ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
DOI: 10.17559/TV-20150314133532

A DSA-BASED SCHEME FOR DEFENDING AGAINST IP PREFIX HIJACKING WITHOUT REPOSITORIES

Bo Yang

Original scientific paper

IP prefix hijacking poses a serious threat to the security of the Internet. Cryptographic authenticating origin ASes (Autonomous Systems) of advertised prefix, which is an effective way of preventing IP prefix hijacking, has received wide acceptance. However, these existing schemes received various critical comments on their inefficiency when cryptographic authenticating origin ASes. For improving efficiency, we take full advantage of specific characteristics of DSA (Digital Signature Algorithm) and thus present a scheme for preventing IP prefix hijacking. There are two characteristics, which are DSA-based and efficient, in the proposed scheme. Firstly, because DSA is a United States Federal Government standard for digital signatures, the DSA-based can maintain compatibility with the DSA and its analytical tools, and thus it is easier for proposed scheme to be widely accepted and applied into practice. Secondly, public key certificates are not necessary because public keys can be computed by using a formula. Separated verifying signatures in these certificates, which are inevitable in almost all existing cryptography-based schemes, can be replaced with computing of a multi-exponentiation formula. Thus, the efficiency is achieved.

Keywords: *certificates; DSA; IP prefix hijacking; multi-exponentiation*

Shema za obranu od krađe IP prefiksa bez repozitorija utemeljena na DSA

Izvorni znanstveni članak

Krađa IP prefiksa predstavlja ozbiljnu prijetnju za sigurnost Interneta. Kriptografsko ustanovljavanje autentičnosti porijekla ASes (Autonomnih Sustava) oglašenog prefiksa, što predstavlja učinkovit način sprećavanja krađe IP prefiksa, široko je prihvaćeno. Međutim, postojećim se shemama upućuju različiti kritički komentari vezani za njihovu neučinkovitost kod kriptografskog ustanovljavanja autentičnosti porijekla ASes. U svrhu poboljšanja učinkovitosti, koristimo prednosti specifičnih obilježja DSA (Digital Signature Algorithm) te predstavljamo shemu za sprećavanje krađe IP prefiksa. Postoje dva obilježja predložene sheme, temeljena na DSA i učinkovita. Prvo, budući da je DSA standard za digitalne potpise federalne vlade SAD, DSA temeljeno obilježje može zadržati kompatibilnost s DSA i njegovim analitičkim alatima te je na taj način olakšano široko prihvaćanje i primjena u praksi predložene sheme. Drugo, državni ključni certifikati (key certificates) nisu potrebni jer se mogu izračunati pomoću formule. Odvojeni potpisi za verifikaciju u tim certifikatima, koji su neizbježni u gotovo svim postojećim shemama temeljenim na kriptografiji, mogu se zamijeniti računanjem multi-eksponencijalne formule. Na taj je način postignuta učinkovitost.

Ključne riječi: *certifikati; DSA; krađa IP prefiksa; multi-eksponencijacija*

1 Introduction

In the Internet, networks share information via routers. A group of routers under the same administrative control is considered an autonomous system [15]. There are about 46 000 Autonomous Systems (ASes) [1] in the Internet (see Fig. 1). Border Gateway Protocol (BGP), which is the de-facto protocol enabling interdomain routing in the Internet, cannot authenticate origin ASes when update messages are broadcast among ASes. An AS can advertise a prefix from address space unassigned by or belonging to another AS. This kind of attack, an example of which is shown as Fig. 2, is called IP prefix hijacking [2, 3].

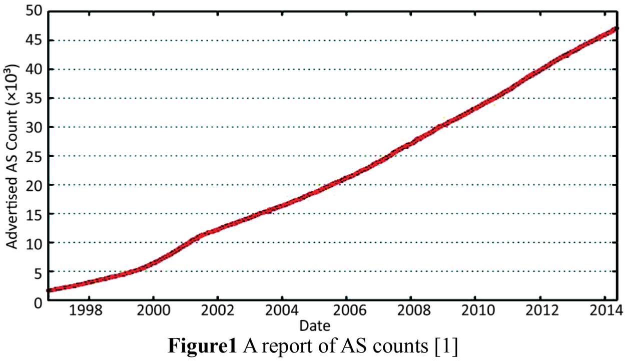

Figure1 A report of AS counts [1]

Left part of Fig. 2 presents how update messages and traffic broadcast when there is no prefix hijacking. AS_0 is owner of IP prefix 129.82.0.0/16. It sends an update message to AS_4 for announcing itself is the origin AS of this prefix. This announcing is shown as a dotted directed line from AS_0 to AS_4. If this announcing is accepted, AS_4 will send AS_0 all of traffic whose destination is 129.82.0.0/16, which is shown as a full line from AS_4 to AS_0. AS_4 sends an update message to AS_3 for announcing interdomain path <129.82.0.0/16, AS_0, AS_4>, which is shown as a dotted directed line from AS_4 to AS_3. If this announcing is accepted, AS_3 will send AS_4 almost all of traffic whose destination is 129.82.0.0/16, which is shown as a full line from AS_3 to AS_4. Each of ASes in the route will append itself the number of AS into AS-PATH it receives from upstream neighbour AS. Left part of Fig. 2 presents how update messages and traffic broadcast when there is no prefix hijacking. AS_0 is owner of IP prefix 129.82.0.0/16. It sends an update message to AS_4 for announcing itself is the origin AS of this prefix. This announcing is shown as a dotted directed line from AS_0 to AS_4. If this announcing is accepted, AS_4 will send AS_0 all of traffic whose destination is 129.82.0.0/16, which is shown as a full line from AS_4 to AS_0. AS_4 sends an update message to AS_3 for announcing interdomain path <129.82.0.0/16, AS_0, AS_4>, which is shown as a dotted directed line from AS_4 to AS_3. If this announcing is accepted, AS_3 will send AS_4 almost all of traffic whose destination is 129.82.0.0/16, which is shown as a full line from AS_3 to AS_4. Each of ASes in the route will append itself number of AS into AS-PATH it receives from upstream neighbour AS.

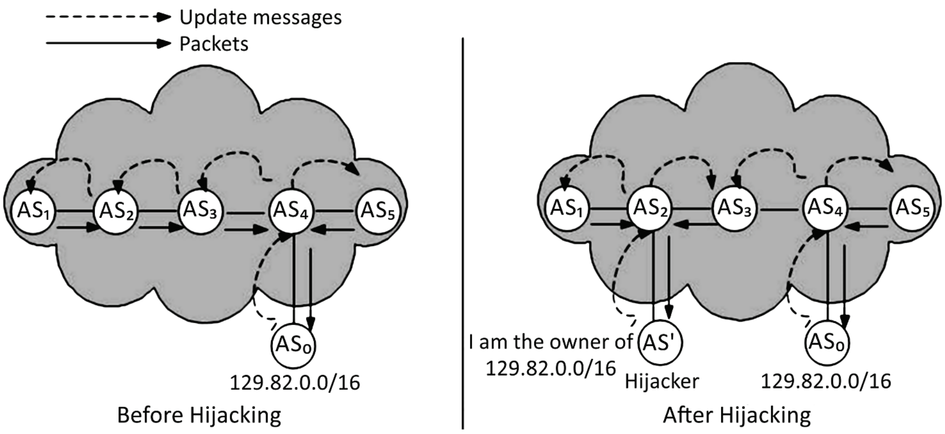

Figure 2 An example of IP prefix hijacking

Right part of Fig. 2 presents how AS', which is **not** the true origin AS (or owner) of 129.82.0.0/16, launches IP prefix hijacking for attracting traffic whose origin destination is 129.82.0.0/16. AS', who is malicious and pretends to be origin AS of IP prefix 129.82.0.0/16, sends an update message to AS_2 in order to announce itself is the origin AS of IP prefix 129.82.0.0/16. Comparing with true path <129.82.0.0/16, AS_0, AS_4, AS_3, AS_2>, fake path <129.82.0.0/16, AS', AS_2> is shorter and thus more attractive, almost all traffic sent by AS_2 and destinated to 129.82.0.0/16, will be redirected to AS'. This attack launched by AS' cannot only pollute AS_2, but also can pollute many other ASes by broadcast of update messages. As a result, AS' succeeds in pretending to be origin AS of targeted prefix and hijacking some traffic. This kind of attack is called IP prefix hijacking. If some measurements can prevent malicious AS from succeeding in pretending to be origin AS (or owner) of targeted prefix, then the security of the Internet about IP prefix hijacking is achieved.

