-272-0/06/0003 ...$5.00.

1. INTRODUCTION

The Internet routing infrastructure is a large distributed system composed of many independently managed networks, called Autonomous Systems (AS). To find routes across multiple domains, ASes exchange routing information using an interdomain routing protocol. The de facto standard of interdomain routing protocol is Border Gateway Protocol (BGP) [28], a path vector protocol. BGP peers exchange routing information incrementally using UPDATE messages. BGP is developed under the assumption that the UPDATE message advertised by peers is correct. However, this assumption is challenged in the current Internet environment. This is because BGP is vulnerable to many kinds of attacks [24]. Even a simple misconfiguration can disrupt significant parts of the Internet [8]. Therefore, it is important to reduce the vulnerability of BGP to make the Internet routing more robust.

There exist quite a few proposed solutions for addressing the vulnerability of BGP [19, 20, 35, 25, 36, 16, 21, 17, 32, 33]. Most approaches are difficult to be adopted to the Internet due to modifications of existing protocols or routing message formats, cost of heavy operation, and lack of backward compatibility. At present, route filtering is an effective way to address BGP vulnerabilities by removing incorrect or malicious BGP UPDATE messages and is widely deployed in the current Internet. In order to build correct filters, ASes should have the knowledge about the policies of the global Internet. Generally, this knowledge is provided by the Internet Routing Registry (IRR), the set of 50+ databases of routing policy information [18].

The IRR records routing policies and topological information for all ASes and moreover this information can be used by ASes to validate the BGP UPDATE messages. For example, in Figure 1, all ASes submit their peering relationships to the IRR. If AS5 receives a route from AS4 that claims it has the direct path to AS1, AS5 can identify AS4 is misbehaving by checking with the topology information in the IRR and reject the route. In order to make this process dependable, it is crucial to have the information in the IRR complete and correct. However, the IRR information is not well-maintained and updated. The reason is that ASes

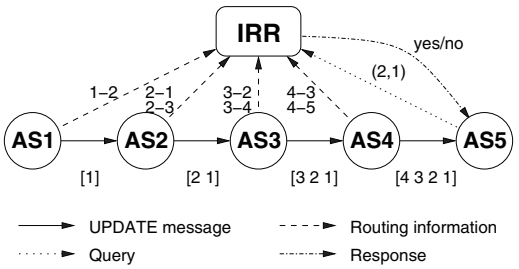

Figure 1: A simple illustration of the IRR in operation. Each AS submits its routing information to the IRR. For instance, AS2 registers its neighbor information AS2-AS1 and AS2-AS3 to the IRR. AS5 can issue a query to the IRR about the existence of routing information between AS2 and AS1 in the received UPDATE message [AS4 AS3 AS2 AS1]. The IRR responds with 'yes' because it has that routing information.

consider their business relationships, policies and topology information to be confidential. At present, there is no authorization of database queries to the IRR and this sensitive information is not protected. Moreover, the information in the IRR can be forged by an adversary. Therefore, making IRR secure is required to address the vulnerabilities in BGP routing.

However, the security of the IRR is not well studied and supported. Our purpose of this paper is to build the routing information database that supports authorized and verifiable search. There are three requirements on building such secure database. First, only the authorized ASes can query the IRR database. Second, no ASes can submit other ASes' routing information. Finally, the IRR database can be validated for its correctness. By ensuring the above requirements, the sensitive information in the IRR is well protected. Thus, ASes have incentives to contribute their routing information and to make the IRR information complete.

1.1 Our Approach

We take the cryptographic approach in this paper. In the cryptographic research area, there have been studies dealing with searching on encrypted data [30, 14, 12, 6, 5, 34]. It enables an untrusted server to store and search encrypted data without revealing any other information about data. When we build encrypted database of the Internet routing information to substitute the IRR, such techniques can be used. However, we are not able to apply the previous searchable encryption schemes directly to the IRR. There are multiple queriers and the information provider has no prior knowledge on the possible queriers. That is, the provider has no prior knowledge which keys he has to use for encrypting data.

We construct an identity-based registry with authorized and verifiable search (RAVS) scheme. We build the Internet routing information in a centralized repository[1] using our registry scheme. Our scheme is based on the three concepts: identity-based registry, authorized search, and verifiable search.

[1]Conceptually, we assume that it is centralized throughout the paper. In practice, it can be deployed in a distributed fashion like the current IRR servers.

- **Identity-based registry**: We make use of identity-based cryptosystem [29] because each AS has its own AS number as the identity in the Internet. The AS number is the public key of the corresponding AS in our registry scheme. Therefore, we do not need to depend on the PKI-based certificates. We use AS number and public key interchangeably in the rest part of the paper.
- **Authorized search**: Our scheme supports the *authorized search* by introducing a new third party, named the Search Permission Generator (SPG). This fully-trusted server is similar to the Private Key Generator (PKG) in the identity-based cryptosystem in terms of the function and the security. A legitimate AS can obtain permissions for searching registries from the SPG, and then it can query the registry database using these permissions. This ensures the first requirement of the IRR – only authorized AS can query to the IRR.
- **Verifiable search**: Our scheme supports the *verifiable search*. The registry provider constructs a registry using its own private key and submits it to the IRR. Other ASes can verify this registry with the provider's AS number. This guarantees that the registry should be constructed only by the right AS having the corresponding private key. Hence, the second requirement of the IRR is thus guaranteed – no ASes can submit other ASes' routing information.

ASes can validate the correctness of the IRR database using the network topology's mutuality (see Section 6 for details). A single compromised AS can forge an invalid registry with the right private key and submit the incorrect routing information to the IRR. Other ASes can detect this incorrect information by using mutual relation between normal and misbehaving ASes.

Our scheme overcomes the security vulnerabilities of the IRR. It helps ASes to submit their routing information such as network topology to the IRR without worrying about disclosure of their private information to unauthorized parties. It also enables ASes to verify the validity of received routing messages based on the correct and dependable information in the IRR.

1.2 Our Contributions

Our contributions are summarized as follows:

- We define an identity-based registry with authorized and verifiable search (RAVS) scheme, and construct an efficient RAVS scheme.
- We formulate security models for RAVS scheme known as semantic security against adaptive chosen keyword and identity attacks (IND-ID-CKA) and existential unforgeability against chosen keyword and identity attacks (EUF-ID-CKA). We then prove our RAVS scheme is IND-ID-CKA secure and EUF-ID-CKA secure in the random oracle model.
- We show how to securely detect various malicious attacks against BGP using our RAVS scheme. Our experimental results show that our scheme can be implemented efficiently and the incurred overhead, in terms of time and space, is acceptable in the Internet.

From the practical point of view, the approach based on the RAVS scheme has several advantages. First, our method modifies nothing in both the BGP code and the routing message format. We utilize the existing infrastructures as much as possible. This helps to make our method more deployable than the previous approaches. Second, our method can also work in concert with other approaches though it can be performed independently. It can support route filtering or other topology based security mechanisms [35, 16, 21]. Finally, our method can be incrementally deployed in the Internet.

1.3 Outline

The rest of the paper is organized as follows. We summarize the related work in Section 2 and BGP security threats are addressed in Section 3. We then present the definition of a RAVS scheme and describe the framework of the RAVS scheme with respect to BGP in Section 4. We construct a RAVS scheme and prove its securities in Section 5. We explain how the approach based on our RAVS scheme detects and prevents BGP security threats in Section 6. Section 7 shows the experimental results. Finally, we discuss practical deployment issues in Section 8 and conclude in Section 9.

2. RELATED WORK

2.1 BGP Security Solution

BGP security solutions can be classified as cryptographic or non-cryptographic approaches.

