SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •
Special Focus on Future Internet Architecture and Protocol

December 2015, Vol. 58 120102:1–120102:14
doi: 10.1007/s11432-015-5461-0

Building an IPv6 address generation and traceback system with NIDTGA in Address Driven Network

LIU Ying[1,3], REN Gang[1,3]*, WU JianPing[1,2,3], ZHANG ShengLin[1,2,3], HE Lin[1,2,3] & JIA YiHao[1,2,3]

[1]*Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing* 100084*, China;*
[2]*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084*, China;*
[3]*Tsinghua National Laboratory for Information Science and Technology, Beijing* 100084*, China*

Received September 24, 2015; accepted October 28, 2015; published online November 12, 2015

Abstract In the design and construction process of Next Generation Internet, it is important to identify the source of each IP packet forwarding accurately, especially for the support of precise fine-grained management, control, traceability and improving the trustworthiness of the Internet. This paper designed a scalable Network Identity (NID) scheme for the Internet users, proposed NIDTGA (Network Identity and Time Generated Address), an IPv6 address generation algorithm embedded NID and time information, then designed and implemented an IPv6 address generation and traceback system based on NIDTGA. The design of NIDTGA, which reflects the length, time and owner attributes of the IP address, can be a good support to ADN (Address Driven Network). At the same time, by embedding the key elements of user identity and time in the IPv6 address, and by taking into account both the traceability and privacy, NIDTGA can provide a technical basis for the establishment of the network trust mechanism, and achieve the traceability of security event.

Keywords network identity, IPv6, Address Driven Network, IP traceback, IP address generation

Citation Liu Y, Ren G, Wu J P, et al. Building an IPv6 address generation and traceback system with NIDTGA in Address Driven Network. Sci China Inf Sci, 2015, 58: 120102(14), doi: 10.1007/s11432-015-5461-0

1 Introduction

1.1 Challenges to the Internet and the Address Driven Network

With an increasing trend of human beings' reliance on the Internet, the evolution of the Internet is confronted with a multitude of challenges involving scalability, security, quality of service, traffic engineering and mobility.

Scalability: It is scalability that shapes and makes the Internet the most successful technology, but because of the limitation of IPv4 address, IPv4 technology cannot meet the demands of the new scenarios as the development of IOT (Internet of Things) or other special applications. Actions should be taken immediately because the IPv4 addresses have completely exhausted.

Security: Security is almost passive as an isolated branch of Internet science without any systemic or architectural support. Another vital challenge in the research of the Next Generation Internet is to

* Corresponding author (email: rengang@cernet.edu.cn)

thoroughly settle the security issue at the architecture level, which means that any attacks on the Internet will be strongly traceable by binding network identities for each user with their authentic IP addresses.

Quality of Service (QoS): Based on the TCP/IP architecture, the Internet, whose staple principle is to guarantee the reachability of the packet, serves data on it with "Best effort". Nevertheless, there is no pledge of the efficiency in the bandwidth utility with a shortest-path address strategy.

Mobility: The most severe challenge in mobility, at present, is the communication discontinuity due to the IP address switching during the movement. Though the users' identities remain the same, IP address will still be switched whenever a location shift occurs at the movement. Conspicuously, the semantic overloading should assume the prime responsibility of this issue.

Through the above analysis, it is the disordered understanding of the semantics and grammar of IP address other than TCP/IP itself that shapes the underlying dilemmas of the Internet. Based on this, ADN (Address Driven Network), which is proposed by Tsinghua University at 2011, advocates regarding IP address as the core of the innovation in solving the issues widespread in the Internet. Thanks to the huge space of IPv6 address, under the ADN system, the regulation of network layers can be enhanced to a tremendous extent without any modification of the current Internet architecture. Kernels of ADN involves:

(1) Separating IP address into identity label and location label.
(2) The warranty of the authentication of source IP address.
(3) "Planar" routing forwarding based on the source and destination address.
(4) Dynamic switching of IP address.

1.2 IP address and identity

The topology of the Internet is to transit from static to dynamic along with the birth of multi-homing and other mobile applications, leading to a multitude of schemes that to separate IP address space into an identifier and a locator. Just as HIP (Host Identity Protocol) and Shim6 protocol, in general, hosts or nodes in the network topology should be allocated with a unique identity which remains the core of routing with a special mapping mechanism into a locator. The schedule of IP address is almost the same under ADN architecture: each IPv6 address will be detached into identity IP space and location IP space, which means that the 64 bits prefix IP address assigning for each node will change along with any shift of its location, while 64 bits interface IP address will remain fixed as the same identity for each node.

The New ARCH project [1] has made an explicit statement: identity validation system, including the IP address authentication, is a milestone of the security in the Next Generation Internet. From the architectural and algorithm perspective, the accurate validation of the source of each IP packet in forwarding will become a key point in supporting and promoting the fine-grained regulation and traceback with high credibility during the construction of Next Generation Internet.

To achieve the goal of traceability, the third-party authority must be involved to manipulate the whole mapping mechanism. Consequently, how to generate the self-verification network identity without the third-party authority remains a hot issue in IP traceback. Under normal circumstances, an ordinary method adopted widely is to bind the IP address with a public key, so it can be valid without any global credible authority by signature with a private key. Fortunately, the huge space of 128-bit IPv6 address turns to be the threshold for embedding the identity information with an appropriate format.

In the field of international Internet standardization and research, several solutions have been proposed to settle such issue by utilizing the last 64-bit interface IP address as the container of users' identities, taking the CGA (Cryptographically Generated Addresses) [2] as the most representative one.

The main 4-tuple information must be included in validating the source of the IP packet: source IP address, device identity (MAC address), network identity for each user and the time of the source address generated. However, the fatal drawback for CGA is its scarcity of the recording of network identity, which is, as the supplement of IP address, the most crucial factor in tracing.