In the way described above, update messages and traffic broadcast throughout the whole Internet.

Any AS whose prefix is hijacked may experience reachability problems and cannot easily identify the actual cause [3]. IP prefix hijacking is essentially a special form of denial of service attack. Hijacked prefixes can also be used for carrying out malicious activities, raising the challenge of identifying the actual perpetrator [2].

IP prefix hijacking poses a serious threat on the security of the Internet such as the traffic hole on global scale brought about by AS7007 [4], the access interruption of Youtube caused by misoperation of Pakistan Telecom (AS17557) [5], interception induced by China's misconfiguration [6], and so on.

Most existing proposals on prefix hijack fall into two categories. The first category is based on cryptography [8] [9÷13], and the second category is based on detection [3, 16÷19].

Cryptographic authenticating origin ASes (Autonomous Systems) of advertised prefix, which is an effective way of preventing IP prefix hijacking, has received wide acceptance [15].

The reason why IP prefix hijacking is caused is that there is no signal for verifier to judge whether an AS is the origin AS (or owner) of targeted prefix or not. Certainly, the signal should be unchangeable. If there is this kind of signal, malicious AS cannot pretend to be origin AS of targeted prefix, because this pretending can inevitably be found and prohibited. If this signal is a cryptographic digital signature (including asymmetric and symmetric signature), then corresponding measurements are called cryptographic authenticating origin ASes of advertised prefixes.

For instance, in S-BGP, the typical method of cryptographic authenticating origin ASes, the prefix owner has an asymmetric private key for each prefix, generated by a global trust entity. A digital signature called as address attestation, is created by the owner using its private key. This address attestation is used to signal whether the AS is origin AS of targeted prefix or not. Each AS along a path will verify that the prefix actually belongs to the AS with the corresponding public key.

Cryptographic authenticating origin ASes of advertised prefixes, as the fundamental method used by asymmetric cryptography based solutions to prevent IP prefix hijacking, can be divided into two parts [7]. In the first part, owners of prefixes sign the AS numbers (AS#s) of origin ASes using themselves private keys. In the second part, receivers of update messages verify these signatures using public keys corresponding to advertised prefixes.

However, these existing cryptography-based schemes received various critical comments on their computational inefficiency when cryptographic authenticating origin ASes [15]. Efficiency is important for authenticating update messages because BGP speakers receive large amounts of such messages, and sometimes they arrive in bursts. These bursts always are brought about when network topology changes. Data packets will not be correctly routed until routing reconverges, as a result of which efficient verification during these bursts is of utmost importance. Moreover, efficient authentication is more desirable since periodic update messages rather than only event driven messages may be required [2].

There are various reasons why existing schemes are inefficient. Separate verifications of digital signatures and frequent communication with out-band repositories are obviously two obstacles to efficiency of existing schemes, which can be to a certain extent overcome by the presented scheme.

For improving efficiency, we take full advantage of specific characteristics of DSA (Digital Signature Algorithm) and thus present a scheme for preventing IP prefix hijacking, where DSA [26] is the abbreviation of

Digital Signature Algorithm proposed by the U.S. National Institute of Standards and Technology (NIST). It obviously belongs to asymmetric cryptography based solutions as S-BGP [8], SoBGP [9], psBGP [10], and OA [11].

There are two characteristics, which are DSA-based and efficient, in the proposed scheme. Firstly, because DSA is a United States Federal Government standard for digital signatures, the DSA-based can maintain compatibility with the DSA and its analytical tools, and thus it is easier for proposed scheme to be widely accepted and applied into practice. Secondly, public key certificates are not necessary because public keys can be computed by using a formula. Separated verifying signatures in these certificates, which are inevitable in almost all existing cryptography-based schemes, can be replaced with computing of a multi-exponentiation formula. Moreover, there are no repositories, whose deployment and management are removed. Therefore, the proposed scheme is efficient.

The rest of paper is organized as follows. Section 2 presents the proposed scheme, which includes the framework and details of each part of this framework. In Section 3, we explain two theoretical bases including DSA and existing authentication of origin AS, on which proposed scheme is based. Section 4 discusses the result of our work. Section 5 compares proposed scheme with S-BGP according to size of related information, and convergence time. Finally, we present the conclusion and the future works in Section 6.

2 DSA-based method for authenticating origin autonomous systems

In this section, we present our scheme, the cryptographic basis of which is the DSA. We firstly describe the framework of the proposed scheme, which is followed by details of each part of this framework.

2.1 Framework of proposed scheme

In existing asymmetric cryptography based methods for authenticating origin ASes, overall steps have been formed, which can be described as follows [8÷10] (see Fig. 3).

Firstly, public/private key pairs are issued to the owner of advertised prefix. Secondly, an address attestation for prefix$_i$ is created by the owner. The address attestation is a signature signed by the owner's private key. Thirdly, the owner uploads this address attestation to repositories from which all of BGP speakers can download this address attestation. Obviously, this distributing is out of band rather than in band (in update messages). Therefore, in most cases, the address attestation should be distributed to all of ASes before corresponding update messages are sent. S-BGP [8] suggests that repositories should be used from which every AS can download the entire address attestations, certificates, and CRLs (see Fig. 4). Fourthly, this address attestation can be verified by all of ASes. Certainly, the public key of owner of advertised prefix should also be gotten before verifying. Public keys usually exist in public key certificates. Before verifying, verifiers should authenticate public keys.

We modify above framework as following in order to improve efficiency. Creators of address attestations do not upload address attestations to repositories any more. These address attestations are put into update messages after verifying them. Address attestations can be drawn out directly from update messages, and be verified immediately. In one word, address attestations are regarded as components of update messages in proposed scheme (see Fig. 5).

Next, we compare the verifying framework of RPKI [7] (or S-BGP [8]) with that of proposed scheme in more detail.

In RPKI or S-BGP, for authenticating origin autonomous systems, repositories spreading all over the Internet are necessary. These repositories have two main functions. One function is that address attestations, related public key certificates, and CRLs can be uploaded to these repositories by IANA, RIRs, key issuers, and owners of prefixes. The other function is that every ASes can download necessary objects from these repositories.

As shown in Fig. 4, in RPKI, relying ASes download and verify RPKI objects out of band (rather in real time as part of BGP), and RPKI objects are uploaded and stored at directories that are controlled by their issuers [7, 26]. The main obstacle to put RPKI objects into update messages is that an update message is limited in length to 4096 bytes and thus update messages are too small to carry the necessary public key certificates for most update messages (note that an x. 509 public key certificate would be about 500 ÷ 1000 bytes long [8]).

In proposed scheme, it is not necessary for authenticating origin autonomous systems to deploy repositories because update messages can carry necessary information related public keys. In RPKI or S-BGP, information related public key includes public key certificates. In proposed scheme, there are no public key certificates in information related public keys. Therefore, information related public key is much smaller than that of RPKI. Public key certificates are mainly used to authenticate public keys. In our scheme, it is not necessary for authenticating public keys to resort to certificates.