Cryptographic approach. Secure BGP (S-BGP) [19, 20] uses Public Key Infrastructure (PKI) to validate BGP UPDATE messages. Though this protocol addresses most of the security problems of BGP, it is difficult to use S-BGP in the Internet because of routing message overheads, expensive computation costs, and deployment problem.

Secure Origin BGP (soBGP) [35, 25] also uses PKI for authenticating ASes and authorizing address ownership. It maintains a database of network topology based on received policy information. The routers detects invalid routes based on this database.

Recently, Wan et al. [33] propose Pretty Secure BGP (psBGP). psBGP makes use of a centralized trust method for AS number authentication and a decentralized trust method for verifying the propriety of IP prefix ownership.

Instead of heavy digital signatures, a light mechanism which relies on symmetric cryptographic functions is designed by Hu et al. [17]. This mechanism proposes to use hash chains to prevent an attacker from modifying and truncating the AS_PATH in the UPDATE message.

Subramanian et al. [32] suggest two mechanisms Listen and Whisper. The Whisper protocol uses cryptographic functions along with routing redundancy to detect fake route advertisements in the routing control plane. The Listen protocol detects invalid routes in the data plane.

Different from these approaches, we build a secure database to validate BGP UPDATE messages. We use identity-based cryptosystem instead of PKI.

Non-cryptographic approach. We will discuss three major non-cryptographic approaches.

First, the Internet Routing Registry (IRR) [18, 3, 2] is a centralized database of routing policy information. ASes register their policies and topological information into the database. ASes also query this database for validating BGP UPDATE messages.

Second, BGP route filtering is the most widely deployed and effective technique for protecting BGP in the current Internet [11, 26]. It is mainly used to enforce business relationships between ASes. Routers can use access control lists to filter out prefixes or ASes when it sends or receives UPDATE messages.

Third, the Interdomain Route Validation (IRV) is proposed by Goodell et al. [16]. The IRV is independent of BGP. Every AS contains an IRV server. IRV server sends queries to IRV servers in other ASes for validating received routing information. Each message can be validated by querying directly to the AS from which it originates. Access control can be used with the IRV to protect sensitive policies from untrusted parties.

Our approach controls the access to the routing information database with our own authorization mechanism. We use the concept of permission for searching the database.

2.2 Searching on Encrypted Data

Song et al. [30] propose an efficient searching scheme on encrypted data in a symmetric key setting. This scheme uses a sequence of pseudorandom values as a key. The same key is used for encrypting data and searching on that encrypted data. Their solution is very efficient in terms of search time and communication overheads.

Goh [14] combines an encrypted index with pseudorandom functions and Bloom filters. They define an index scheme and formulate its security model (IND-CKA). They construct an IND-CKA secure index scheme. Their method also uses a symmetric key. The search time is constant per document. Chang and Mitzenmacher [12] also construct two secure index schemes using pre-built dictionaries and prove stronger security (IND2-CKA).

Boneh et al. [6] propose a searchable encryption scheme which uses public key encryption. They devise a scheme based on the identity-based encryption scheme in [7]. They define a public key encryption with keyword search scheme, and provide a scheme which is semantically secure against an adaptive chosen keyword attack.

Bellovin and Cheswick [5] suggest a search scheme based on Bloom filters and Pohlig-Hellman encryption. They use a semi-trusted third party for transforming one party's search queries to a form for the other party's database. As a result, neither the third party nor the database owner can learn the content of query.

Waters et al. [34] build two schemes for searching on encrypted audit logs. The symmetric key scheme is partly based on the above Goh's technique; the asymmetric key scheme uses identity-based encryption scheme in [7].

Finally, Golle et al. [15] and Park et al. [27] suggest extended schemes which allow conjunctive keyword search.

Our method combines a searchable registry with identity-based cryptosystem. It can provide authorized and verifiable search simultaneously. To the best of our knowledge, there is no previous work that provides the verifiable search.

3. BGP SECURITY THREAT

In this section, we describe the BGP threats which we address in this paper. We focus on the threats in BGP control plane. Figure 2 shows an example of abstract AS

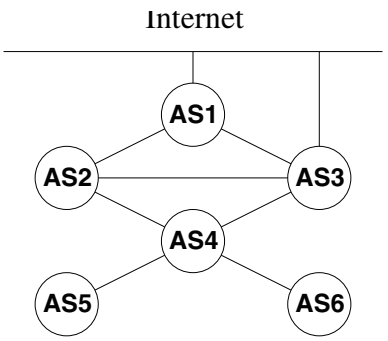

Figure 2: An example AS topology. AS4 is a customer of AS2 and AS3 and the provider of AS5 and AS6. AS4 uses the link AS3-AS4 as primary link and the link AS2-AS4 as backup link. AS2 and AS3 uses the link AS2-AS3 as backup link.

topology in [26]. AS4 is a multihomed AS and is a customer of AS2 and AS3. It is also the provider of AS5 and AS6.

Modification attack. The *modification attack* is that a malicious AS modifies the path attributes in the BGP UPDATE message. The most important path attribute in the UPDATE message is AS_PATH. BGP determines the routing path based on the AS_PATH information. The length of the AS path is the second criterion in BGP path selection process. If an attacker modifies a valid AS_PATH to a shorter but invalid one and advertises the UPDATE message which includes this invalid AS_PATH to neighboring ASes, then they prefer the fake path and update their routing table entries. For example, suppose that AS3 wants to redirect traffic destined to AS5 in Figure 2. It could advertise that it has a direct connection to AS5. Traffic from AS1 and other parts of the Internet would pass through AS3 due to the shorter AS_PATH announced by AS3.

The modification attack may affect interdomain traffic engineering. A multihomed AS (AS4) can send UPDATE messages with a padded AS_PATH to one of its providers (AS2 and AS3). For example, suppose that AS4 uses the link AS3-AS4 as primary connection to the global Internet, and the link AS2-AS4 as backup. We also assume that the link AS2-AS3 is a backup link, hence AS3 does not advertise UPDATE messages to AS2. AS4 sends UPDATE messages with the padded AS_PATHs [AS4 AS4 AS5] and [AS4 AS4 AS6] to AS2. On the other hand, it sends UPDATE messages [AS4 AS5] and [AS4 AS6] to AS3. Other ASes would prefer the shorter path with the link AS3-AS4. An attacker can use such padding technique to redirect traffic. For instance, AS1 can dump traffic to AS2 and advertise UPDATE messages with the AS_PATH [AS1 AS2 AS4 AS5] instead of [AS1 AS2 AS4 AS4 AS5], and [AS1 AS1 AS3 AS4 AS5] instead of [AS1 AS3 AS4 AS5]. This makes traffic from other part of the Internet for AS5 take the backup link AS2-AS4.

Misconfiguration. BGP configuration errors can disrupt the Internet connectivity. Mahajan et al. [23] study various kinds of BGP *misconfigurations*. Two forms of misconfigurations are identified: origin and export misconfiguration. The former is that an AS injects specific prefixes into the global BGP tables, or announces other ASes' address prefixes. The latter is that a router incorrectly exports a route it should filter. For instance, suppose that AS4 uses the link AS2-AS4 as backup link only and the link AS2-AS3 is a backup link. AS4 advertises a route [AS4 AS4 AS4] to AS2 and [AS4] to AS3. Traffic destined to AS4 from the Internet goes through AS3-AS4 link. AS3 filters out the route [AS3 AS4] to AS2 so that the link AS2-AS3 cannot be used in normal case. Assume that AS3 accidentally exports that route to AS2 by misconfiguration. From AS2's point of view, [AS3 AS4] is shorter than [AS4 AS4 AS4], hence it chooses the route [AS3 AS4] instead of its direct link AS2-AS4.