Once the network identity could be embedded in IPv6 address with a sophisticated pattern, users' identities are able to be obtained at the same time during the traceback. To complete this, there are two schemes. The first one is to encrypt the users' private identical attributes directly, such as SSN (Social

security number), as the holistic interface IP address with a sophisticated algorithm. Nonetheless, the privacy of individuals is to readily divulge due to the fixed format. According to the defect, we devised the NID (Network Identity), with a universal pattern, to identify the user as the only entrance to access the Internet. By encrypting the NID and the time accessing the Internet, security will be pledged for the strengthening of privacy protection.

1.3 Contribution

This paper designed a scalable Network Identity (NID) scheme for the Internet users, proposed NIDTGA, an IPv6 address generation algorithm embedded NID and time information, and then designed and implemented an IPv6 address generation and traceback system based on NIDTGA. The design of NIDTGA, which reflects the length, time and owner attributes of the IP address, can be a good support to ADN. At the same time, by embedding the key elements of user identity and time in the IPv6 address, and by taking into account both the traceability and privacy, NIDTGA provided a technical basis for the establishment of the network trust mechanism, and achieve the traceability of security event.

2 Related work

The main focuses of NIDTGA are Internet user identification, authentication, traceback and IPv6 address allocation.

Researches related to network identification and address allocation have already become the hot topics accompanied by plenty of research papers and IETF (Internet Engineering Task Force) standards.

HIP [3] and Shim6 [4] detached identity label from location label by altering the hierarchical structure of the protocol family. HIP de-coupled the transport layer and network layer via the import of the host layer into it. An IPv6 address generated by HIP protocol consists of three parts, a prefix allocated by IANA to distinguish the IPv6 address from other kinds of addresses, a 4-bit algorithm code indicating the hash algorithm, and a 96-bit hashed string, the original string of which includes the host identity, domain identifier, and so on. Shim6 brought the Shim layer, which is between the IP end sub layer and IP routing sub layer, to carry the same point.

In GSE [5] and LIN6 [6], hosts and routers were isolated with the external topology by the utilization of a part of IPv6 address. GSE proposed to divide the IPv6 address into 3 portions, and multi-homing will be realized by using the supreme 16-M-N bit (the RG part), while the last 64-bit was denoted for identity to implement separation in LIN6.

CGA, which owned a 128-bit format, generated the last 64-bit based on a series of hash functions. To establish the authenticity validation in CGA, senders need to put a signature with a private key for receivers' verification.

AIP [7], which was a hierarchical self-validation structure proposed by David.G. Andersen et al., initially suggested using accountability at the network layer to thoroughly tackle the issue of management of address security. It formatted the address as AD1:AD2EID to establish the integrated system, while the AD and EID are the hash value of the network area and the host public key respectively.

Overall, HIP, Shim6, GSE and LIN6 try to separate the identification and location of IPv6 address with different manners, while CGA and AIP find a way to guarantee that the source addresses are verified. However, none of them achieves network user identification, authorization, and traceback in a whole system. Accordingly, in the above schemas, it is not easy for the receiver to trace the sender's identification.

Network user identification and traceback have emerged as a hotspot in the research area over recent years. eID (electronic Identity) [8] was designed for Chinese citizens' remote identity validation on the Internet. Based on the existing management of civil identity, public security organs issued it for each individual with the cryptography assistance and the support of smart chip card. eID_code, which will not record any identity information itself, corresponded it with a unique sequence instead. However,

the network user can be authorized only with the additional hard devices issued by the authority, which makes eID not very flexible.

3 Outline of NIDTGA

This paper presents NIDTGA, an IPv6 address generation scheme based on NID and time information. This scheme is based on the premise: the network has achieved SAVA (Source Address Validation Architecture) [9], and the IPv6 source address is authentic.

SAVA makes it possible to construct a secure environment for the compulsory source address validation at the access network, intra-domain and inter-domain level. Attack traffic with forged IP address will be filtered while any packet on the Internet can be traced back highly accurately. The fine-grained regulation, at the same time, can not only account or measure the real-time situation of the network more conveniently, but also precisely position the source of each packet without the acquaintance of its locus in the topology. SAVI [10], the substructure of the SAVA, realized the source IP address validation against the corresponding attack at level of access network by the binding table including host MAC address, port number of switch and the IPv6 address. Under such circumstance (packets are going to be forwarded only with the authentic source address), users' identity can be readily captured for further utilization during the process of trace back.

Based on SAVA, where all packets on the Internet own the authentic source IP address, NIDTGA is about to solve the key issues as follows.

(1) Design a scalable structure of NID.

(2) Define the format of time embedded in the IPv6 address.

(3) Select an appropriate algorithm during the embedding.

(4) Devise the scheme to implement the whole system for the generating, allocating, management and traceback of IPv6 address with the privacy protection.

To make this paper well understood, the glossary related to the prototype of NIDTGA is interpreted as follows.

(1) DID (Division Identity): the unique identity allocated by the current organization that users belong to, including the users' account number in campus network, the user's account number in ISP, or the ID number in Citizen Identity Information System of Ministry of public security, and so on.

(2) NID (Network Identity): the core label to denote the user identity for network accessing as a 10-bit hexadecimal sequence including Division Part, Organization Part and User Part.

(3) AID (Address ID): the last 64-bit interface IP address generated by NIDTGA.

While SAVA guarantees that all of the IPv6 addresses are valid, NIDTGA helps SAVA achieve a high level security: a receiver can trace the sender's identification of any packet he received. Even in the process of packet forwarding, the network administrators can trace the sender's identification of any packet passing through some router or switch. Although NIDTGA is designed for ADN, it is also applicable for ordinary IPv6 networks.