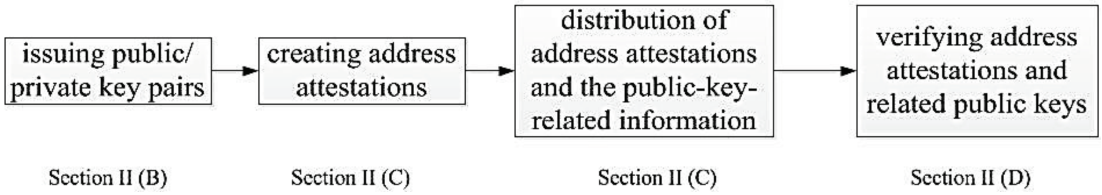
issuing public/ private key pairs
creating address attestations
distribution of address attestations and the public-key-related information
verifying address attestations and related public keys
Section II (B)
Section II (C)
Section II (C)
Section II (D)

Figure 3 Overall steps of existing schemes for origin authentication

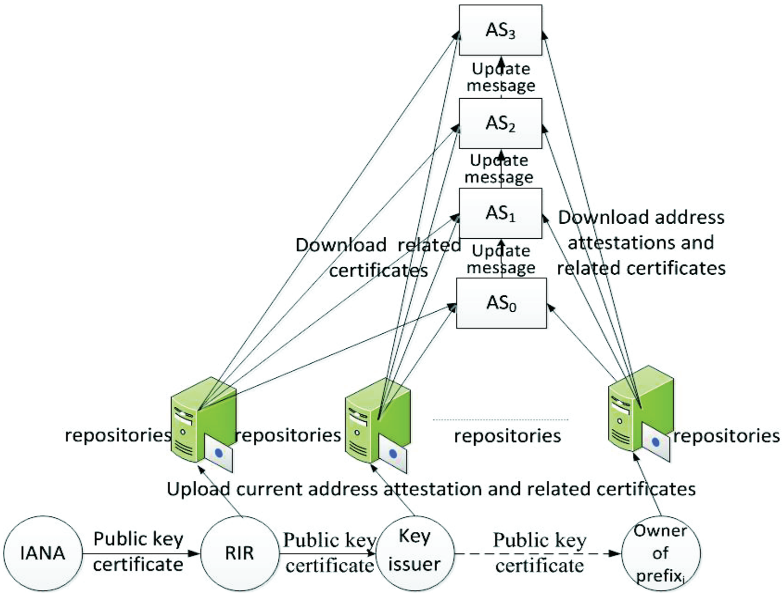

Figure 4 The framework of authenticating origin AS in typical existing scheme

As shown in Fig. 5, current address attestations and information related public keys are sent to origin AS (denoted as AS_0), which are broadcast in update messages. Relying ASes can draw them directly from update messages. It is obviously more convenient and efficient than RPKI or S-BGP. There are no repositories, which can relieve burden of cost of purchasing, deploying, and managing repositories all over the Internet.

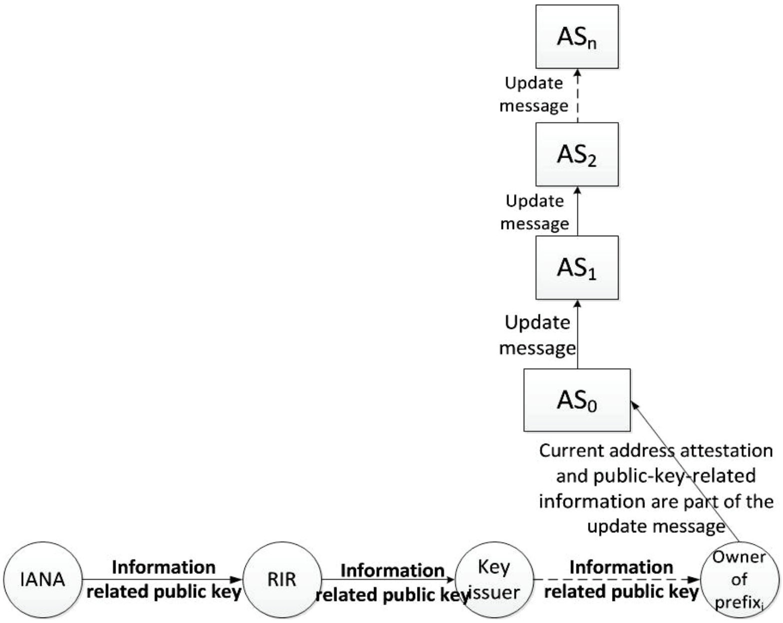
ASn
Update message
AS2
Update message
AS1
Update message
AS0
Current address attestation and public-key-related information are part of the update message
IANA
Information related public key
RIR
Information related public key
Key issuer
Information related public key
Owner of prefix$_i$

Figure 5 The framework of authenticating origin AS in proposed scheme

2.2 Public/private key pairs issuing for prefix owners

In proposed scheme, public/private key pairs have to be issued to prefix owners. Private keys, which are in these key pairs, are used to create address attestations by prefix owners. IANA (Internet Assigned Numbers Authority) acts as trust root in our scheme. IANA is regarded as initial owner who is owner of all possible prefixes. The IANA issues some large prefixes to some organizations and makes them owners of these prefixes who in turn issue parts of these prefixes to other organizations and make them owners of these smaller prefixes.

Shown as Fig. 6, steps of key pairs issuing are initiation, key pair issuing, and DSA-based validation of key pairs, for which we will give detailed explanation.

The DSA-based algorithms for issuing keys are as follows.

(1) Initiation: Public/private key pair itself and parameters of the whole signature/verification system are created by IANA. This initiation uses the key generation for DSA [26], whose steps are as follows.

Algorithm 1: Initiation

```
1. Select a prime number $q$ which is 160 bits.
2. Choose $t$ so that $0 \le t \le 8$, and select a prime number $p$ where $2^{511} + 64t < p < 2^{512} + 64t$, with the property that q divides $(p-1)$.
3. Select a generator $g$, whose order is $q$, of the unique cyclic group $Z_p^*$.
```

```
    3.1 Select an element a in $Z_p^*$ and compute $g = a^{(p-1)/q} \bmod p$.
    3.2 If $g$ =1 then go to step 3.1.
4. Select a random integer $s_0$ such that $1 \le s_0 \le q-1$.
5. Compute $PK_0 = g^{s0}$ mod p.
6. IANA's public key is ($p$, $q$, $g$, $PK_0$); IANA's private key is $s_0$.
```

(2) Key pairs issuing: Shown as Fig. 6, key pairs issuing will be elaborated in three aspects. The three aspects include hierarchical issuing of key pairs, secure and hierarchical issuing of key pairs, control of validity period. Algorithm 2 is about hierarchical issuing of key pairs, Algorithm 3 is about secure and hierarchical issuing of key pairs, and Algorithm 4 is about control of validity period.

We assume current prefix is prefix$_i$, and prefix$_{i-1}$ is parent prefix of prefix$_i$. Address block denoted by prefix$_{i-1}$ contains address block denoted by prefix$_i$.

We assume that public/private key pair corresponding prefix$_{i-1}$ is pk$_{i-1}$/s$_{i-1}$. A DSA-based signature is signed by the owner of prefix i. For issuing owner of prefix $_i$ public/private key pair, which is as the following algorithm.

Algorithm 2: Hierarchical issuing of key pairs

```
1. Select a random secret integer $k_i$, $0 < k_i < q$
2. Compute $r_i = (r_{i-1}^{k_i} \bmod p)$, $r_i' = r_i \bmod q$
3. Compute $k_i^{-1} \bmod q$
4. Compute $s_i = k_i^{-1}(h(\text{prefix}_i\#) + r_i' \cdot s_{i-1}) \bmod q$
   ($h()$ is a hash function h: $\{0,1\}^* \to Z_q$)
5. The signature for prefix$_i$# signed by the owner of
   prefix $_{i-1}$ is the pair ($r_i$, $s_i$)
6. The owner of prefix$_{i-1}$ sends the pair ($r_i$, $s_i$) and
   $r_1, r_2, \ldots, r_{i-1}$ to the owner of prefix$_i$, where $s_i$ acts
   as the private key of prefix$_i$
```

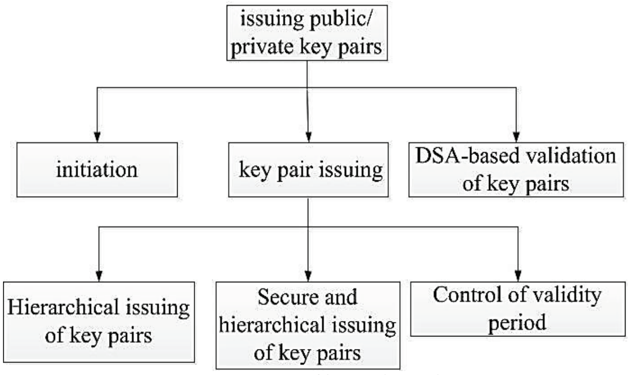
issuing public/ private key pairs
initiation
key pair issuing
DSA-based validation of key pairs
Hierarchical issuing of key pairs
Secure and hierarchical issuing of key pairs
Control of validity period

Figure 6 Steps of key Pair Issuing

There are the following two differences between algorithm 2 and DSA:

The first difference: The r_i' in resultant signature is being replaced with the r_i. If the signature is a DSA signature on prefix $_i$# by the corresponding private key of prefix$_{i-1}$, the signature should be the pair (r'_i, s_i), rather than the pair (r_i, s_i). The reason why we make this transform will be explained in the Subsection 2.4.