Exposing attack. The *exposing attack* is an attack to retrieve the sensitive information which ASes do not want to reveal to the others such as peering relationships, routing policies and routes. This attack becomes severe if there is a centralized database server like the IRR which records such information. We categorize adversaries into two groups: passive and active exposing adversary. The former only eavesdrops the query and response packets between the database server and ASes. The latter can actively attack the database server to access the database.

Contamination attack. The *contamination attack* means forging the information in database server like the IRR. The effects of the contamination attack become severe if many ASes depend on the IRR for route validation. The existing IRR provides very weak security mechanisms. It neither guarantees the integrity of the database contents nor provides authorization of changes to the database. Suppose that an attacker notices that the target AS uses the IRR information to validate incoming UPDATE messages. If the attacker wants to mount the modification attack to the target AS, it would modify the information in the IRR to make the target AS accept its fake UPDATE message. We classify these contamination attackers into two types.

- Local adversary: A compromised AS can register incorrect BGP routing information of its neighborhood to the IRR.
- Global adversary: An attacker can disguise itself as other legitimate ASes and add incorrect routing information or remove valid entries from the IRR.

We address these BGP threats using secure routing information database. We present our method in the following sections. In Section 6, we will show how our method can address these threats.

4. RAVS DEFINITION AND FUNCTION

In this section, we first present notations which are used throughout the paper and define a RAVS scheme. Furthermore, we demonstrate the general framework of the RAVS scheme with respect to BGP.

4.1 Notations and Definition

Notations. Throughout the paper, we use ID to stand for the identity of each entity, and d_{ID} as the corresponding private key. We use W to denote the information or the search keyword we want to conceal. Let $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ or $\mathrm{RAVS}(d_{\mathsf{ID}}, \mathsf{W})$ denote the registry of an entity with an identity ID for a keyword W. Let $\mathcal{P}_{\mathsf{W}}$ denote the search permission for a keyword

W. We use $x \xleftarrow{R} \mathbb{Z}_q^*$ to denote a random variable x chosen uniformly at random from the set $\mathbb{Z}_q^*$. We use $\|$ to denote string concatenation.

Definition 1. An *identity-based Registry with Authorized and Verifiable Search* (RAVS) scheme is specified by the following polynomial-time randomized algorithms:

SETUP: The Private Key Generator (PKG) takes a security parameter k and returns system parameters and the master keys. The system parameters are publicly known, while the master keys are known only to the PKG and the Search Permission Generator (SPG).

EXTRACT: The PKG takes master keys and an identity ID and returns the corresponding private key d_{ID}.

RAVS: This algorithm is run by the entity which wants to construct its registry. The inputs are constructor's private key d_{ID} and a keyword W. The algorithm builds the registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ which includes the identity ID and the keyword W as the concealed information.

This algorithm should guarantee that no information can be leaked from the registry. The registry should be searchable, only if the corresponding search permission $\mathcal{P}_{\mathsf{W}}$ is available. Furthermore, it should be verifiable by the constructor's identity ID.

PERMISSION: The SPG takes master keys and a keyword W and returns a search permission $\mathcal{P}_{\mathsf{W}}$.

The search permission should not leak any information about the keyword W.

TESTREGISTRY: This algorithm takes a registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$, an identity ID^*, and a search permission $\mathcal{P}_{\mathsf{W}^*}$ as inputs. It tests the registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ for the followings: (1) whether the registry conceals the keyword W^* of the search permission $\mathcal{P}_{\mathsf{W}^*}$, and (2) whether the registry is constructed by the entity with the identity ID^*. If both conditions are satisfied, then the algorithm outputs 1; otherwise outputs 0.

This algorithm should be performed without revealing any information of ID and W from the registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ and W^* from the search permission $\mathcal{P}_{\mathsf{W}^*}$.

4.2 Framework and Components

We demonstrate the general framework of RAVS scheme with respect to BGP in Figure 3. We first explain each component's role in the framework.

For the RAVS scheme to be applied to BGP, we require a centralized database server for maintaining registries and testing the registries with search queries. This server does not need to be fully trusted. It does not need to have any function of authenticating ASes. It is enough to guarantee the correct computation of searching tests. The reason why we require the server with such security level is that we already have similar infrastructure on the Internet, the Internet Routing Registry (IRR). We utilize the existing IRR with adding the function TESTREGISTRY. We refer to this modified server as the *modified Internet Routing Registry (mIRR)* in the rest part of the paper.

The *Private key Generator (PKG)* is the server used in the identity-based cryptosystem. Every AS has its own AS number as its identity on the Internet. Hence, we can use AS number as the public key of that AS in our identity-based

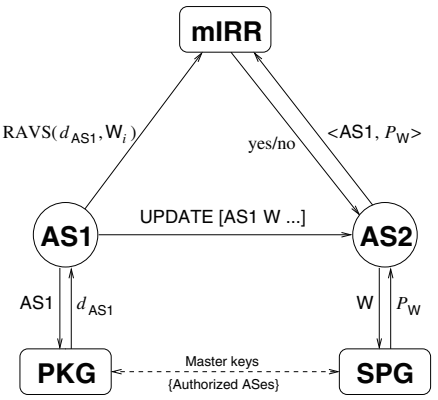

Figure 3: Framework of the RAVS scheme with respect to BGP. Assume that AS1 advertises an UPDATE message [AS1 W ...] to AS2. $\mathsf{AS1}$ is AS1's identity and d_{AS1} is AS1's private key. W is a search keyword and $\mathcal{P}_{\mathsf{W}}$ is the search permission for W. AS1 submits $\mathrm{RAVS}(d_{\mathsf{AS1}}, \mathsf{W}_i)$ for each routing information W_i. AS2 wants to check whether AS1 has a routing information W and sends a search query composed of $\mathsf{AS1}$ and $\mathcal{P}_{\mathsf{W}}$. The mIRR tests each $\mathrm{RAVS}(d_{\mathsf{AS1}}, \mathsf{W}_i)$ with $\mathsf{AS1}$ and $\mathcal{P}_{\mathsf{W}}$. If there is a match ($\mathsf{W}_i = \mathsf{W}$), the mIRR returns 'yes' to AS2; otherwise returns 'no'.

scheme. As usual, an AS with identity ID authenticates itself to the PKG. The PKG then generates the corresponding private key $d_{\mathsf{ID}} = \text{EXTRACT}(\mathsf{ID})$ and returns it to the AS.

The *Search Permission Generator (SPG)* produces search permissions only to the legitimate ASes. The AS, which wants to query to the mIRR, must obtain the search permission $\mathcal{P}_{\mathsf{W}}$ corresponding to that search keyword W. The AS first authenticates itself and sends W to the SPG. The SPG constructs the search permission $\mathcal{P}_{\mathsf{W}} = \text{PERMISSION}(\mathsf{W})$ and returns it to the AS. Therefore, the SPG can prevent illegal access to the mIRR.

In our scheme, the PKG and the SPG share the master keys, hence they have the same security level. When an AS obtains its private key from the PKG to construct its registries, this AS is automatically put into the set of authorized ASes by the PKG. The PKG shares the set of authorized ASes with the SPG. The SPG generates search permissions based on this set. In other words, only the ASes in this set are authorized to obtain search permissions from the SPG.