4 Design of NID and its generation algorithm

In this section, we describe the philosophy in designing the NID, based on which the structure of the NID is discussed in detail. Particularly, emphasis is placed on the "User Part" of the NID structure.

4.1 NID design principles

The principles of NID design include: hierarchy, scalability, confidentiality, flexibility, memorability and usability.

(1) Hierarchy: The structure of NID should be clearly divided into different organizations so that organizations can be easily managed in isolation. In view of that, NID should be formed by 3 parts: Division Part, Organization Part and User Part, which could lead to the feature of strong hierarchy. The

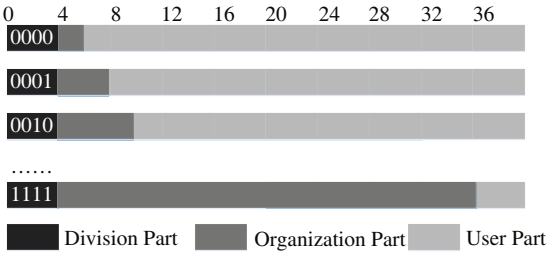

Figure 1 Structure of NID.

organization that users belong to can be readily confirmed by NID in order to establish the hierarchical regulation.

(2) Scalability: Authorization must be warranted when a legal organization or individuals want to apply a NID, and during the process of which the licensor would ensure that NIDs applied by different persons cannot be collided. When the organizations own the minimal population, the quantity of organization should reach the peak of 232, while the organizations own the maximal population, the quantity of organization should reach the valley of 22.

(3) Confidentiality: NID would not easily divulge the privacy of users' basic information. Hackers cannot easily deduce the info from NIDs if the rational encryption methods, such as SHA, are adopted.

(4) Flexibility: The Network identity that is devised in other ways can be integrated into NIDTGA program easily. The variable length of User Part sequence guaranteed that any structure of network identity whose length is less than 36 bits is feasible to be integrated.

(5) Memorability: Compared with China Citizen ID Number (18-bit) and mobile phone number in China (11-bit), a 10-bit hexadecimal sequence of NID is easy to remember for individuals.

(6) Usability: Usability pledges that NID can be applied to connect the Internet without any external bother. Without any hardware electronic devices in hand, once one attempts to connect the Internet, wherever and whenever, access should be guaranteed if he holds the validated NID and the corresponding password.

4.2 Structure of NID

The NID is devised as 40 bits including Division Part (fixed 4 bits), Organization Part (m bits) and User Part ($(36-m)$ bits) with varied bits. The graphic description is shown as Figure 1.

Division Part is in the front part of NID with a fixed length of 4 bits, changing from "0000" to "1111", which means that it contains 16 kinds of forms. The reason why we create Division Part is that we can put Organization Part and User Part in a more reasonable and logically rational way so that the rest bits of NID can be utilized more efficiently.

Organization Part, whose length depends on different values of the Division Part, is in the middle of NID, containing 16 kinds of forms to distinguish different organizations. In general, the length of Organization Part offered by licensors relies on the potentially possible maximum quantity of members that belong to the organization. Consequently, the organizations that need to register an NID license should evaluate its potentially possible maximum quantity of members to get a more reasonable Organization Part sequence.

The User Part is the end part of NID. Its length is inferable since the total length of NID is fixed 40 bits. Attention should be paid that the sequence of the User Part for different individuals must be unique because it is the only identity to identify the members in the same organization.

4.3 General description of the length in different parts

Denoting "u" as the potentially possible maximum quantity of the members for an organization, the lengths of the Organization Part, the User Part and the value of the Division Part are $35-n$, $n+1$,

Table 1 General description of NID

	Value of Division Part	Length of Organization Part	Length of User Part
Odd	$(33-n)/2$	$35-n$	$n+1$
Even	$(34-n)/2$	$36-n$	n

$(33-n)/2$ respectively when "n" is odd. Similarly, the answers are $35-n$, n, $(34-n)/2$ respectively when "n" is even, as is shown in Table 1.

Instance 1. Taking 32-bit NeID, which is transformed from the 32-byte eID published by the Ministry of Public Security of China, as an example, the Division Part of it will be set as "0001" adjoined by a 4-bit Organization Part to service 4.29 billion individuals at most.

Instance 2. If Tsinghua University becomes one of the organizations using NID, the Division Part could be set as "1000" with an 18-bit Organization Part sequence. Thus Tsinghua University can own nearly 260,000 members. The length of the User Part is 18 bits for this organization.

4.4 Scheme for generating User Part

Other than the Division Part and the Organization Part, the User Part is the sequence that intercepted from the string encrypted by SHA-256 as the final solution with the wide trade-offs. In this subsection, we would like to describe some insights or intuitions in selecting DID and SHA-256 as the basis of the User Part. Normally, an individual who is affiliated to an organization would be distributed with a unique ID number which is associated with his or her privacy information such as name, address, email or phone number. In the rest part of this paper, we use DID to describe that dedicated ID number. Under ordinary circumstances, ISP can consider the DID directly as the permission account for users to access the Internet.

To protect the Internet users from obtaining their basic information from DID, the User Part sequence should be calculated by some hash algorithms from the DID sequence instead of using it directly. We use SHA-256 to encrypt the DID sequence. The reasons why we choose SHA-256 are given in Subsection 7.1.

After the hash process of SHA-256, DID is transformed to a 256-bit string. According to the User Part's length calculated in Subsection 4.3, the string of the first n bits of the 256-bit string is used as the User Part. We apply a common but effective and efficient way, i.e., quadratic probing, to solve collisions in the hash table. Specifically, if the string of the first n bits of the 256-bit string is in the position of the hash table, and it is already occupied by another NID, then the User Part will use the string of the position a+12 in the hash table. If the string of the position $a+1^2$ in the hash table is also occupied by another NID, the User Part will try the string of the position $a+2^2$, $a+3^2$,$a+4^2$, , until a string in the hash table that is not yet used by any NID is found.

