Master's Thesis

Title

Design and Implementation of
Secure IPv6 Communication Architecture using
Non-negotiated Specific Service Addresses

Supervisor

Professor Masayuki Murata

Author

Kazuyuki Nishida

February 15th, 2011

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Design and Implementation of
Secure IPv6 Communication Architecture using
Non-negotiated Specific Service Addresses

Kazuyuki Nishida

Abstract

In the current Internet architecture, IP address used for the node identifier, that is, generally a single IP address is assigned to a node, and used parmanentally until the node becomes inactive. The same address is used for all communications from/to the node.

However, this communication paradigm has a fundamental problem regarding security that the information of IP address of the node is open not only to nodes who intend to communicate to it, but also to anonymous parties who try to attack the node.

To solve this problem, we change our traditional paradigm completely and propose a new solution called *Unified Multiplex Communication Architecture*. The most difference from the current Internet is that an IP address is not used for node identifier, but for service identifier. In the Unified Multiplex Communication Architecture, we change IP addresses session-by-session, and the assigned address is invalid immediately after the session terminates. This architecture simply changes the direction for use of IP address but enhances the security significantly.

However, there is a major issue on Unified Multiplex how to determine the IP address to connect the server, since IP address is assigned to session one-by-one. Prior to communication, the client should know the IP address of the server which is used for awaiting the connection from the client.

For this problem, in this thesis we propose a new, non-negotiation type IP address determination mechanism that is feasible by updating the operating system on end hosts only (no modification of application is needed). In our mechanism, IP address generation is performed on both server and client independently, but generated addresses are synchronized because time information is used for address generation. We then analyze the interval of address update (i.e., the lifetime of generated address) for avoiding unexpected failure due to our mechanism. Our numerical result shows that our address update mechanism is extremely robust against brute-force type attacks. Moreover, detailed design and implementation methods are described for realization.

Keywords

Communication Architecture
Secure Communication
IPv6

Contents

List of Figures

List of Tables

1 Introduction

Recently, due to the significant increase of number of Internet nodes, the problem called *IPv4 Address Exhaustion* has become critical and serious, and strong world-wide cooperation is mandatory to solve it urgently [1]. IANA (Internet Assigned Numbers Authority), a headquarter of management of IP addressing space, has completed its major task to assign unallocated IP address block to its subordinate agencies (called RIR; Regional Internet Registry) on February 3, 2011. No more IP addresses will not be offered by IANA. Moreover, it is expected the last date of IP address allocation by RIRs to their subordinate coutry organizations will be around November 2011 (in Asia-Pacific region). After that, it is extremely hard to ISPs (Internet Service Providers) to get IP address as requested.

Since the IPv4 Address Exhaustion is a fundamental problem due to the limitation of structure of IPv4 where its length of addressing is only 32 bits, a complete solution is to migrate the new version of IP called IPv6 (Internet Protocol version 6) which has huge addressing space (128 bits) enough to accomodate all nodes even in the future. To keep the global connectivity to all and upcoming nodes continuously, urgent, smooth, but fast migration to IPv6 is major task in several years. In Japan, Ministry of Internal Affairs and Communications leads to take measures for IPv4 Address Exaustion, and the Task Force on IPv4 Address Exaustion was established to take actions cooperatively among network operators.

Though it is commonly agreed that the motivation behind the migration to IPv6 is to extend the addressing space, it is also a big chance to solve other problems on the current Internet communication style together. One of such problems can be found in the security.

Traditionally, the Internet is designed and operated based on the ethical doctrine that human nature is fundamentally good. For example, there is neither mechanism to prevent packets from ill-behaved or malicious TCP nodes, nor mechanism to validate the mail sender, nor mechanism to avoid any other attack traffics. As the result, users who intend to connect their equipments to the Internet must take security risks into consideration. Regardless of experience, all people have to take care about the security for safe communication. Such requirement is so strict and enough to let people feel that *the Internet is something dangerous.*

One of interested issues on the security is anonymity and privacy in the Internet. Anonymity in this thesis is defined to satisfy the following characteristics.

- **Uni-directionality:** The server can not connect to the client using the client's source address in the packet originated the clients.

- **Discontinuity:** Prevent grouping the same addresses recorded in the communication session, so that it is difficult to guess the behavior of the node by monitoring a set of sessions associated by the same address.