Registry submission. Suppose an AS with identity ID has n routing information $\mathsf{W}_1, \mathsf{W}_2, \ldots, \mathsf{W}_n$. For example, the routing information could be peering relationships, traffic engineering policies, and other things. The AS first obtains its private key d_{ID} from the PKG. For each routing information W_i ($1 \le i \le n$), the AS constructs $\mathcal{R}_{\mathsf{ID},\mathsf{W}_i} = \mathrm{RAVS}(d_{\mathsf{ID}}, \mathsf{W}_i)$ using its private key d_{ID}. It then submits n registries $\mathcal{R}_{\mathsf{ID},\mathsf{W}_1}, \mathcal{R}_{\mathsf{ID},\mathsf{W}_2}, \ldots, \mathcal{R}_{\mathsf{ID},\mathsf{W}_n}$ to the mIRR. Note that the mIRR cannot learn the content W_i from the registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}_i}$. For example, suppose that AS1 is going to submit its routing information to the mIRR in Figure 3. AS1 first obtains its own private key d_{AS1} from the PKG. AS1 then constructs the following registries for its n routing

information:

$$\mathrm{RAVS}(d_{\mathsf{AS1}}, \mathsf{W}_1), \mathrm{RAVS}(d_{\mathsf{AS1}}, \mathsf{W}_2), \ldots, \mathrm{RAVS}(d_{\mathsf{AS1}}, \mathsf{W}_n)$$

where d_{AS1} is AS1's private key and W_i $(1 \leq i \leq n)$ is AS1's routing information. W_i can be either a single AS number or concatenation of several AS numbers. For more general applications, we use keywords in this way. We explain the practical use of keywords in Section 6.

If an AS wants to update or remove its registry already stored in the mIRR, it should submit the same registry as the one in the mIRR. Hence, an AS cannot update or remove other ASes' registries in the mIRR.

Search query and response. Suppose an AS wants to investigate whether a target AS with identity ID has W as its routing information. The investigator AS first obtains the search permission $\mathcal{P}_{\mathsf{W}}$ for a keyword W from the SPG. The investigator AS sends a query to the mIRR with the target AS's identity ID and the search permission $\mathcal{P}_{\mathsf{W}}$. The mIRR tests each registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}_i}$ with ID and $\mathcal{P}_{\mathsf{W}}$ using the TestRegistry algorithm. Note that the mIRR could learn nothing from both the registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}_i}$'s and the permission $\mathcal{P}_{\mathsf{W}}$ during performing TestRegistry. Finally, the mIRR finishes searching all registries of the AS ID. If there is a match, it responses with the answer 'yes'. If there is no match throughout tests, it responses with the answer 'no'. For instance, assume that AS2 receives a BGP UPDATE message from AS1 in Figure 3. Moreover, let us assume that AS1 has already submitted its registries to the mIRR. If the received UPDATE message includes a new route AS1-W, AS2 wants to check whether AS1 has W as its routing information. AS2 first obtains the search permission $\mathcal{P}_{\mathsf{W}}$ from the SPG. The AS2's search query consists of the identity $\mathsf{AS1}$ and the search permission $\mathcal{P}_{\mathsf{W}}$. The mIRR tests each $\mathcal{R}_{\mathsf{AS1},\mathsf{W}_i}$ with $\mathsf{AS1}$ and $\mathcal{P}_{\mathsf{W}}$. If there is a match $(\mathsf{W}_i = \mathsf{W})$, the mIRR returns 'yes' to AS2; otherwise it returns 'no'.

Registry provider verification. We verify the registry provider when TestRegistry is invoked for the registries belonged to that provider. Suppose that an attacker constructs a fake registry $\mathcal{R}_{\mathsf{AS3},\mathsf{W}}$ and submits it to the mIRR as $\mathsf{AS1}$'s registry. Since the mIRR has no way to learn the content of registry, it cannot check the registry's validity directly. However, the fake registry should never produce a match result returned by TestRegistry since $\mathsf{AS3} \neq \mathsf{AS1}$. Therefore, we can verify the invalid registry provider at the search time.

5. RAVS SCHEME

We present a RAVS scheme in this section. We then define the securities for RAVS scheme and prove that our RAVS scheme is secure in the random oracle model under cryptographic assumptions. Finally, we introduce an additional algorithm to our scheme to hide AS's identity in the search query. We start by explaining the admissible bilinear map.

Admissible bilinear map. Let k be a security parameter and q be a k-bit prime number. Let $\mathbb{G}_1$, $\mathbb{G}_2$ be two groups of prime order q. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be an admissible bilinear map with the following properties [7]:

1. *Bilinear*: $\forall\, P, Q \in \mathbb{G}_1, \forall\, a, b \in \mathbb{Z}_q^*$, we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
2. *Nondegenerate*: If P is a generator of $\mathbb{G}_1$, then $\hat{e}(P, P)$ is a generator of $\mathbb{G}_2$.
3. *Computable*: $\forall\, P, Q \in \mathbb{G}_1$, there is a polynomial-time algorithm to compute $\hat{e}(P, Q) \in \mathbb{G}_2$.

As mentioned in [7], the Weil or the tate pairing satisfies the above properties and can be used to obtain such nondegenerate admissible maps. We refer the readers to Boneh and Franklin [7] for details.

5.1 Construction

We construct a RAVS scheme which is similar to a signcryption scheme in [9]. However, its construction is much simpler since it does not need to encrypt or decrypt a message.

Setup(k): Given a security parameter k, choose groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q > 2^k$. Choose an admissible bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Let P be a generator of the group $\mathbb{G}_1$. Choose $s, t \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and compute $S = sP \in \mathbb{G}_1$ and $T = tP \in \mathbb{G}_1$. We need three cryptographic hash functions $H_1, H_2 : \{0,1\}^* \to \mathbb{G}_1$ and $H_3 : \mathbb{G}_2 \to \mathbb{Z}_q^*$.

- The system parameters are $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_1, H_2, H_3, P, S, T\rangle$.
- The master keys are $s, t \in \mathbb{Z}_q^*$.

Extract(ID): Given an identity $\mathsf{ID} \in \{0,1\}^*$, compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}_1$ and set the private key $d_{\mathsf{ID}} = sQ_{\mathsf{ID}} \in \mathbb{G}_1$ where s is the master key.

RAVS$(d_{\mathsf{ID}}, \mathsf{W})$: Given a private key $d_{\mathsf{ID}} \in \mathbb{G}_1$ and a keyword $\mathsf{W} \in \{0,1\}^*$,

1. Choose $r \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and compute $U_1 = rT \in \mathbb{G}_1$ and $U_2 = rP \in \mathbb{G}_1$.
2. Compute $\omega = \hat{e}(rS, H_2(\mathsf{W})) \in \mathbb{G}_2$.
3. Compute $h = H_3(\omega) \in \mathbb{Z}_q^*$ and $V = hd_{\mathsf{ID}} \in \mathbb{G}_1$.

The registry of ID for a keyword W is $\mathcal{R}_{\mathsf{ID},\mathsf{W}} = \langle U_1, U_2, V\rangle \in \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1$.

Permission(W): Given a keyword $\mathsf{W} \in \{0,1\}^*$,

1. Choose $m \overset{R}{\leftarrow} \mathbb{Z}_q^*$ such that $\gcd(t+m, q) = 1$ where t is one of the master keys.
2. Compute $z = s(t+m)^{-1} \pmod q \in \mathbb{Z}_q^*$ where s, t are the master keys.
3. Compute $M = zH_2(\mathsf{W}) \in \mathbb{G}_1$.

The permission for a keyword W is $\mathcal{P}_{\mathsf{W}} = \langle M, m\rangle \in \mathbb{G}_1 \times \mathbb{Z}_q^*$.