For the length n of the User Part, it takes n steps to compare an n bits intercepted string with a given n bits string in the hash table. In addition, the maximum number of the times of the comparisons is 2^n-1, and the minimum number is 1. In brief, the average number of the times of the comparisons is 2^{n-1}.

Overall, the average number of cumulative steps of comparisons involving in finding a suitable string in the hash table for the User Part is $n\times 2^n$. When n comes to the maximum number 34, the average number is 8589934592, and the comparison can be completed in less than 1s in normal servers if the frequency of CPU is larger than 1 GHz. Please notice that the comparison occurs only when a network user applies an NID, and we do not need the comparison in authentication and tracing.

As can be seen from the structure of NID, NID meets the principles we proposed above skillfully and subtly. In the NIDTGA program, the last 64 bits of IPv6 address (AID) are generated by NID and the time information. In what follows, the details are described to get the time information.

4.5 Demonstration of generating NID from DID

Given that a university has up to 70000 users and one student who has a DID number of 2014110001. According to the rules in Subsection 4.3, the lengths of the Organization Part and the User Part are

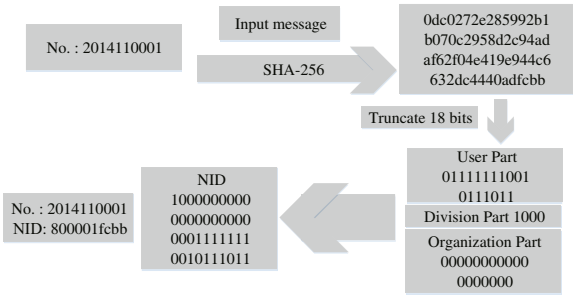

Figure 2 Example process of generating NID.

the same 18. For a general presentation, we define the value of the Division Part sequence and the Organization Part sequence as 1000 and 00000000000000000 respectively.

The process that NID is generated by DID of the specified student is depicted in Figure 2 by the following steps:

Step 1. Put the DID as the input sequence of SHA-256 algorithm;

Step 2. Get the output sequence:

0dc0272e285992b1b070c2958d2c94adaf62f04e419e944c6632dc4440adfcbb;

Step 3. Intercept its first 18 bits: 011111110010111011;

Step 4. Confirm that there is no collision with the 18 bits sequence in the database;

Step 5. Combine the Division Part, the Organization Part and the User Part together in order to get the NID: 10000000000000000000000011111110010111011 (in binary) or 800001fcbb (in hexadecimal).

5 Design of AID and its generation algorithm

5.1 Design of AID

In this paper, we focus on the generation of the last 64 bits of IPv6 address because we can obtain IPv6 address prefix according to the DHCPv6 server or ND protocol. There are many generation schemes of the last 64 bits of IPv6 address, such as EUI-64, privacy extension scheme in RFC3041, CGA and so on. In our scheme, we use IDEA algorithm to encrypt the 64 bits string (40 bits NID and 24 bits time information) to obtain the last 64 bits of the IPv6 address. We first introduce the 24 bits time information in AID in this sector, and then depict the encryption algorithm of AID.

The advantages of encrypting NID to obtain the last 64 bits of IPv6 address (AID) are as follows:

(1) NID has a hierarchical structure and avoids the mass of different user information in different ASes. It is convenient to realize the cross-domain traceback.

(2) NID has a uniform structure. Network users can use NID in other domains, so it is not necessary for users to use different basic identities in different ASes.

(3) NID is obtained through the calculation of the hash of the user's identity attribute, which avoids the decryption of the IPv6 address, thus can directly get access to the network user's basic identity information, and better protect the user's privacy.

5.2 24 bits time information

To make it convenient to change the key of encryption algorithm and enhance the security of NIDTGA scheme, we embed time information into AID to return to the original condition while tracing back. The time information embedded into AID is the time that we use NID to generate AID.

Specifically, the steps of generating the time information are as follows:

Step 1. Calculate the difference (in seconds) between the current time and 00:00 Jan. 1st of the current year.

Step 2. Use a proper degree of accuracy to change the time difference into a proper format, where the degrees of accuracy can be 30 s, 10 s, or 5 s.

Step 3. Transform the result of Step 2 into a binary string. If the length of the binary string is less than 24 bits, add some "0" ahead of the binary string.

For example, the time difference between 2013-12-31 23:59:59 and 2013-01-01 00:00:00 is 31535999 s. If we set the degree of accuracy as 1 min, the time difference is 525599 min. We transform it into the binary string '10000000010100011111', and its hexadecimal form is '08051f' after adding "0" before the binary string.

In this paper, the reasons why we use the above scheme to determine the time information are as follows:

(1) If we store the UNIX time (absolute time), the length of time information may be longer than 24 bits.

(2) It is not necessary to store the absolute time, because the aim of embedding time information into AID is to return to the original condition in which network users can get access to the Internet and manage the update of the keys in the encryption algorithm. It is almost impossible to traceback users' actions one year later. Moreover, the updated cycle of the keys is less than one year (introduced in the follow-up).

(3) To traceback the network users' performance and manage the update of the keys, there is no need to use second as the degree of accuracy. It depends on the need of time in different conditions to use different degrees of accuracy.

5.3 AID generation

In order to avoid analyzing NID according to AID, then analyzing the corresponding NID of any IPv6 address, and even the case of the theft of the NID, we use IDEA (International Data Encryption Algorithm) [11] to encrypt the 64 bits string of NID and time information.