The second difference: R$_i$ and r_i' are computed from r_{i-1} i rather than g. Note that r_0 is just the g. In DSA, r_i and r_i' are computed from g, where r_i' = (g^{ki} mod p) mod q. The reason why we make this transform will be explained in the Subsection 2.4.

According to ability of an authority impersonating users, there are three levels of trust [27] as follows in Tab. 1.

Table 1 Three levels of trust

level 1	The users' secret keys are known by the authority who thus can impersonate any user without being detected.
level 2	The users' secret keys are not known by the authority who is still capable of impersonating any user by generating false certificates that may be used without being detected.
level 3	The users' secret keys are not known by the authority who will be detected if it generates false certificates for users.

Clearly, the level 3 is the most desirable one. Above Algorithm 2 only reaches trust level 1 because the owner of prefix$_{i-1}$ knows the private key (which is s_i) of owner of prefix$_i$, which may be insufficient in authenticating origin autonomous systems.

We modify Algorithm 2 to reach trust level 3 by using a kind of weak blind signature, the detail steps of which can be seen in Algorithm 3.

Because the weak blind signature is introduced, the secret key s$_i$ can be hidden to the owner of prefix $_i$ who is even if the issuer of this private key, which causes that Algorithm 3 reaches trust level 3.

The weak blind signature introduced by Algorithm 3 is elaborated as follows.

The owner of prefix$_{i-1}$, whereas it creates the $\tilde{s}_i$ does not know the random secret integer $\tilde{k}_i$. Thus, it cannot compute $s_i = (\tilde{k}_i)^{-1}\gamma\tilde{s}_i$.

The owner of prefix$_{i-1}$ cannot create $\tilde{s}_i$ which can go through verification of the owner of prefix $_I$ if it does not execute step 3 of Algorithm 3, which is because the owner of prefix$_i$ multiples $\tilde{s}_i$ by $(\tilde{k}_i)^{-1}$ to compute s_i in step 11 of Algorithm 3, and verifies s_i by using r_i sent from the owner of prefix$_{i-1}$.

Fig. 7 shows the deliveries of information between the key issuer and key receipter for creating a weak blind signature. Key receipter delivers $\tilde{r}_i$ to key issuer. The $\tilde{r}_i$ computedly links to secret random integer $\tilde{k}_i$ which is hidden to key issuer. According to intractability of discrete logarithm problem, key issuer cannot compute $\tilde{k}_i$ from $\tilde{r}_i$ because $\tilde{r}_i = r_{i-1}^{\tilde{k}_i} \bmod p$.

Algorithm 3: Secure and hierarchical issuing of key pairs

```
1. The owner of prefix$_{i-1}$ send $r_1, r_2, \ldots, r_{i-1}$ to the
   owner of prefix
2. The owner of prefix$_i$ select a random secret
   integer $\tilde{k}_i$, $0 < \tilde{k}_i < q$.
3. Compute $\tilde{r}_i = (r_{i-1}^{\tilde{k}_i} \bmod p)$
```

```
4. The owner of prefix$_i$ send $\widetilde{r_i}$ to the owner of
   prefix$_{i-1}$
5. The owner of prefix$_{i-1}$ select a random secret
   integer $k_i$, $0 < k_i < q$
6. Compute $r_i = (\widetilde{r_i} \cdot r_{i-1}^{k_i} \bmod p)$, $r' = r_i \bmod q$
7. Compute $k_i^{-1} \bmod q$
8. Compute $\widetilde{s_i} = k_i^{-1}[h(prefix_i \#) + r' \cdot s_{i-1}] \bmod q$
   ($h()$ is a hash function h: $\{0,1\}^* \rightarrow Z_q$)
9. The signature for prefix$_i$ # signed by the owner of
   prefix$_{i-1}$ is the pair $(r_i, \widetilde{s_i})$
10. The owner of prefix$_{i-1}$ send the pair $(r_i, \widetilde{s_i})$ and $r_1$,
    $r_2$, ..., $r_{i-1}$ to the owner of prefix$_i$
11. The owner of prefix$_i$ compute $s_i = (\widetilde{k_i})^{-1} \cdot \widetilde{s_i}$
    mod $q$
12. $s_i$ acts as the private key of prefix$_i$
```

On the other hand, key issuer has to multiply $r_{i-1}^{k_i}$ by $\widetilde{r_i}$ to get a suitable r_i. Otherwise, the r_i cannot be corresponding to s_i because s_i is the result of multiplying $\widetilde{s_i}$ by $(\widetilde{k_i})^{-1}$.

Therefore, the pair (r_i, s_i) is a weak blind signature of key issuer for message prefix$_i$ #. The s_i, which is part of the signature and only known by key receipter, is hidden to key issuer and thus secure when treated as private key.

Comparing with Algorithm 2, Algorithm 3 can hide s_i from the owner of prefix$_{i-1}$. However, there is no control of validity period of public keys in Algorithm 3, which is used to reduce the impact of replay attacks in proposed scheme.

If there is no replay protection, replay attacks, where a legitimate prefix advertisement which has been previously heard (even if this advertisement has been drawn), can be launched by malicious attackers.

In proposed scheme, we prevent replays through the use of validity period: Each route is read advertised in the validity period. A validity period is an implicit timeout: the advertised route times out after the validity period ends.

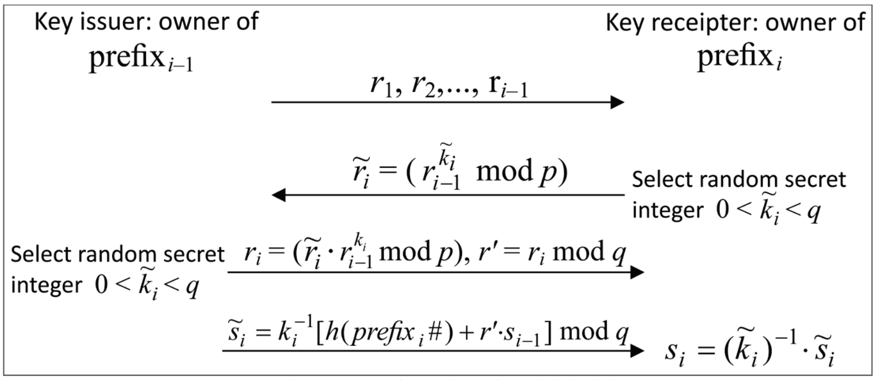
Key issuer: owner of prefix$_{i-1}$
Key receipter: owner of prefix$_i$
$r_1, r_2, \ldots, r_{i-1}$
$\widetilde{r_i} = (r_{i-1}^{\widetilde{k_i}} \bmod p)$
Select random secret integer $0 < \widetilde{k_i} < q$
Select random secret integer $0 < \widetilde{k_i} < q$
$r_i = (\widetilde{r_i} \cdot r_{i-1}^{k_i} \bmod p)$, $r' = r_i \bmod q$
$\widetilde{s_i} = k_i^{-1}[h(prefix_i \#) + r' \cdot s_{i-1}] \bmod q$
$s_i = (\widetilde{k_i})^{-1} \cdot \widetilde{s_i}$

Figure 7 The process of creating of weak blind signature

We can choose the length of a validity period in a way that provides higher security or lower overhead. Yet, a minimum period should be set such as one day at least. Some attackers may launch DoS (Denial-of-Service) attacks by excessive validity period changing. This kind of DoS attacks can be prevented by reducing the priority of verifying new validity periods in excess of five or six per day.