TestRegistry$(\mathcal{R}_{\mathsf{ID},\mathsf{W}}, \mathsf{ID}, \mathcal{P}_{\mathsf{W}})$: To test a registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}} = \langle U_1, U_2, V\rangle \in \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1$ with an identity ID and a permission $\mathcal{P}_{\mathsf{W}} = \langle M, m\rangle \in \mathbb{G}_1 \times \mathbb{Z}_q^*$,

1. Compute $\omega = \hat{e}(U_1 + mU_2, M) \in \mathbb{G}_2$, $h = H_3(\omega) \in \mathbb{Z}_q^*$, and $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}_1$.
2. Output $\begin{cases} 1 & \text{if } \hat{e}(P, V) = \hat{e}(S, hQ_{\mathsf{ID}}), \\ 0 & \text{otherwise.} \end{cases}$

The above RAVS scheme is consistent since

$$\begin{aligned}\hat{e}(U_1 + mU_2, M) &= \hat{e}(rT + m(rP), zH_2(\mathsf{W})) \\ &= \hat{e}(r(t+m)P, s(t+m)^{-1}H_2(\mathsf{W})) \\ &= \hat{e}(rP, sH_2(\mathsf{W}))^{(t+m)\cdot(t+m)^{-1}} \\ &= \hat{e}(rS, H_2(\mathsf{W})),\end{aligned}$$

$$\hat{e}(S, hQ_{\mathsf{ID}}) = \hat{e}(sP, hQ_{\mathsf{ID}}) = \hat{e}(P, hd_{\mathsf{ID}}) = \hat{e}(P, V).$$

According to the definitions in [1], the consistency here does not mean perfect consistency but computational consistency.

5.2 Securities

Formally, we define two securities for RAVS scheme: semantic security against adaptive chosen keyword and identity attacks (IND-ID-CKA) and existential unforgeability against adaptive chosen keyword and identity attacks (EUF-ID-CKA).

Semantic security. We want to show that RAVS($d_{\mathsf{ID}}, \mathsf{W}$) does not reveal any information about a keyword W without permission $\mathcal{P}_{\mathsf{W}}$. An active attacker can obtain private keys d_{ID}, search permissions $\mathcal{P}_{\mathsf{W}}$, and registries $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ for any identity ID and any keyword W of his choice. The attacker should not be able to distinguish a registry of ID^* for a keyword W_0 from a registry of ID^* for a keyword W_1 under the restriction that he has not obtained the permissions for W_0 and W_1.

We define semantic security against adaptive chosen keyword and identity attacks using the following game between a challenger $\mathcal{C}$ and an attacker $\mathcal{A}$.

Setup The challenger $\mathcal{C}$ runs the algorithm SETUP for a given security parameter k, and obtains the system parameters and sends them to the attacker $\mathcal{A}$. It keeps the master keys for itself.

Phase 1 $\mathcal{A}$ adaptively issues a series of queries, each of which is one of the followings:

- EXTRACT query (ID): $\mathcal{C}$ runs the algorithm EXTRACT to generate the private key d_{ID} corresponding to ID. It returns d_{ID} to $\mathcal{A}$.
- PERMISSION query (W): $\mathcal{C}$ runs the algorithm PERMISSION to generate the permission $\mathcal{P}_{\mathsf{W}}$ for a keyword W. It returns $\mathcal{P}_{\mathsf{W}}$ to $\mathcal{A}$.
- RAVS query (ID, W): $\mathcal{A}$ queries the registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ for any identity ID and for any keyword W of his choice. $\mathcal{C}$ first runs the algorithm EXTRACT to generate the private key d_{ID} corresponding to ID and runs the algorithm RAVS to generate the registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$. It returns $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ to $\mathcal{A}$.

Challenge $\mathcal{A}$ sends $\mathcal{C}$ two keywords $\mathsf{W}_0, \mathsf{W}_1$ and an identity ID^* on which it wishes to be challenged. The only restriction is that $\mathcal{A}$ has made no permission query on W_0 or W_1 in Phase 1.

$\mathcal{C}$ runs the algorithm EXTRACT to generate the private key d_{ID^*} corresponding to ID^*. $\mathcal{C}$ picks a random bit $b \in \{0,1\}$ and gives the challenge RAVS($d_{\mathsf{ID}^*}, \mathsf{W}_b$) to $\mathcal{A}$.

Phase 2 $\mathcal{A}$ can continue to query as in Phase 1. The only restriction is that $\mathsf{W} \neq \mathsf{W}_0, \mathsf{W}_1$ in PERMISSION(W) query. $\mathcal{C}$ responds as in Phase 1.

Response $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ and wins the game if $b = b'$.

We define an attacker $\mathcal{A}$'s advantage in breaking the RAVS scheme as the probability that $\mathcal{A}$ wins in the above game, taken over the coin tosses of $\mathcal{C}$ and $\mathcal{A}$,

$$\mathbf{Adv}^{\text{IND-ID-CKA}}_{\text{RAVS},\mathcal{A}}(k) = \left|\Pr[b = b'] - \frac{1}{2}\right|.$$

Definition 2. A RAVS scheme is said to be *semantically secure against adaptive chosen keyword and identity attacks*, or IND-ID-CKA *secure*, if for any probabilistic polynomial-time attacker $\mathcal{A}$ and any polynomial function p,

$$\mathbf{Adv}^{\text{IND-ID-CKA}}_{\text{RAVS},\mathcal{A}}(k) \leq \frac{1}{p(k)},$$

for almost all $k > 0$.

Existential unforgeability. Next, we want to show that RAVS($d_{\mathsf{ID}}, \mathsf{W}$) cannot be constructed without private key d_{ID}. An active forger can obtain private keys d_{ID}, search permissions $\mathcal{P}_{\mathsf{W}}$, and registries $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ for any identity ID and any keyword W of his choice. The forger should not be able to construct a registry for which TESTREGISTRY returns '1' with an identity ID^* and a permission for a keyword W^* of his choice under the restriction that he has not obtained the private key for ID^* and the registry $\mathcal{R}_{\mathsf{ID}^*,\mathsf{W}^*}$.

We define existential unforgeability against adaptive chosen keyword and identity attacks using the following game between a challenger $\mathcal{C}$ and a forger $\mathcal{F}$.

Setup The challenger $\mathcal{C}$ runs the algorithm SETUP for a given security parameter k, and obtains the system parameters and sends them to the forger $\mathcal{F}$. It keeps the master keys for itself.

Queries $\mathcal{F}$ adaptively issues a series of queries, each of which is one of the followings:

- EXTRACT query (ID): $\mathcal{C}$ runs the algorithm EXTRACT to generate the private key d_{ID} corresponding to ID. It returns d_{ID} to $\mathcal{F}$.
- PERMISSION query (W): $\mathcal{C}$ runs the algorithm PERMISSION to generate the permission $\mathcal{P}_{\mathsf{W}}$ for a keyword W. It returns $\mathcal{P}_{\mathsf{W}}$ to $\mathcal{F}$.
- RAVS query (ID, W): $\mathcal{F}$ queries the registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ for any identity ID and for any keyword W of his choice. $\mathcal{C}$ first runs the algorithm EXTRACT to generate the private key d_{ID} corresponding to ID and runs the algorithm RAVS to generate the registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$. It returns $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ to $\mathcal{F}$.

Output After a polynomial number of queries, $\mathcal{F}$ outputs a tuple $\langle \mathsf{ID}^*, \mathsf{W}^*, \mathcal{R}^* \rangle$ and wins the game if

1. ID^* was never asked to the EXTRACT oracle,
2. $\mathcal{R}^*$ was never returned by the RAVS oracle on the input $(\mathsf{ID}^*, \mathsf{W}^*)$, and
3. TESTREGISTRY$(\mathcal{R}^*, \mathsf{ID}^*, \mathcal{P}_{\mathsf{W}^*}) = 1$, where $\mathcal{P}_{\mathsf{W}^*}$ is a permission for W^* returned by the PERMISSION oracle.