IDEA algorithm is proposed by Massey, Lai et al., and it belongs to the block cipher. IDEA uses 128 bits key and the data block length is 64 bits. In theory, IDEA belongs to the "strong" encryption algorithm, and there is no effective attack on it.

As for the reason why we choose the IDEA algorithm, a security analysis of the IDEA algorithm is given in Subsection 7.2.

5.4 Process of generating AID by NID

Like the student example in Subsection 4.5, after a student gets an NID, we can encrypt his/her NID with time information through IDEA to get the AID. Assume that the time we use NID and generate AID is 2013-12-31 23:59:00, and the time degree of accuracy is 1 min. According to Subsection 5.2, we can get the time information 08051f (presented by the hexadecimal format). Assume that the key of address generation server is 6b48c2bd883461dc866e64bc5b40650b. Figure 3 shows the process of generating AID by NID 800001fcbb and time information 08051f.

Specifically, the process of generating AID with 40 bits NID and 24 bits time information is as follows:

(1) The 64 bits concatenated the string of NID (800001fcbb) and the 24 bits time information (08051f) is 800001fcbb08051f.

(2) Use 128 bits IDEA key (6b48c2bd883461dc866e64bc5b40650b) to encrypt the 64 bits concatenated string to get AID 68518eafc0993146.

6 NIDTGA system design

6.1 System structure

In the NIDTGA system (Figure 4), there are three kinds of servers: Address Generation Server, NID Management Server and NID Traceback Server. Address Generation Server generates the IPv6 address

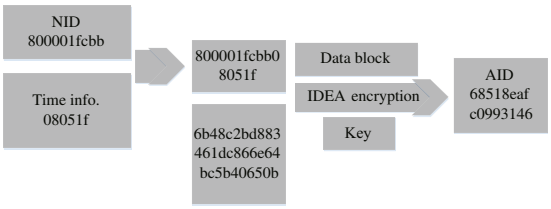

Figure 3 Example of generation process of AID with NID and time information.

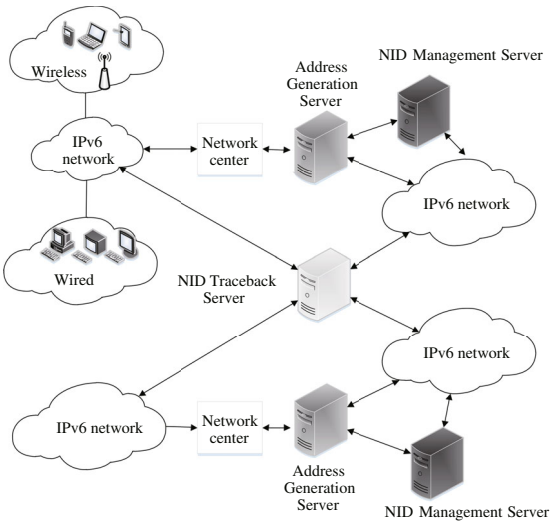

Figure 4 IDEA key distribution and management.

based on NID. NID Management Server generates, manages and distributes NID. NID Traceback Server traces back to the network users' real identities based on the IPv6 address. The detailed functions are given as follows:

(1) Address Generation Server. It implements the basic functions of DHCP and SLAAC. After it starts, it sends the 128 bits IDEA key to the NID Traceback Server. According to the concatenated string of NID and time information, it uses IDEA to encrypt the concatenated string to generate AID. Moreover, it will send AID to the requested host, and the host can modify the last 64 bits of IPv6 address with AID.

(2) NID Management Server. It generates NID based on the host's DID and validates the user's identity with the host's NID and password. It can search identity information according to NID that the NID Traceback Server provides.

(3) NID Traceback Server. According to the first 64 bits of the IPv6 address, it can determine which AS the IPv6 address belongs to. It can use a key to decrypt AID to get NID and time information. It can send NID to NID Management Server to search NID's identity information. It can send NID's identity information to the routers and destination host.

6.2 IDEA key distribution and management

Although the brute-force attack on IDEA is not easy, other kinds of attacks cannot be eliminated. In addition, the process of transport keys has the risk of leaking the keys.

To improve the security of IDEA algorithm in NIDTGA, Address Generation Server updates the IDEA key (one hour for example) and sends the updated key and the corresponding time to the NID Management Server. When the Address Generation Server modifies the host's IPv6 address, the last 64 bits of IPv6 address AID is encrypted from 40 bits NID and 24 bits time information by IDEA. That is, AID contains time information.

When NID Traceback Server stores the key, it stores the identification of Address Generation Server (like IP address of Address Generation Server) and time.

When one administrator sends a request for the user's identity of some IPv6 addresses to the NID Traceback Server, he sends that IPv6 address. At first, NID Traceback Server determines the IP address of Address Generation Server based on the first 64 bits of the IPv6 address, and then uses the Address Generation Server's latest key to decrypt AID. If the time information is in the duration of generating the key, we get the right information. Otherwise, use the last key to decrypt AID. Specifically, assume that the time t corresponds to k_t and the cycle of the updating key is d, after the NID Traceback Server