A side effect caused by validity period, which can bring about a flood of advertisements of different autonomous systems because of synchronized period, should be considered carefully. This flood of advertisements is not launched by attackers. It emerges owing to similar periods and similar start time. For example, if most of validity periods are 2 days and most of start time is UTC (Universal Time Coordinated) 00:00, then in most of UTC 00:00 a flood of advertisements of different ASes probably emerges.

To overcome this side effect, random and uniform boundaries between validity period are chosen by proposed scheme. These boundaries are one-way hash values of different prefix number. These hash values are regarded as offset from some well-known time such as UTC 00:00. For instance, the start time of first validity period of prefix "128.25.128.128/16", whose hash value is "e6f9d..." and is regarded as 7 hours 36 minutes 52 seconds, is UTC 07:36:52.

The hierarchical issuing of key pairs including weak blind signature and validity period is as Algorithm 4.

In the step 8 of Algorithm 4, signing message includes validity period which is denoted as VP_i. Validity periods are treated as parts of signing messages, so as to prevent attackers from replay attackers or tampering validity periods.

Once the validity period expires, all of verifiers will check it out and thus reject this update message, which can be seen in step 3 and step 4 of Algorithm 5.

Public/private key pair is issued to the owner of prefix$_i$, where public key is value of a function of pk_{i-1} and r_i, and private key is the signature $s_{i.}$ This function, which is used to compute public key, will be derived and explained in Subsection 2.4.

(3) DSA-based validation of key pairs: Public/private key pair, which is issued to the owner of prefix $_{\text{i}}$ by Algorithm 2, should be validated by the owner of prefix $_{\text{i}}$. The algorithm to validate public/private key pairs is as Algorithm 5.

The step 3 of Algorithm 5 is used to check whether validity period expires or not. This validity period cannot be tampered because tampered validity period cannot pass through the verification of step 4 7 of Algorithm 5.

In step 5 of Algorithm 5, the "$v_i = (r_i - 1)^{u1i}\gamma pk_{i-1}^{u2i}$ mod p" substitutes for "$v_i = (g)^{u1i}\gamma pk_{i-1}^{u2i}$ mod p" in DSA. The reason why the base g is replaced with the r_{i-1} will be explained in Subsection 2.4.

Algorithm 4: Secure and hierarchical issuing of key pairs which can prevent replay attacks

```
1. The owner of prefix$_i$ send $r_1$, $r_2$,..., $r_{i-1}$, $VP_1$, $VP_2$,..., $VP_{i-1}$ to the owner of prefix $_{\mathrm{i}}$
2. The owner of prefix $_{\mathrm{i}}$ select a random secret integer $\tilde{k}_i$, $0 < \tilde{k}_i < q$
3. Compute $\tilde{r}_i = (r_{i-1}^{\tilde{k}_i}$ mod $p)$
4. The owner of prefix$_i$ send $\tilde{r}_i$ to the owner of prefix$_{i-1}$
5. The owner of prefix$_{i-1}$ select a random secret integer $k_i$, $0 < k_i < q$
6. Compute $r_i = (\tilde{r}_i\gamma r_{i-1}^{k_i}$ mod $p)$, $r_i' = r_i$ mod $q$
7. Compute $k_i^{-1}$ mod $q$
8. Compute $\tilde{s}_i = k_i^{-1}(h(\text{prefix}_i\,\#\,//\,VP_i) + \mathrm{r_i}'\cdot \mathrm{s_{i\text{-}1}})$ mod $q$
   ($h$() is a hash function h: {0,1}*→ $Z_q$, $VP$ denotes validity period in current signature, whose form is as "$T_B$～$T_E$" where $T_B$ denotes start day and $T_E$ denotes end day.)
9. The signature for prefix$_i$ # signed by the owner of prefix$_{i-1}$ is the pair ($r_i$, $\tilde{s}_i$)
10. The owner of prefix $_{\mathrm{i\text{-}1}}$ send the pair ($\mathrm{r_i}$, $\tilde{s}_i$) and $r_1$, $r_2$, ..., , $r_{i-1}$, $VP_1$, $VP_2$,..., $VP_{i-1}$, $VP_i$ to the owner of prefix$_i$
11. The owner of prefix$_{i-1}$ compute $s_i = (\tilde{k}_i)^{-1}\cdot\tilde{s}_i$ mod $q$
12. $s_i$ acts as the private key of prefix$_i$
```

Algorithm 5: Validation of key pairs

```
1. Obtain authentic public key corresponding prefix$_{i-1}$, which is $pk_{i-1}$
2. Check that $0 < r_i <$ p and $0 < s_i < q$; if not, then reject the issued key pair.
3. Check that current time is in $VP_i$; if not, then reject the issued key pair.
4. Compute $w_i = s_i^{-1}$ mod $q$, $h$(prefix$_i$ # // $VP_i$) and $r'_i = r_i$ mod $q$
5. Compute $u_{1i} = w_i\cdot h$(prefix$_i$ # // $VP_i$) mod $q$ and $u_{2i} = r'_i\cdot w_i$ mod $q$
6. Compute $v_i = (r_{i-1})^{u1i}\cdot pk_{i-1}^{u2i}$ mod $p$
7. Accept the key pair ($r_i$, $s_i$) if and only if $v_i = r_i$
```

2.3 Address attestations creating and distributing

The private key, which is denoted by s_i, is issued by the owner of prefix$_{i-1}$ according to Algorithm 2. An address attestation of prefix$_i$ is created by the owner of prefix$_i$ by using private key s_i, the algorithm to do which is as follow.

Algorithm 6: Creating an address attestation of prefix$_i$

```
1. Select a random secret integer $k$, $0 < k < q$
2. Select an origin autonomous system of prefix$_i$, whose number is denoted by AS$_0$#
3. Compute r = ($g^k$ mod p) mod $q$
4. Compute $k^{-1}$ mod q
5. Compute $s = k^{-1}\cdot$ ($h$(AS$_0$#) + $r\cdot s_i$) mod $q$
6. The signature for AS$_0$ signed by the owner of prefix$_i$ is the pair ($r$, $s$)
7. The owner of prefix$_i$ sends the pair ($r$, $s$) and $r_1$, $r_2$,..., $r_{i-1}$ to the origin autonomous system of prefix$_i$
```

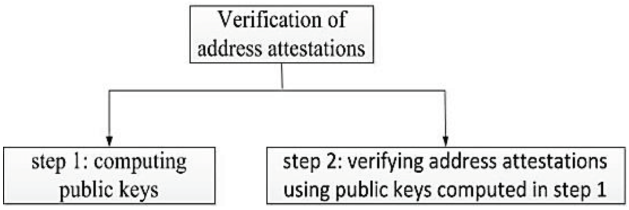
Verification of address attestations
step 1: computing public keys
step 2: verifying address attestations using public keys computed in step 1

Figure 8 Steps of verification of address attestations

Address attestation will be sent to the origin AS of prefix$_i$ after creating by Algorithm 6. This address attestation can be used to authenticate the origin AS of advertised prefix by any other ASes in the AS PATH of update messages for advertised prefix (note that origin AS can be denoted by AS$_0$). The authentication of origin AS is actually the verification of an address attestation which is a DSA signature.

Address attestations should be distributed to all of the ASes in the AS_PATH attributes of update messages for prefix$_\mathrm{i}$ before verifying. As to what is described in subsection 2.1, in our scheme, creators of address attestations do not upload address attestations to repositories anymore. These address attestations are put into update messages after verifying them. Address attestations can be drawn out directly from update messages, and be verified immediately. Address attestations are regarded as components of update messages in proposed scheme.

2.4 Verification of address attestations

Because address attestations are components of update messages in proposed scheme, they can be taken by receivers directly from update messages. After being taken from update messages, address attestations, which are DSA-based signatures, can be verified by veri.ers.

In RPKI or S-BGP, for verification of an address attestation, two steps have to been executed by a verifier. One is to verify address attestation using public key corresponding to advertised prefixes, where a signature is verified. The other is to authenticate public key corresponding to advertised prefixes, where several signatures exist in the chain from IANA to the owner of advertised prefix have to be verified separately.