We define a forger $\mathcal{F}$'s advantage in breaking the RAVS scheme as the probability that $\mathcal{F}$ wins in the above game, taken over the coin tosses of $\mathcal{C}$ and $\mathcal{F}$,

$$\mathbf{Adv}^{\mathsf{EUF\text{-}ID\text{-}CKA}}_{\mathrm{RAVS},\mathcal{F}}(k) = \Pr[\mathcal{F} \text{ wins}].$$

Definition 3. A RAVS scheme is said to be *existentially unforgeable against adaptive chosen keyword and identity attacks*, or EUF-ID-CKA *secure*, if for any probabilistic polynomial-time forger $\mathcal{F}$ and any polynomial function p,

$$\mathbf{Adv}^{\mathsf{EUF\text{-}ID\text{-}CKA}}_{\mathrm{RAVS},\mathcal{F}}(k) \leq \frac{1}{p(k)},$$

for almost all $k > 0$.

5.2.1 Assumptions

The securities of our RAVS scheme are based on the difficulties of the Elliptic Curve Discrete Logarithm Problem (ECDLP), the computational Diffie-Hellman (CDH) problem, and the bilinear Diffie-Hellman (BDH) problem [7]. Let $\mathbb{G}_1$, $\mathbb{G}_2$ be two groups of prime order q and let P be a generator of $\mathbb{G}_1$.

Elliptic Curve Discrete Logarithm Problem (ECDLP) is defined as follows: given $\langle P, aP\rangle$, find an $a \in \mathbb{Z}^*_q$. We choose the security parameter so that the discrete logarithm problem is hard in $\mathbb{G}_1$ [7].

Computational Diffie-Hellman (CDH) problem is defined as follows: given $\langle P, aP, bP\rangle$ for some $a, b \in \mathbb{Z}^*_q$, compute $abP \in \mathbb{G}_1$.

CDH assumption If the probability to solve the CDH problem is negligible in k for all probabilistically polynomial-time algorithms, we say CDH is intractable.

Bilinear Diffie-Hellman (BDH) problem is defined as follows: given $\langle P, aP, bP, cP\rangle$ for some $a, b, c \in \mathbb{Z}^*_q$, compute $\hat{e}(P, P)^{abc} \in \mathbb{G}_2$.

BDH assumption If the probability to solve the BDH problem is negligible in k for all probabilistically polynomial-time algorithms, we say BDH is intractable.

5.2.2 Security proofs

We prove the securities of our scheme in the random oracle model [4] under the above intractability assumptions. Our proofs use the similar arguments in the proof of [6] for IND-ID-CKA security and the proof of [22] for EUF-ID-CKA security. Both of proofs are based on the Coron's analysis [13]. The details of the proofs are in the full version of the paper.

THEOREM 1. *If BDH is intractable, then our RAVS scheme is semantically secure under adaptive chosen keyword and identity attacks in the random oracle model.*

THEOREM 2. *If CDH is intractable, then our RAVS scheme is existentially unforgeable under adaptive chosen keyword and identity attacks in the random oracle model.*

5.3 Hidden Identity Query

In our scheme, the search query includes the registry provider's identity ID in plaintext form. We can also hide this identity during querying. However, the mIRR also could not know which registries it tests with such a hidden identity query. The only solution is the brute force searching throughout the database. Therefore, the search time increases dramatically.

We add a new algorithm HIDDENIDENTITY to our scheme. We modify TESTREGISTRY slightly to TESTREGISTRY2 for processing such queries.

HIDDENIDENTITY(ID): Choose $l \stackrel{R}{\leftarrow} \mathbb{Z}^*_q$ and compute $X = lP \in \mathbb{G}_1$. Compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}_1$ and $Y = lQ_{\mathsf{ID}}$. The hidden identity is $\mathcal{H}_{\mathsf{ID}} = \langle X, Y\rangle \in \mathbb{G}_1 \times \mathbb{G}_1$.

TESTREGISTRY2($\mathcal{R}_{\mathsf{ID},\mathsf{W}}, \mathcal{H}_{\mathsf{ID}}, \mathcal{P}_{\mathsf{W}}$): To test a registry $\mathcal{R}_{\mathsf{ID},\mathsf{W}} = \langle U_1, U_2, V\rangle \in \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1$ with a hidden identity $\mathcal{H}_{\mathsf{ID}} = \langle X, Y\rangle \in \mathbb{G}_1 \times \mathbb{G}_1$ and a permission $\mathcal{P}_{\mathsf{W}} = \langle M, m\rangle \in \mathbb{G}_1 \times \mathbb{Z}^*_q$,

1. Compute $\omega = \hat{e}(U_1 + mU_2, M) \in \mathbb{G}_2$ and $h = H_3(\omega) \in \mathbb{Z}^*_q$.
2. Output $\begin{cases} 1 & \text{if } \hat{e}(X, V) = \hat{e}(S, hY), \\ 0 & \text{otherwise.} \end{cases}$

The modified RAVS scheme with hidden identity query is consistent since

$$\hat{e}(X, V) = \hat{e}(lP, hd_{\mathsf{ID}}) = \hat{e}(sP, hlQ_{\mathsf{ID}}) = \hat{e}(S, hY).$$

6. RAVS APPLIED TO BGP

We present the applications of the RAVS scheme in BGP. We focus on the adversaries defined in Section 3. We describe how our RAVS scheme addresses each of these adversaries.

Modification attack. We consider modification attacks in Figure 2.

AS_PATH modification In the previous example, these attacks are easily identified by querying the mIRR to see whether there is a route or not. AS3 announces that it has a direct connection to AS5. AS1 obtains a search permission $\mathcal{P}_{\mathsf{AS3}}$ for a keyword AS3 from the SPG. It queries the mIRR with identity AS5 and permission $\mathcal{P}_{\mathsf{AS3}}$. The mIRR performs TESTREGISTRY($\mathcal{R}_{\mathsf{AS5},\mathsf{AS}i}, \mathsf{AS5}, \mathcal{P}_{\mathsf{AS3}}$) for each AS5's registries (ASi is a neighboring AS of AS5). AS5 has no neighbor as AS3, hence there is no such registry in the mIRR. All the test results must be '0'.

Padding modification AS4 submits the RAVS(d_{AS4}, AS4$\|$ *AS*2) and the RAVS(d_{AS4}, AS3) to the mIRR. AS4$\|$AS2 and AS3 are keywords in each registry. Suppose that an AS in the outer Internet receives UPDATE messages from AS1. One UPDATE includes the AS_PATH [AS1 AS2 AS4 AS5], and the other includes [AS1 AS1 AS3 AS4 AS5]. The AS which wants to investigate the validness of received UPDATE messages first obtains search permissions $\mathcal{P}_{\mathsf{AS2}}$ and $\mathcal{P}_{\mathsf{AS3}}$ from the SPG. The investigator AS queries the mIRR with identity AS4 and permissions $\mathcal{P}_{\mathsf{AS2}}$ and $\mathcal{P}_{\mathsf{AS3}}$. The mIRR tests and returns results: '0' for $\mathcal{P}_{\mathsf{AS2}}$ and '1' for $\mathcal{P}_{\mathsf{AS3}}$. As a result, the investigator AS choose the route [AS1 AS1 AS3 AS4 AS5] though its AS_PATH length is longer than the other. We successfully prevent AS1's malicious behavior. If AS1 propagates valid AS_PATHs, ASes in the other part of the Internet could receive UPDATE messages including [AS1 AS2 AS4 AS4 AS5] and [AS1 AS3 AS4 AS5]. In this case, both test results are '1's and the investigator AS choose [AS1 AS3 AS4 AS5] with shorter length.