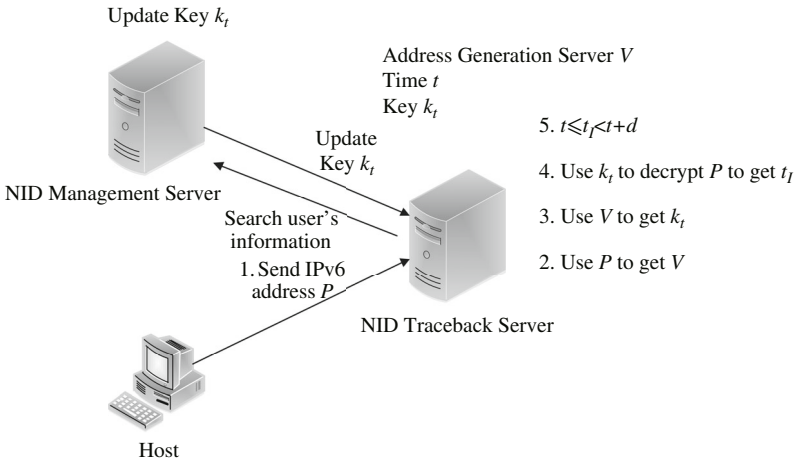

Figure 5 Key distribution and management of IDEA.

receives the IPv6 address P:

(1) Determine the identification V of Address Generation Server according to the first 64 bits of P;

(2) Use V to determine the latest key of Address Generation Server, that is, corresponding key of time t is k_t;

(3) Use k_t to decrypt the last 64 bits of P to obtain NID and 24 bits time information string I;

(4) If the time k_I satisfies $t \leqslant k_I < t + d$, the process ends;

(5) Otherwise, set $k_t = k_{t-d}$, and go back to Step (3)).

In NIDTGA, Address Generation Server is in the core of the key distribution and management of IDEA algorithm. Address Generation Server updates the key and sends it to NID Traceback Server by using SSL.

Figure 5 shows the key distribution and management of IDEA algorithm.

In NIDTGA, 24 bits time information and 40 bits NID consist of 64 bits string. To protect the privacy of the network users, the NIDTGA scheme uses IDEA to encrypt this 64 bits string to obtain 64 bits cipher that works as AID. Once we determine the NID generating scheme and the AID generating scheme, the corresponding AID can be generated when NID is obtained. Then we have an IPv6 address containing NID and time.

6.3 Cross-domain authentication

Since different management domains may have different identities and authentication techniques, we need to design two kinds of schemes to solve the problem of the generation and traceback of the IPv6 address in the cross-domain scene. One is federal authentication based on multi-domain coordination. Another is centralized authentication based on the central authentication server. Figure 6 shows the exchanged information of these two schemes.

6.4 Traceback

When the NID Traceback Server gets an IPv6 address that we want to trace back to, it extracts the AID of this IPv6 address. According to Subsection 6.2, it can determine the corresponding key to decrypt the IPv6 address. Figure 7 shows the process of obtaining the identity of network users based on AID and the key.

The process of extracting the user's real identity based on 64 bits AID and 128 bits key is as follows:

(1) Make AID 68518eafc0993146 as the data block and 6b48c2bd883461dc866e64bc5b40650b as the key. Then we use the IDEA algorithm to decrypt the AID and to obtain the 64 bits string 800001fcbb08051f.

(2) Split 800001fcbb08051f, and we get NID 800001fcbb and time information 08051f.

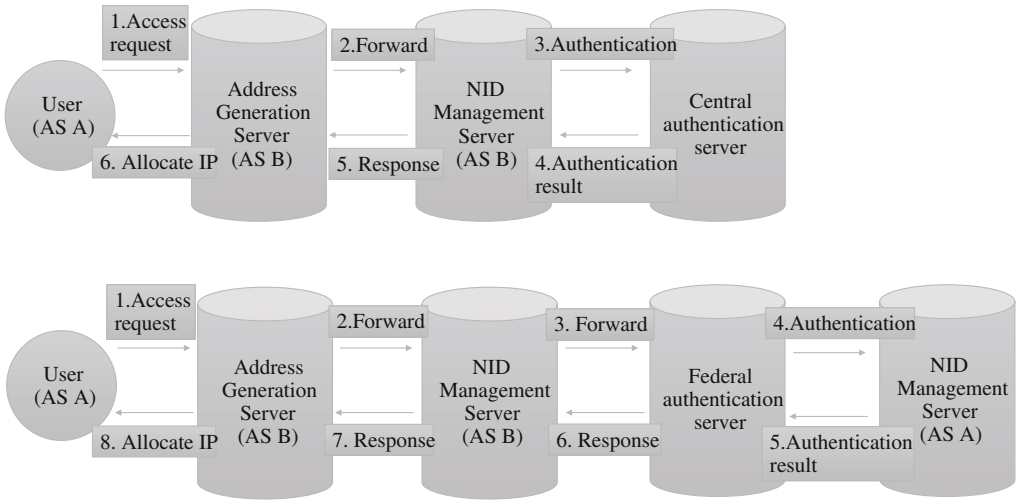

Figure 6 Federal authentication and centralized authentication.

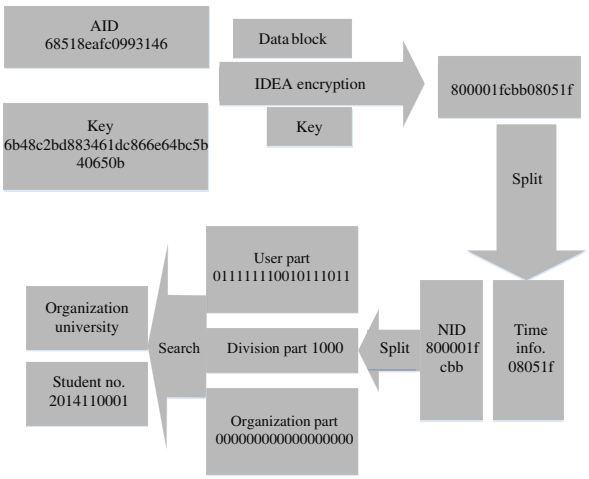

Figure 7 Use AID to trace back to network user's identity.