In our scheme, there are also two steps to be executed for verifying an address attestation by a veri.er (see Fig. 8). The first step is to compute public key, which corresponds to advertised prefix, by using a formula and information drawn directly from current update message. The second step is to verify current address attestation, which is a DSA-based signature, by using this public key computed in the first step.

We firstly describe how to verify an address attestation using a public key gotten by computing, and then explain how to compute a public key according to information within an update message.

(1) Verifying an address attestation using an existing public key: The Algorithm 7 describes how to verify an address attestation using a public key gotten by computing.

Algorithm 7: Validation of address attestations

```
1. Computing authentic public key corresponding prefix$_i$, which is $pk_i$
2. Verify that $0 < r < p$ and $0 < s < q$; if not, then reject the issued key pair.
3. Compute $w = s^{-1} \bmod q$, $h(\mathrm{AS}_0\#)$
4. Compute $u_1 = w \cdot h(\mathrm{AS}_0\#) \bmod q$
   and $u_2 = r \cdot w \bmod q$.
5. Compute $v = (r_i)^{u_1} \cdot (pk_i)^{u_2} \bmod p$.
6. Accept the key pair $(r, s)$ if and only if $v = r$
```

In DSA, for an entity, the function mapping a private key to corresponding public key is that $pk = g^s \bmod p$, where pk denotes public key, s denotes private key, g and p are parameters of the current DSA system in the initiation phase.

In Algorithm 7, the step 1 is to compute current public key. Next, we will elaborate how to compute a public key according to information drawn from an update message.

(2) Computing a public key according to information within an update message: For reducing the burden of computing, we transform $pk = g^s \bmod p$ (which is function mapping a private key to corresponding public key) into $pk_i = (r_i)^{s_i}$ mod p, where r_i has been described in Algorithm 2, pk_i/s_i is public/private key pair corresponding to prefix$_\mathrm{i}$. As function $pk = g^s \bmod p$, the function $pk_i = (r_i)^{s_i}$ mod p is also a one way function whose security is based on the intractability of the discrete logarithm problem, and the private key cannot be extracted from the corresponding public key. In a similar way, for the owner of prefix$_{\mathrm{i-1}}$, whose function should be $pk_{i-1} = (r_{i-1})^{s_{i-1}}$ mod p. The algorithm for signing a message is the same as the DSA except that the r_i = (g^{k_i} mod p) mod q is replaced with the r_i = ($(r_{i-1})^{k_i}$ mod p) mod q which has been described in the Step 2 of Algorithm 2. For verifiers of the signatures signed by the owner of prefix$_i$, the verifying algorithm is almost the same as that of DSA except that the base g is replaced by the base r_{i-1}, which has been described in the Step 5 of Algorithm 5.

By analyzing the Algorithm 5, we find that this DSA-based verifying algorithm for validation of key pairs is actually to check the equation

$$(r_i)^{s_i} = (r_{i-1})^{h(\mathrm{prefix}_i\#)} \cdot (pk_{i-1})^{(r_i \bmod q)} \bmod p \qquad (1)$$

If this equation holds, then signature (r_i, s_i) is validated. Otherwise, signature (r_i, s_i) is false. Because $pk_i = (r_i)^{s_i}$, we can replace $(r_i)^{s_i}$ with pk_i, and thus

$$pk_i = (r_{i-1})^{h(\mathrm{prefix}_i\#)} \cdot (pk_{i-1})^{(r_i \bmod q)} \bmod p \qquad (2)$$

The formula (2) can be applied recursively as the following.

$$pk_i = (r_{i-1})^{h(\mathrm{prefix}_i\#)} \cdot (pk_{i-1})^{(r_i \bmod q)} \bmod p$$

$$pk_{i-1} = (r_{i-2})^{h(\mathrm{prefix}_{i-1}\#)} \cdot (pk_{i-2})^{(r_{i-1} \bmod q)} \bmod p$$

$$pk_{i-2} = (r_{i-3})^{h(\mathrm{prefix}_{i-2}\#)} \cdot (pk_{i-3})^{(r_{i-2} \bmod q)} \bmod p$$

......

$$pk_1 = (g)^{h(\mathrm{prefix}_1\#)} \cdot (pk_0)^{(r_1 \bmod q)} \bmod p$$

Thus,

$$pk_i = (r_{i-1})^{h(\mathrm{prefix}_i\#)} \cdot (pk_{i-1})^{(r_i \bmod q)} \bmod p =$$

$$= (r_{i-1})^{h(\mathrm{prefix}_i\#)} \cdot ((r_{i-2})^{h(\mathrm{prefix}_{i-1}\#)} \cdot (pk_{i-2})^{(r_{i-1} \bmod q)}$$

$$\bmod p)^{(r_i \bmod q)} \bmod p =$$

$$= (r_{i-1})^{h(\mathrm{prefix}_i\#)} \cdot ((r_{i-2})^{h(\mathrm{prefix}_{i-1}\#)} \cdot ((r_{i-3})^{h(\mathrm{prefix}_{i-2}\#)} \cdot$$

$$\cdot (pk_{i-3})^{(r_{i-2} \bmod q)} \bmod p)^{(r_{i-1} \bmod q)} \bmod p)^{(r_i \bmod q)} \bmod p$$

......

Therefore, we can achieve the following equation (note that r_0 is just the g).

$$pk_i = \prod_{j=1}^{i} r_{j-1}^{(h(prefix_j\#) \cdot \prod_{k=j}^{i} r_k) \bmod q}$$

$$pk_0^{(\prod_{j=1}^{i} r_j) \bmod q} \bmod p$$

Above formula has not thought of the factor of validity period. If we consider the factor of validity period according to Algorithm 4, then the following equation can be achieved.

$$pk_i = \prod_{j=1}^{i} r_{j-1}^{(h(prefix_j\#\|VP_j) \cdot \prod_{k=j}^{i} r_k) \bmod q}$$

$$pk_0^{(\prod_{j=1}^{i} r_j) \bmod q} \bmod p \qquad (3)$$

From the derivation process of formula (3), we can see that the private key of current prefix is just the second half of DSA signature of current prefix# signed by the private key of upstream prefix. Every substitution of iteration is just the right part of formula (1), and thus can be regarded as a validation of corresponding DSA signature under the definition that $pk_j = (r_j)^{s_j}$, where j = 0, 1,..., i.

Any AS in AS PATH of update messages can use public key of advertised prefix to verify corresponding address attestations, the algorithm to do which is Algorithm 7. By using formula (3), any AS can compute the public key of advertised prefix. If verification of an address attestation succeeds by using computed public key, then we can draw a conclusion that the origin AS is authenticated and corresponding public key is issued correctly, which denotes that we can rule out the possibility that IP prefix hijacking exists in process of broadcast of current update message.

3 Theoretical basis

There are two theoretical bases which include DSA and existing authentication of origin AS, on which the proposed scheme is based.

3.1 DSA

DSA lays cryptographic foundation of the proposed scheme, which is a United States Federal Government standard for digital signatures. The standard was expanded in 2009 as FIPS 186-3 [26].

The DSA consists of three algorithms including key generation, DSA signature generation, and DSA signature verification. Details of these three algorithms can be seen as following [26].