- **Dissimilarity:** Even if the server address is known to a third party, it can not communicate to the server by using the known address or some other addresses inferred from the original ones.

Particularly as the importance of the content in the communication increase, the association of communication and specific nodes by third parties encompasses the potential factors such as privacy leaks and reduced security strength. In order to establish a safe and secure Internet communication infrastructure, it is highly desirable to provide anonymity and privacy as a function of the network layer.

From the research background in the above, we consider the transition to the IPv6 is a great opportunity to replace the network layer. We propose a new communication architecture that realizes safe and secure communication infrastructure that provides anonymity and privacy by taking an advantage of the vast IPv6 address space.

Using Unified Multiplex [2–4] that uses different IPv6 address for each communication session, we propose a new communication architecture with a high anonymity just by implementing the system in the end nodes.

We first describe the parameter setting that maintains connectivity and ensures safety plus the detailed structure and the implementation method of communication architecture. In addition, we verify the proposed communication architecture by the quantitative evaluation of experimental network.

The rest of this thesis is organized as follows. Section 2 states the problem and surveys existing studies. Section 3 presents the outline of Unified Multiplex.

The proposed network architecture is described in Section 4. Section 5 describes the implementation approach. Section 6 evaluates the validity of the proposed architecture. Section 7 concludes this thesis by briefly summarizing the main points and mentioning future work.

2 Problem Statement and Related Work

In this chapter, We describe the specific issues addressed in this thesis. Then We discuss the problems raised regarding the approaches in the past researches.

2.1 Problem Statement

The problem we address in this thesis is that the current Internet does not ensure anonymity. In this thesis, we propose the communication method to secure the anonymity of servers. The anonymity is defined as follows.

- **Uni-directionality:** The server can not connect to the client using the client's source address in the packet originated the clinet. Since this states the anonymity of the clinet, we do not describe in detail in this thesis. This problem is solved by changing the source address of each session used when the client connects to the server by the Unified Multiplex communication architecture that we propose.
- **Discontinuity:** By grouping the same addresses recorded in the communication session, multiple communications are associated so that it is difficult to guess the behavior of the node.The node does not use the same address so that it can not be associated by grouping and it is necessary to change the used address.
- **Dissimilarity:** Even if the server address is known to a third party the server can not be communicated using the address or some other addresses inferred from the original address.The server should change the address. Moreover, the updated address should not be something that can be inferred nor guessed from the old address.

The technique for securing anonymity should be able to be used anywhere in the Internet. Given the eses of implementation the proposed method should only be completed by the end host without relying on intermediaries such as routers.

2.2 Related Work

The past researches on anonymity can be classified into three types; achieving anonymity in the link layer, in the network layer, and in the overlay network.

The proposal by Author et al. [5] achives anonymity in the link layer by encrypting MAC address with a pair of key and sequence number.

Encrypted MAC address provides anonymity by varying sequence number.

However, this anonymity is guaranteed only in a single segment of networks, such as wireless networks and does not provide global anonymity.

Network Address Translation (NAT) [6] is dominantly used to acheive anonymity in the network layer. NAT was designed to solve the IPv4 depletion. Where a single global IP address is shared with multiple nodes. In addition, NAT provides anonymity for nodes since it is unidirectional. Even IPv6 implements the idea of NAT one of where the initial goal was to eliminate the NAT. But,there is consideration of NAT of high security [7].

However, NAT destroys end-to-end principle of the Internet. As a result, there is a serious operational issues such as being unable to detect the cause of network trouble especially when applications show unexpected behavior.

Tarzan [8] [9], Crowds [10], ANON [11] [12], TOR [13] , and the proposal by Author et al. [14] have provide anonymity for nodes by constructing overlay networks and having the traffic go around multiple nodes.

The proposal by Author et. al [15] verifies the utilization of TOR. TOR provides anonymity for nodes with a low latency. But, the throughput of TOR is less than expected in addition, middle nodes of router and other nodes are newly-implemented in overlay network method. Therefore, there are difficulties in migration to the new system from the conventional one. There is also a method of setting a firewall but secure server operation is difficult to achieve without a complex configuration.