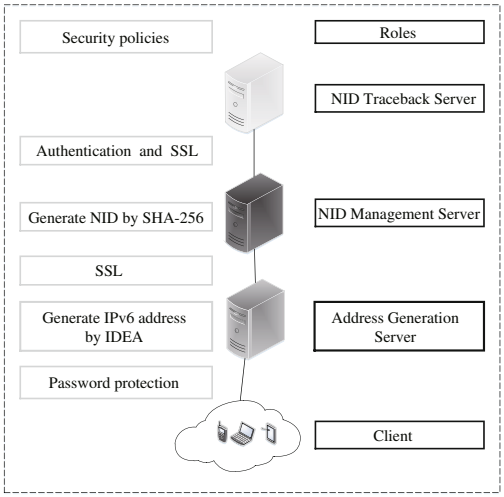

Figure 8 Security solutions in NIDTGA.

(3) Split NID, and we get the Division Part 1000, the Organization Part 000000000000000000, and the User Part 011111110010111011.

(4) According to the Organization Part, we can determine which organization the user belongs to. We know the user's DID 2014110001 based on the User Part.

We demonstrate the process of obtaining DID based on the above AID. The implementation requests the collaboration of Address Generation Server, NID Management Server and NID Traceback Server in the NIDTGA scheme.

7 Security analysis

NIDTGA scheme embeds NID into IPv6 address. NIDTGA scheme runs based on SAVA which ensures that the IPv6 addresses are authentic. In this section, present the security analysis of the NIDTGA scheme is presented.

Figure 8 shows the security solutions in the NIDTGA scheme.

(1) When a client wants to get an IPv6 address to get an access to the Internet, he/she needs to show his/her NID and password to the NID management server. We protect the client's password in the

process of authentication.

(2) We use the IDEA algorithm to encrypt NID to get the IPv6 address.

(3) The communications between address generation server and NID management server must be protected under SSL.

(4) We use SHA-256 to encrypt DID to generate NID.

(5) As for the communicating messages between NID management server and NID traceback server, it is best to use a bidirectional authentication between these two servers. Moreover, their communications should be protected under SSL.

7.1 Why choose SHA-256

In recent years, SHA (Secure Hash Algorithm) has been the most popular algorithm among all hash functions. Because of the defect discovered in other hash functions, actually, since 2005, SHA has become the only remaining standard of the hash algorithm. In 2002, NIST (National Institute of Standards and Technology) released the reviewed FIPS 180-2, which published 3 new formats in the clan of SHA: SHA-256, SHA-384 and SHA-512 according to the length of the encrypted length, and that is SHA-2. Similar to SHA-1, SHA-2 uses the same iterative structure, mold arithmetic and binary logic operations. In addition, SHA-2 is also depicted in RFC 4634 [12] as nearly a duplication of FIPS 180-3.

It is apparent that SHA-2 is more secure than MD5 and SHA-1, so we have three choices to encrypt DID: SHA-256, SHA-384 and SHA-512. Because of the truncating operations on the encrypted results of our selected hash algorithm, we should choose SHA-256 whose output is the shortest among the three proper algorithms. Once we choose another two, the collision possibility is becoming bigger when we truncate the output messages.

7.2 Why choose IDEA

In NIDTGA, the requirements of the encryption algorithm used to encrypt 40 bits NID and 24 bits time information to obtain AID are as follows:

(1) To trace back to the users' real identity, the encryption algorithm in the AID generation scheme must be symmetric. That is, we can encrypt the blocks of data to get the cipher text, and we can decrypt the cipher text to get the plaintext.

(2) Considering the fact that the length of the concatenated string of 40 bits NID and 24 bits time information is 64 bits, the encryption algorithm must support the encryption of the 64 bits data.

(3) Considering the fact that we use the encrypted cipher text as AID, the encryption algorithm must generate the 64 bits cipher text.

There are several popular encryption algorithms, such as DES, 3DES, AES, IDEA and so on. Table 2 shows the comparisons of the above algorithms. According to Table 2, AES does not meet our requirements and DES is not secure now. Therefore, we can use 3DES and IDEA in the AID generation algorithm. However, the two or three keys of 3DES will lead to more complexity of the distribution and management of the keys. Last, we choose the IDEA algorithm to encrypt NID and time information to obtain AID.

In EuroCrypt'97 conference, John Borst [13] et al. propose two kinds of attacks decreasing the rounds: truncate differential attack on 3.5 rounds IDEA and differential linear attack on 3 rounds. However, they also point out that these two kinds of attacks had no influence on 8.5 rounds IDEA in essence.

In 2007, applying all keys' optimal attacks can crack 6 rounds IDEA [14]. This "crack" means that the operations are less than 2128. The attack on IDEA needs 264 known messages and 2126.8 operations.

In 2012, Dmitry Khovratovich et al. realize an attack on IDEA [15]. They use narrow-bicliques to attack IDEA and make effective keys' length decrease 2 bits. However, they also point out that the analysis complexity of their cryptographic algorithm is too high and it has no influence on the application of IDEA in reality. Currently, there are many applications of IDEA.