Key generation for DSA

```
1. Select a prime number q which is 160 bits.
2. Choose t so that $0 \leq t \leq 8$, and select a prime number p where $2^{511} + 64t < p < 2^{512} + 64t$, with the property that $q$ divides $(p-1)$.
3. Select a generator g, whose order is q, of the unique cyclic group $Z_p^*$
    3.1 Select an element $a$ in $Z_p$ and compute $g = a^{(p-1)/q} \bmod p$.
    3.2 If g =1 then go to step 3.1.
4. Select a random integer $s_0$ such that $1 \leq s_0 \leq (q-1)$.
5. Compute $PK_0 = g^{s0} \bmod p$.
6. IANA's public key is $(p, q, g, PK_0)$; IANA's private key is $s_0$.
```

DSA signature generation

```
1. Select a random secret integer k, $0 < k < q$
2. Compute r = $(g^{k_i} \bmod p) \bmod q$
3. Compute $k^{-1} \bmod q$
4. Compute $sig = k^{-1}(h(m) + r \cdot s) \bmod q$, where s is the private key of signer.
   (the signed message is $m$, $h()$ is a hash function $h$: $\{0,1\}^* \ldots \rightarrow Z_q$)
5. The signature for $m$ is the pair $(r, sig)$
```

DSA signature verification

```
1. Obtain authentic public key, which is $pk$
2. Verify that $0 < r < p$ and $0 < sig < q$; if not, then reject the signature.
3. Compute $w = sig^{-1} \bmod q$, $h(m)$
4. Compute $u_1 = w \cdot h(m) \bmod q$ and $u_2 = r \cdot w \bmod q$
5. Compute $v = g^{u1} \cdot pk^{u2} \bmod p$
6. Accept the key pair $(r, sig)$ if and only if $v = r$
```

3.2 Existing authentication of origin AS

Certificates are necessary for authenticating public keys and binding IP prefixes to these public keys belonging to the organization to which the IP prefixes are assigned [7], in existing cryptography-based schemes for preventing IP prefix hijacking such as some typical schemes of which are S-BGP [8], SoBGP [9], psBGP [10], OA [11], SPV [13], HCBGP [12], and so on. Each certificate contains a private extension that specifies the set of address blocks that have been allocated to the organization. The initial deployment of the Resource Public Key Infrastructure (RPKI) [7] taken up by the IETF SIDR working group [15] shows this kind of methods is to some extent accepted. For authenticating origin ASes, this kind of methods mainly uses X.509 digital certificates and RSA [7].

An X.509 certificate is used to bind a public key to an organization and to a set of prefixes. There are two binds in a certificate. One is to bind the DNS name of an organization and a list of prefix(es) owned by this organization, the other is to bind this DNS name and its public key. This shows the DNS name is the bridge between public key and the list of prefix(es). If there is no certificate in this architecture, the bind of public key and the list of prefix(es) will be lost because the DNS name no longer exists [14].

However, in proposed scheme, it is not necessary to carry certificates in update messages when address attestations are distributed in update messages. An earlier variant of our work, previously published in [14], provided similar guarantees but cannot maintain compatibility with the DSA and its analytical tools. Maintaining compatibility with the DSA and its analytical tools, which is sophisticated, is not only the key point of this paper but also very important to our method because DSA is a United States Federal Government standard for digital signatures and thus this method becomes easier to practical application.

IANA assigns IP address blocks (or prefixes) to organizations, which in turn assign smaller prefixes to service organizations. These service organizations often assign these blocks to their customers. At each step in the delegation, the recipient organization of the prefix generates an asymmetric private key to represent the organization [8]. The prefix issuer uses its private key to sign the public key of the recipient organization, together with a list of prefixes which are delegated to the organization, forming the public key certificate of the recipient organization, or simply certificate. The organization that owns one or more prefix(es) thus has a certificate signed by the issuer of the prefix(es).

There is one signature in each certificate. Signatures which are in certificates have to be verified separately, which causes heavy overhead of public-key-related information and on-line computation.

4 Results

In our DSA-based scheme described in Section II, a verifier can compute public key corresponding advertised prefix only according to information contained in update messages. The information includes the address attestation and all of $r_j (j = 1, 2, ..., i)$ relaid by all issuers in hierarchical issuing of key pairs. Using the formula (3), the public key of advertised prefix can be achieved and used to verify the address attestations. If this verification fails, update messages will be rejected. Otherwise, the success of this verification shows that an address attestation is signed by the private key which is issued to the owner of advertised prefix by the hierarchical issuing process described as Algorithm2 and thus the corresponding public key is authenticated. Therefore, for authenticating the public key, it is not necessary for a verifier to check a series of certificates existing in certification path from the owner of advertised prefix to the IANA. Thus the authentication of origin ASes is efficient.

Moreover, in our scheme, for defending against IP prefix hijacking, computing public key of owner of prefix using formula (3) is main burden as well as using this public key to verify the address attestation which is a DSA signature. Formula (3) is a multi-exponentiation. Using some trick algorithms [28], separated verifying signatures in public key certificates, which is inevitable in almost all existing cryptography-based schemes (such as S-BGP), can be replaced with computing of this multi-exponentiation formula. Thus, the efficiency is achieved.

5 Comparison

In this section, we compare proposed scheme with S-BGP according to size of related information, and convergence time.

5.1 Size of related information

According to size of related information for origin authentication, we compare the proposed scheme with the S-BGP [8].

In S-BGP, related information includes 5 public key certificates in a single issuing chain on average. Related information also includes an address attestation which is about 128+20 bytes. In proposed scheme, related information includes r_j (j=1,..., 5) where each r_j is about 128 bytes long, as well as an address attestation.

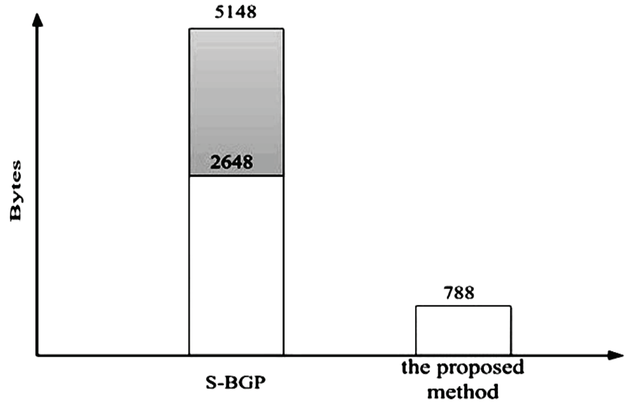

Figure 9 A comparison between sizes of public-key-related of S-BGP and proposed scheme

Size of related information in S-BGP:
Size1≈ (500 ~ 1000)×5+128+20 = 2648 ~ 5148 bytes
Size of related information in proposed scheme:
Size2≈128×5+128+20 = 788 bytes

By comparing Size1 with Size2 (see Fig. 9) we can see that in S-BGP, because update messages are limited in length to 4096 bytes and thus are too small to carry the necessary related information; however, in the proposed method, the limitation of 4096 bytes cannot prevent the related information from being part of update messages, and thus a verifier can authenticate address attestations and public keys only according to the information within update messages.

5.2 Convergence time

The SSF Net (Scalable Simulation Framework Network models) simulator is used by us to compare the impact that update processing under S-BGP and proposed method might have on convergence time. The most of default values of options for SSFNet configuring are used by us, such as MRAI (Minimum Route Advertisement Interval) = 30 s. The key options for SSFNet configuring in our experiments are the proc-delay model and proc-time (includes min-proc-time and max-proc-time, and we set them equal).

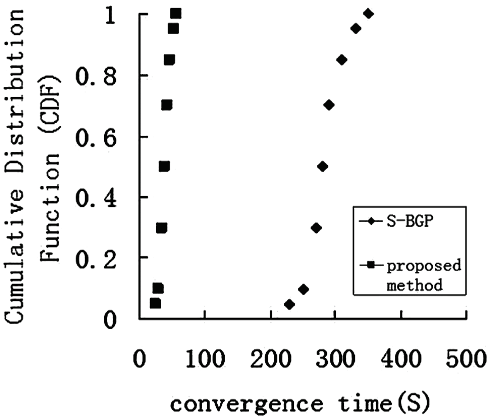

Figure 10 A comparison between convergence times of S-BGP and proposed scheme

Fig. 10 is about the convergence time comparing S-BGP with proposed method.

6 Conclusions and future work

In this paper, for improving efficiency of authentication of origin AS, we present a DSA-based scheme for preventing IP prefix hijacking. By taking full advantage of specific characteristics of DSA, proposed scheme can at least enjoy the two following advantages: (1) Owing to extensive body of experience and literature associated with the DSA which is a United States Federal Government standard for digital signatures, proposed scheme is easier to be widely accepted and put into practice. (2) Public key certificates, which are used to authenticate public keys, are not necessary. In most existing cryptography-based schemes, for authenticating public keys, public key certificates are necessary where signatures have to be verified separately, which causes heavy overhead of public-key-related information and on-line computation. In proposed scheme, public keys used to verifying origin attestations can be directly computed by using a multi-exponentiation formula, where correctness of public keys is up to corresponding verifying of origin attestations. Therefore, the proposed scheme is efficient because separated verifying of signatures of certificates can be replaced with computing of a formula.