Misconfiguration. Our method can mitigate some export misconfiguration. In Figure 2, AS3 submits the RAVS(d_{AS3}, $\mathsf{AS3}\|\mathsf{AS4}$) and AS4 submits the RAVS($d_{\mathsf{AS4}}, \mathsf{AS3}$) to the mIRR. AS3 advertises an UPDATE message including a route [AS3 AS3 AS4] to the outer Internet, but not to AS2. ASes in the global Internet query the mIRR with identity $\mathsf{AS3}$ (respectively $\mathsf{AS4}$) and permission $\mathcal{P}_{\mathsf{AS3}\|\mathsf{AS4}}$ (respectively $\mathcal{P}_{\mathsf{AS3}}$). Both tests return the result '1' as required. Since AS2 advertises a route [AS2 AS4 AS4 AS4], traffic destined to AS4 still passes through AS3. Assume that AS3 accidentally exports the route [AS3 AS4] or [AS3 AS3 AS4] to AS2. The first route is rejected since that information is not in the mIRR. From AS2's point of view, the length of the second route and that of the existing route are same. Hence, AS2 does not change the route to AS4 just based on the path length.

Exposing attack. A passive exposing adversary may try to retrieve the AS topology by eavesdropping the queries and the responses between legitimate ASes and the mIRR. Suppose that a legitimate AS issues a query $\langle \mathsf{ID}, \mathcal{P}_{\mathsf{W}} \rangle$ to the mIRR. The adversary listens this query and corresponding result from the mIRR. Though the adversary could know ID's registries are searched and what is the result of the tests, it has no way to learn the content W from the permission $\mathcal{P}_{\mathsf{W}}$. As a result, the adversary learns nothing about the Internet routing information from query and response.

An active exposing adversary can hack the mIRR server and obtain the database of the routing information. Suppose that $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ is one of the stolen registries. It is impossible to retrieve the information W from the $\mathcal{R}_{\mathsf{ID},\mathsf{W}}$ without the search permission $\mathcal{P}_{\mathsf{W}}$ by the semantic security of the RAVS scheme (see Theorem 1). The adversary even has no information which permission he needs.

Contamination attack. We consider contamination attacks in Figure 2.

Global adversary Suppose that AS3 wants to redirect all traffic destined to customers AS4, AS5 and AS6 through itself not AS2 by making AS1 believe that there is no connection between AS2 and AS4. AS3 has to remove one of RAVS($d_{\mathsf{AS2}}, \mathsf{AS4}$) and RAVS($d_{\mathsf{AS4}}, \mathsf{AS2}$) or both from the mIRR. For doing this, AS3 must construct the exact same registries stored in the mIRR. AS3 cannot perform this task since it is impossible to construct the valid registries without the private key d_{AS2} and d_{AS4} by the existential unforgeability of the RAVS scheme (see Theorem 2).

Local adversary If AS2 is compromised by the attacker, that attacker now has the AS2's private key d_{AS2}. For making other ASes accept its incorrect UPDATE message, the compromised AS submits a fake registry to the mIRR consistent with the route in the fake UPDATE message. Here, AS2 is the adversary and can construct valid registries using the private key d_{AS2}. Suppose that AS2 has uploaded RAVS($d_{\mathsf{AS2}}, \mathsf{AS5}$) for the fake connection between AS5 and itself to the mIRR, and AS1 receives the fake UPDATE message from AS2. AS1 can issue two queries $\langle \mathsf{AS2}, \mathcal{P}_{\mathsf{AS5}} \rangle$ and $\langle \mathsf{AS5}, \mathcal{P}_{\mathsf{AS2}} \rangle$ to the mIRR. The response from the mIRR should be '1' for the first query and '0' for the second one. AS1 notices that the received UPDATE message is incorrect and drops it. Note that we use the network topology's mutuality: If node 1 has connection with node 2, node 2 also has connection with node 1.

7. EVALUATION

Implementation. We implemented the RAVS scheme using the Stanford IBE library [31]. This library uses the supersingular elliptic curve $E : y^2 = x^3 + 1$ defined over $\mathbb{F}_p$. The group $\mathbb{G}_1$ is an order q subgroup of the group of points on the curve E over $\mathbb{F}_p$. The group $\mathbb{G}_2$ is an order q subgroup of $\mathbb{F}^*_{p^2}$. We set p as a 512-bits prime and q as an 160-bits prime in our implementation. We measured the performance on an Intel Xeon 3.0GHz processor machine running RedHat Enterprise Linux v.3 with 2GB of memory.

Table 1 shows the number of operations and the running time of each algorithm. Since the exponentiation in $\mathbb{G}_2$ is faster than the scalar multiplication in $\mathbb{G}_1$, we replaced the latter with the former using bilinearity in the second steps of RAVS and TestRegistry algorithms. The cost of an inversion in $\mathbb{Z}^*_q$ is negligible in the Permission algorithm. Note that we consider the cost of computing hash functions H_1 and H_2. The reason is that the cost of these functions is not negligible in our measurement.[2]

Table 2 shows the order and size of each message for analyzing storage requirements and communication overheads. We need $2 \log q$ bits for representing a point in $\mathbb{G}_1$. Since a registry tuple consists of three $\mathbb{G}_1$ points, the size of one registry is $3 \times 2 \log q$ bits. A search query consists of one identity and one permission. Since the identity is AS number and the total number of ASes is currently less than 20,000 in the Internet, 16 bits are enough to represent one identity. One permission consists of one $\mathbb{G}_1$ point and an integer in $\mathbb{Z}^*_q$. Hence, we need $3 \log q$ bits for representing one permission. Finally, a test result for one search query requires only one bit for representing 'yes' or 'no'.

Optimization. Table 1 and 2 also shows the optimized time and required space for precomputation. We considered two optimizations to improve the performance of our scheme. First, we can reuse a random number when constructing registries as mentioned in [34, 10]. If we allow to reuse one random number r to construct multiple registries, the time for computing $U_1 = rT$ and $U_2 = rP$ can be saved. Second, we can speed up TestRegistry by precomputing two pairings $\hat{e}(P, V)$ and $\hat{e}(S, Q_{\mathsf{ID}})$. Since these two values are in $\mathbb{F}^*_{p^2}$, it requires 1024-bits to represent each element when p is a 512-bits prime. Instead of storing these two values, we do not need to store V value.

8. DISCUSSION

We discuss several issues which were not covered earlier. First, our RAVS can be used to address the prefix origin validation. In this case, the identity of the RAVS is the address prefix and the keyword is AS number of that prefix owner. Internet Corporation for Assigned Names and Numbers (ICANN) could be the registry provider and it has the private keys corresponding to all possible address prefixes.

Second, routing policies are specified by the Routing Policy Specification Language (RPSL) [3, 2]. How to construct RAVS based on this structured language is our future work.

Third, our method cannot address the colluding adversaries. If two malicious ASes submit the registries of their

[2]As mentioned in [7], it is practically difficult to construct hash functions that map directly onto $\mathbb{G}_1$. The hash function onto $\mathbb{G}_1$ in [31] is an alternative function proposed in [7].

Table 1: Number of operations and timing for running each algorithm.

	Extract	RAVS	Permission	TestRegistry
Pairing ($\hat{e}$) computation	0	1	0	3
Scalar multiplication in $\mathbb{G}_1$	1	3	1	1
Exponentiation in $\mathbb{G}_2$	0	1	0	1
H_1, H_2 computations	1	1	1	1
Time (msec)	21.38	43.50	21.53	54.63
Optimized time (msec)	-	34.93	-	31.89

Table 2: Size of each message when q is an 160-bits prime and $|\mathsf{ID}| = 16$ (bits).