3 Achieving Anonymity and Privacy by using Unified Multiplex Communication Architecture

In this chapter, we describe the outline of Unified Multiplex. In addition, we describe the past research on notification of service address and describe the proposal method.

3.1 Brief Introduction of Unified Multiplex Communication Architecture

The current communication architecture generally assigns an address to a node and always provides services by the same address. In this communication method, the server's service address is reachable by any node if the address is publisized or scanned by third parties. Therefore, The anonymity of server's service address should be protected for a secure network environment to avoid unnecessary access from third parties.

However, the service address in the conventional communication method is used for fixed and for a long period of time. Therefore, it is difficult to ensure the anonymity when the address is being published.

In order to solve these problems, we propose a new communication architecture called Unified Multiplex. Unified Multiplex assigns an address to a service. We review the condition of the conventional communication that a single node has a single address.

The server's service address is called SSA (specific Service Address) in Unified Multiplex.

SSA are assigned for each sessions. When clients connect to SSA, third parties can not communicate using the particular SSA. Since it is already use.

In addition, even if the SSA is revealed to third parties, it can not be used since it is not used anymore.

As shown above, Unified Multiplex ensures the anonymity of servers by using an address per a session.

This method needs a mechanism for client to receive SSA since SSA changes in each communication. We discuss the three mechanisms in the past research in the next Subsection.

3.2 Problems on Sharing Specific Service Address

We discuss the problem of the three methods in the past research.

Table 1: Method of receiving SSA

	Generate on ahead	Generate on demand
External service	Registry service address to DNS	DNSO [16]
Only end node	Notice list of service address on ahead	Sharing rule of generating address

- **Registry service address to DNS:** When server begins the service, server registers service address to DNS. Client changes the FQDN to service address. Then, Client can communicate using the service address. The target of this thesis is closed environment. Client can use service anywhere in the Internet. Client that receive the service address is hard to be pinpointed. However,we do not use this method since the mechanism of DNS has to be changed to allow connections regardless of the connection point since there is a DNS server per a network where it is connected.

- **DNSO:** Server begin to listen to service when the DNS query of the client is detected. We need to implement this method for DNS server of the clients side. This method does not work correctly when the client application issues a single DNS query but multiple access. Since, this method uses DNS query. We need to change the client and the server as well as DNS in this method. Therefore, this method is not used here.

- **Notice list of service address on ahead:** Server generates service addresses on ahead and notices the address list. Client can connect to the address that is described in the address list. Only the end node change in this method. Therefore, the implementation cost is small and client can connect anywhere. However, if client's connection is beyond the number of noticed address list, server notice client address list once again. This address list notification has a risk for exposure of service address to third parties. The proposal method does not need to communicate the address information of server and client. Therefore, notice list of service address on ahead has a higher risk compared to the proposed method.

3.3 Approach for Secure Communication

3.3.1 Architectural Overview

In this Section, we give an outline of the proposed method. In addition, we show the issues that have to be solved in designing the proposal. The outline of the architecture is shown in Figure1.

- The information that has to be shared between the server and the client.

    1. Pass phrase
       The target of this thesis is closed connections in home gateway. Therefore, sharing pass phrase between the server and the client has a low risk. Since, the operator who configures the setting of server can configure the setting of client without connecting to the global network.

- Server

    1. Generates address by entering time information and the pass phrase to the address generating module.

    2. Listen to the service by the regularly generated address.

    3. Delete unused address when the server starts listening the next address
       When server starts to listen to next service address, server deletes address which is assigned LISTEN state (unuse) socket. In this way, we achieve short time establish address. This method has a to tolerance for brute force attack [17] for scanning third parties.

- Client

    1. Regularly generate accessible address by entering the pass phrase and the server's host name to the address generating module.