Future work will cover the extension of the proposed scheme from defending against IP prefix hijacking, to the defence for AS PATH tampering which is also DSA-based and efficient. For further improving the efficiency of authentication of origin AS, based on methods presented in this paper, some other tricks can be integrated in our scheme. For example, signature amortization (where one message can be sent to all the peers, and only one new signature is involved [2]) can be used to relieve the burden of computation of creating signatures and thus improve efficiency.

Acknowledgment

Authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper. This work was supported in part by the National Nature Science Foundation of China (No. 61562036, 61462033, 61262010), Natural Science Foundation of Jiangxi (No. 20142BAB207017, 20142BAB207015, 20132BAB201036), Foundation of Jiangxi Advanced Research Center of Ecommerce Engineering (No. 201312), China Postdoctoral Science Foundation (No. 2014M551855).

7 References

[1] Bates, T.; Smith, P.; Huston, G. The cidr report. URL: http://www.cidr-report.org/as2.0 (Accessed on May, 2014).

[2] Butler, K.; Farley, T. R.; McDaniel, P.; Rexford, J. A survey of BGP security issues and solutions. // Proceedings of the IEEE. 98, 1(2010), pp. 100-122. DOI: 10.1109/JPROC.2009.2034031

[3] Zhang, Z.; Zhang, Y.; Hu, Y. C.; Mao, Z. M.; Bush, R. ISPY: detecting IP prefix hijacking on my own. // IEEE/ACM Transactions on Networking. 18, 6(2010), pp. 1815-1828. DOI: 10.1109/TNET.2010.2066284

[4] Bono, V. J. 7007 explanation and apology. URL: http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html (Accessed on May, 2014).

[5] Huston, G. Youtube IP hijacking. URL: http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study (Accessed on Sept., 2012).

[6] Cowie, J. Rensys blog: China's 18-minute mystery.URL: http://www.renesys.com/blog/2010/11/chinas-18-minute-mystery.shtml (Accessed on September, 2012)

[7] Austein, R.; Huston, G.; Kent, S.; Lepinski, M. Manifests for the resource public key infrastructure. URL: http://datatracker.ietf.org/wg/sidr/documents/ (Accessed on May, 2014).

[8] Kent, S.; Lynn, C.; Seo, K. Secure border gateway protocol (S-BGP). // IEEE Journal on Selected Areas in Communications. 18, 4(2000), pp. 582-592. DOI: 10.1109/49.839934

[9] White, R. Securing BGP through secure origin BGP (soBGP). // The Internet Protocol Journal. 6, 3(2003), pp. 161-172.

[10] Van Oorschot, C. P.; Wan, T.; Kranakis, E. On interdomain routing security and pretty secure BGP (psBGP). // ACM Transactions on Information and System Security (TISSEC). 10, 3(2007), pp. 1-41. DOI: 10.1145/1266977.1266980

[11] McDaniel, P.; Aiello, W.; Butler, K.; Ioannidis, J. Origin authentication in interdomain routing. // Computer Networks. 50, 16(2006), pp. 2953-2980. DOI: 10.1016/j.comnet.2005.11.007

[12] Zhang, Y.; Zhang, Z.; Mao, Z. M.; Hu, Y. C. HC-BGP: Alight-weight and flexible scheme for securing prefix ownership. // Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks / Lisbon, 2009, pp. 23-32.

[13] Hu, Y.-C.; Perrig, A.; Sirbu, M. SPV: Secure path vector routing for securing BGP. // Proceedings of the ACMSIGCOMM conference on Data communication / Oregon, 2004, pp. 179-192. DOI: 10.1145/1015467.1015488

[14] Le, Z.; Xiong, N.; Yang, B.; Zhou, Y. SC-OA: A Secure and Efficient Scheme for Origin Authentication of Interdomain Routing in Cloud Computing Networks. // Proceedings of the 25th IEEE International Parallel and Distributed Processing Symposium / Anchorage, 2011, pp. 243-254. DOI: 10.1109/ipdps.2011.32

[15] Huston, G.; Rossi, M.; Armitage, G. Securing BGP- A Literature Survey. // IEEE Communications Surveys Tutorials. 13, 2(2011), pp. 199-222. DOI: 10.1109/SURV.2011.041010.00041

[16] Karlin, J.; Forrest, S.; Rexford, J. Autonomous security or autonomous systems. // Computer Networks. 52, 15(2008), pp. 2908-2923. DOI: 10.1016/j.comnet.2008.06.012

[17] Rexford, J.; Feigenbaum, J. Incrementally-Deployable Security for Interdomain Routing. // Proceedings of Cybersecurity Applications and Technology Conference for Homeland Security / Anchorage, 2009, pp. 130-134. DOI: 10.1109/catch.2009.35

[18] Lad, M.; Massey, D.; Pei, D.; Wu, Y.; Zhang, B.; Zhang, L. PHAS: A prefix hijacks alert system. // Proceedings of USENIX Security Symposium, Boston, 2006, pp. 176-187.

[19] Zheng, C.; Ji, L.; Pei, D.; Wang, J.; Francis, P. A light-weight distributed scheme for detecting IP prefix hijacks in real-time. // Proceedings of the ACM SIGCOMM Computer Communication Review. 37, 4(2007), pp. 277-288. DOI: 10.1145/1282380.1282412

[20] Qiu, T.; Ji, L.; Pei, D.; Wang, J.; Xu, J. Tower Defense: Deployment strategies for battling against IP prefix hijacking. // Proceedings of International Conference on Network Protocols, Kyoto, 2010, pp. 134-143. DOI: 10.1109/ICNP.2010.5762762

[21] Patelan, H. B.; Patel, D. R. Performance analysis of BGP security proposals. // International Journal of Recent Trends in Engineering. 2, 2(2009), pp. 187-189.

[22] Qiu, T.; Ji, L.; Pei, D.; Wang, J.; Xu, J.; Ballani, H. Locating prefix hijackers using lock. // Proceedings of USENIX Security Symposium / San Diego, 2009, pp. 135-150.

[23] Qiu, J.; Gao, L.; Ranjan, S.; Nucci, A. Detecting Bogus BGP Route Information: Going Beyond Prefix Hijacking. // Proceedings of Security and Privacy in Communication Networks / Nice, 2007, pp. 381-390.

[24] Goldberg, S.; Schapira, M.; Hummon, P.; Rexford, J. How Secure are Secure Interdomain Routing Protocols? // Proceedings of the ACM SIGCOMM Conference on Data Communication / New Delhi, 2010, pp. 87-98. DOI: 10.1145/1851182.1851195

[25] Miller, B.; Rupp, A. Faster Multi-exponentiation through Caching: Accelerating (EC) DSA Signature Verification. // Proceedings of International Conference of Security and Cryptography for Networks / Amalfi, 2008, pp. 39-56. DOI: 10.1007/978-3-540-85855-3_4

[26] Furlani, C. FIPS-186-3, the third and current revision to the official DSA specification. URL: http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf (Accessed on May, 2014).

[27] Cooper, D.; Heilman, E.; Brogle, K.; Reyzin, L.; Goldberg, S. On the Risk of Misbehaving RPKI Authorities. // Proceedings of HotNets-XII / CollegePark, MD, 2013, pp.

36-41.

[28] Girault, M. Self-certified public keys. // Proceedings of International Conference on the Theory and Application of Cryptographic Techniques / Berlin, 1991, pp. 490-497. DOI: 10.1007/3-540-46416-6_42

Author's address

Bo Yang, Associate Professor
School of Information Technology, Jiangxi University of Finance
and Economics,
High Level Engineering Research Center of Electronic-Commerce,
Jiangxi Provincial Colleges and Universities,
330013 Nanchang, China,
E-mail: jxncyangbo2002@163.com