Message type	Original size		Size with precomputation	
	Order	Bitlength	Order	Bitlength
Registry	$O(\log q^6)$	960	$O(\log (pq)^4)$	2688
Query ($\mathsf{ID} + \mathcal{P}_\mathsf{W}$)	$O(\|\mathsf{ID}\| + \log q^3)$	496	-	-
Test result	$O(1)$	1	-	-

bogus connection, we cannot but believing that connection is valid. Actually, most of current countermeasures of routing protocols have difficulties to address this problem.

Finally, we need to consider deployment issues of our scheme. Suppose that not all ASes adopt our scheme on the Internet. The mIRR is not complete in this case. Though our scheme can still be used, the probability to detect invalid route becomes lower. The more ASes participate in the mIRR construction, the higher the probability of successful detection is. Thus, we issue the search permission as an incentive to the ASes. Since ASes need to obtain search permissions for querying the mIRR, ASes should participate in the mIRR construction to make the routing registry complete.

9. CONCLUSION

In this paper, we defined a RAVS scheme. We introduced the concept of authorized search using a trusted third party, the Search Permission Generator (SPG). A registry is constructed by the registry provider's private key and a keyword. It can be tested by the registry provider's identity and the corresponding search permission. All these processes reveal no information about both the registry and the search query. Searching on a registry is possible, only if the search permission from the SPG is available. Moreover, we can verify the registry provider by its identity.

We constructed a RAVS scheme and proved its securities. Our scheme guarantees two securities simultaneously. First, an adversary can learn no information from the registry without search permission. Second, an adversary cannot construct a valid registry without the correct private key.

We can apply our RAVS scheme in BGP to construct the Internet routing information securely. We can utilize this routing information to detect invalid routes efficiently. Our scheme enables ASes to have incentives to contribute their routing information and furthermore to make the information complete.

10. REFERENCES

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. In *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 205–222. Springer-Verlag, 2005.

[2] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra. *Routing Policy Specification Language (RPSL). RFC 2622*, Jun 1999.

[3] T. Bates, E. Gerich, L. Joncheray, J.-M. Jouanigot, D. Karrenberg, M. Terpstra, and J. Yu. *Representation of IP Routing Policies in a Routing Registry (ripe-81++). RFC 1786*, Mar 1995.

[4] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS)*, pages 62–73, Nov 1993.

[5] S. M. Bellovin and W. R. Cheswick. Privacy-Enhanced Searches Using Encrypted Bloom Filters. Technical Report Draft, AT&T Labs – Research, Feb 2004.

[6] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public-Key Encryption with keyword Search. In *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer-Verlag, 2004.

[7] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, 32(3):586–615, Mar 2003.

[8] V. J. Bono. 7007 Explanation and Apology. `http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html`.

[9] X. Boyen. Multipurpose Identity-Based Signcryption: A Swiss Army Knife for Identity-Based Cryptography. In *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 383–399. Springer-Verlag, 2003.

[10] R. W. Bradshaw, J. E. Holt, and K. E. Seamons. Concealing Complex Policies with Hidden Credentials. In *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS)*, pages 146–157, Oct 2004.

[11] K. Butler, T. Farley, and P. McDaniel. A Survey of

BGP Security. Technical Report TD-5UGJ33, AT&T Labs – Research, Feb 2004.

[12] Y.-C. Chang and M. Mitzenmacher. Privacy Preserving Keyword Searches on Remote Encrypted Data. Cryptology ePrint Archive, Report 2004/051, 2004. Available at `http://eprint.iacr.org/2004/051/`.

[13] J.-S. Coron. On the Exact Security of Full Domain Hash. In *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235. Springer-Verlag, 2000.

[14] E.-J. Goh. Secure Indexes. Cryptology ePrint Archive, Report 2003/216, 2003. Available at `http://eprint.iacr.org/2003/216/`.

[15] P. Golle, J. Staddon, and B. Waters. Secure Conjunctive Keyword Search over Encrypted Data. In *Proceedings of the 2nd International Conference on Applied Cryptography and Network Security (ACNS)*, volume 3089 of *Lecture Notes in Computer Science*, pages 31–45. Springer-Verlag, 2004.

[16] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin. Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing. In *Proceedings of the 10th Network and Distributed System Security Symposium (NDSS)*, Feb 2003.

[17] Y.-C. Hu, A. Perrig, and M. Sirbu. SPV: Secure Path Vector Routing for Securing BGP. In *Proceedings of ACM SIGCOMM'04*, Aug 2004.

[18] Internet Routing Registry. `http://www.irr.net`.

[19] S. Kent, C. Lynn, J. Mikkelson, and K. Seo. Secure Border Gateway Protocol (S-BGP) – Real World Performance and Deployment Issues. In *Proceedings of the 7th Network and Distributed System Security Symposium (NDSS)*, Feb 2000.

[20] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (S-BGP). *IEEE Journal on Selected Areas in Communications*, 18(4):582–592, Apr 2000.

[21] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur. Topology-based Detection of Anomalous BGP Messages. In *Proceedings of the Symposium on Recent Advances in Intrusion Detection (RAID)*, Sep 2003.

[22] B. Libert and J.-J. Quisquater. The Exact Security of an Identity Based Signature and its Applications. Cryptology ePrint Archive, Report 2004/102, 2004. Available at `http://eprint.iacr.org/2004/102`.

[23] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP Misconfiguration. In *Proceedings of ACM SIGCOMM'02*, Aug 2002.

[24] S. Murphy. BGP Security Vulnerabilities Analysis. IETF Internet Draft, Oct 2004. `draft-ietf-idr-bgp-vuln-01.txt`.

[25] J. Ng. Extensions to BGP to Support Secure Origin BGP (soBGP). IETF Internet Draft, Apr 2004. `draft-ng-sobgp-bgp-extensions-02.txt`.

[26] O. Nordström and C. Dovrolis. Beware of BGP Attacks. *ACM SIGCOMM Computer Communications Review*, 34(2):1–8, Apr 2004.

[27] D. J. Park, K. Kim, and P. J. Lee. Public Key Encryption with Conjunctive Field Keyword Search. In *Proceedings of the 5th International Workshop on Information Security Applications (WISA)*, volume 3325 of *Lecture Notes in Computer Science*, pages 73–86. Springer-Verlag, 2004.

[28] Y. Rekhter and T. Li. *A Border Gateway Protocol 4 (BGP-4). RFC 1771*, Mar 1995.

[29] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Advances in Cryptology – CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.

[30] D. X. Song, D. Wagner, and A. Perrig. Practical Techniques for Searches on Encrypted Data. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 44–55, May 2000.

[31] Stanford Applied Crypto Group. IBE Secure E-mail. `http://crypto.stanford.edu/ibe/`.

[32] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz. Listen and Whisper: Security Mechanisms for BGP. In *Proceedings of First Symposium on Networked Systems Design and Implementation (NSDI)*, Mar 2004.

[33] T. Wan, E. Kranakis, and P. C. van Oorschot. Pretty Secure BGP (psBGP). In *Proceedings of the 12th Network and Distributed System Security Symposium (NDSS)*, Feb 2005.

[34] B. R. Waters, D. Balfanz, G. Durfee, and D. K. Smetters. Building an Encrypted and Searchable Audit Log. In *Proceedings of the 11th Network and Distributed System Security Symposium (NDSS)*, pages 205–214, Feb 2004.

[35] R. White. Securing BGP Through Secure Origin BGP. *The Internet Protocol Journal*, 6(3):15–22, Sep 2003.

[36] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Detection of Invalid Routing Announcement in the Internet. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN)*, pages 59–68, Jun 2002.