Table 2 Comparison of several encryption algorithm

	DES	3DES	AES	IDEA
Plaintext block length	64 bit	64 bit	128 bit	64 bit
Length of key	56 bit	168/112 bit	128/192/256 bit	128 bit
Encryption rounds	16 bit	48 bit	10/12/14 bit	8 bit
Others	Bad security	Complexity in managing keys	Block length longer than 128 bit	None

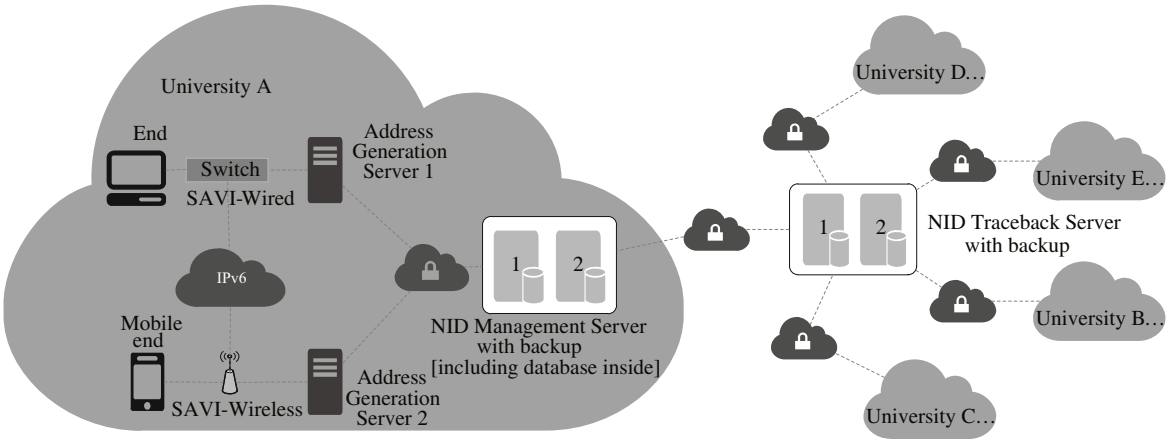

Figure 9 Experimental deployment of NIDTGA.

8 Experimental deployment

The prototypes of our solutions for NIDTGA are implemented. The deployment of NIDTGA is being carried out with the participation of 5 universities.

As is shown in Figure 9, each university deploys 2 NID Management Servers and 2 Address Generation Servers. The NID Management Servers contain user information database. A backup server is needed. 2 Address Generation Servers are intended to serve for different address generation schemes or balance the load of each other. In the center part of the deployment, the NID Traceback Server is essential for users to login across the domain, or for administrators to trace the users. Due to the crucial function we described in this paper, a backup is indispensable.

NIDTGA is a combined system which merged the IPv6 address allocation system (as the alternative of DHCP, etc.), authentication system (as the alternative of Web Authentication, etc.) and traceback system (a mature cross domain system does not exist yet) together in the IPv6 environment.

Due to the particular mission of NIDTGA, it is not very appropriate to evaluate the performance directly with some other solutions only for address allocation, only for authentication, or only for traceback. We focus the performance on the overall solution deployment. Future efforts need to be done in this area.

9 Conclusion

This paper describes a scheme named NIDTGA that generates IPv6 address based on network identity, time information, an IPv6 Address Generation and the Trace-back System.

First, we introduce the NID generating scheme. NID has three parts: Division Part, Organization Part and User Part. We use SHA-256 to encrypt DID and truncate the cipher to get the User Part in NID. NID generating scheme sticks to the principle of scalability, hierarchy, flexibility, privacy, memorization and usability.

After getting 40 bits NID and concatenating with 24 bits time information, we use IDEA to encrypt the 64 bits string to obtain the last 64 bits of the IPv6 address, that is, AID. We prove the feasibility of IDEA and realize the period update of the IDEA key.

As for future work, like the successful deployment of CerID and SAVI, we plan to deploy the NIDTGA scheme on CNGI-CERNET2 based on SAVI. Then we will validate the feasibility of this scheme and modify the shortcomings.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No. NSFC61402257), National Basic Research Program of China (973 Program) (Grant Nos. 2009CB320500, 2009CB320501), and Tsinghua University Self-determined Project (No. 2014z21051).

References

1 Clark D, Braden R, Sollins K, et al. New arch: future generation Internet architecture. Technical Report, DARPA, MIT, ISI, 2003
2 Aura T. Cryptographically Generated Addresses (CGA). RFC3972. 2005
3 Moskowitz R, Nikander P, Jokela P, et al. Host Identity Protocol. RFC5201. 2008
4 Nordmark E, Bagnulo M. Shim6: Level 3 Multihoming Shim Protocol for IPv6. RFC5533. 2009
5 O'Dell M. GSE-an alternate addressing architecture for IPv6. 1997
6 Kunishi M, Ishiyama M, Uehara K, et al. LIN6: a new approach to mobility support in IPv6. In: Proceedings of 3rd Inernational Sympsium on Wireless Personal Multimedia Communications, Bangkok, 2000. 43
7 Andersen D G, Balakrishnan H, Feamster N, et al. Accountable Internet protocol (AIP). ACM SIGCOMM Comput Commun Rev, 2008, 38: 339–350
8 Yan Z M, Zou X, Jin B. Ordered activities depending on eID in cyber virtual society (in Chinese). Netinfo Secur, 2011, 3: 005
9 Wu J, Bi J, Li X, et al. A Source Address Validation Architecture (SAVA) Testbed and Deployment Experience. RFC5210. 2008
10 Wu J, Bi J, Bagnulo M, et al. Source Address Validation Improvement (SAVI) Framework. RFC7039. 2013
11 Daemen J, Govaerts R, Vandewalle J. Weak keys for IDEA. In: Proceedings of 13th Annual International Cryptology Conference, Santa Barbara, 1994. 224–231
12 Eastlake D, Hansen T. US Secure Hash Algorithms (SHA and HMAC-SHA). RFC4634. 2006
13 Borst J, Knudsen L R, Rijmen V. Two attacks on reduced IDEA. In: Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, 1997. 1–13
14 Biham E, Dunkelman O, Keller N. A new attack on 6-round IDEA. In: Proceedings of 14th International Workshop on Fast Software Encryption, Luxembourg, 2007. 211–224
15 Khovratovich D, Leurent G, Rechberger C. Narrow-Bicliques: cryptanalysis of full IDEA. In: Proceedings of 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, 2012. 392–410