    2. The generated address is written to the configuration file ($/etc/hosts$) to map the host name and the IP address.

    3. Client can connect to the address using the generating sharing rule by the host name.

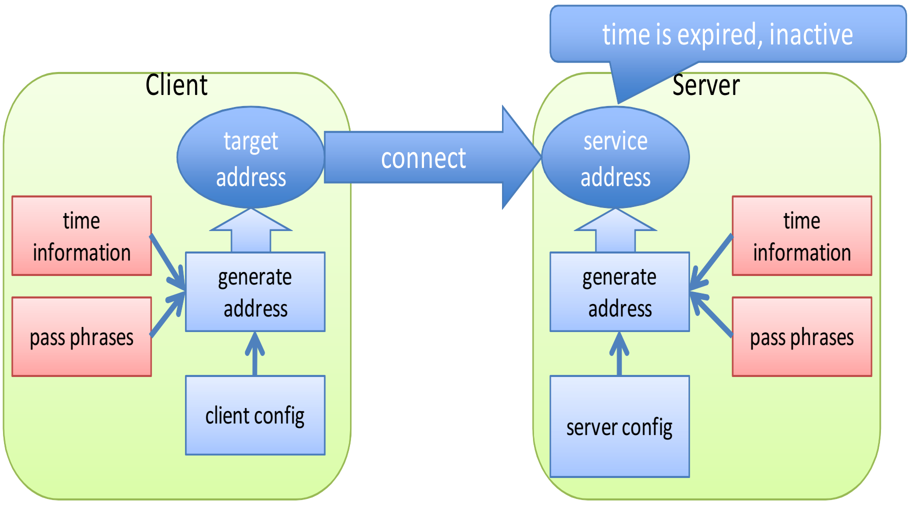


Figure 1: Communication architecture ensuring the security by sharing the generation rule

3.3.2 Setting Lifetime of Address by Achieving Security and Connectivity

By setting the server service is listening for a long time, brute force attack by, increasing the risk of a particular address. However, if you set a short time to listen, time consuming and can not connect to somewhere in the handling of unexpected connections from legitimate clients would stress to the user. You must set the appropriate service waiting time is determined by the following four conditions.

- The influence with DAD of address (D): When address is changed with the server, it will overlap in advance, but (DAD) it is necessary to verify whether it is not. Verification delay one seconds which are decided with RFC4862 [18] causes DAD, use of address is after the DAD completing. Therefore, being the server client both, when it modifies simultaneously in new address, the possibility of connecting to the address which on server side does not become effective occurs.

- Influence (d) by the propagation delay between the server clients: When connecting to the server from the client, the delay for one from the client to the server is generated.

- The influence with the retransmission of the SYN packet: At the time of TCP connection, at the stage where the first SYN packet is forwarded, it is necessary to make address fixed. But, until the SYN packet when it disappears with congestion, connection completes with retransmission, if between, address is not fixed, not be able to receive the retransmission SYN packet at the point in time when address is modified, it cannot establish connection. Therefore, it is necessary to make the time address which is sufficient for the retransmission of the SYN packet effective. With FreeBSD, as for retransmission of the SYN packet after the backoff of the random time of 3 - 24 seconds, maximum of 8 times it is done. Therefore, term of validity of average 13.5 × 8 = 108 seconds is necessary.

- Time for collision evasion of address: In order to prevent the address discovery with brute force from outside, as for address it is necessary to renew at sufficiently close interval. Below, you express necessary time here.

It decides the setting of optimum address term of validity on the basis of whether the address space the subordinate position 64bit which can be set freely with the inside user discretion of 128bit of the address length of IPv6 there is strength how much vis-a-vis the attack frequency of brute force attack. Here, brute force attack it increases address value sequentially, the case where you tried the even 2^{63} trying 10000 connections during the time catching and 1 seconds, $7 * 10^8$ year it is required. In addition, the occasion where the attack person selects address value to random, when Birthday Paradox [19]it is considered, as for the probability which the attack person selects with random when designating the frequency of trial as n, to show with formula below it is possible the address value which the server selects in the address space of 64bit.

$$1 - (2^{64} - 1)/2^{64} * (2^{64} - 2)/2^{64} * (2^{64} - 3)/2^{64} * \cdots (2^{64} - n)/2^{64} \tag{1}$$

The address update time can be decided , considering above as follows. Figure2 shows figure where the address update time was shown.

The delay necessary for C_i and DAD is assumed, and here, the delay for one between D and the server client is assumed and L_S and time of the TCP sending again are assumed to be t_r at d and the address validity term at the reference time of S_i and the client at the reference time of the server to generate the addresses of the i turn eyes. The reference time of the client and the server , considering the influence of the delay for DAD and one

$$C_i = S_i + D - d \tag{2}$$

However, d is a value that greatly depends on the environment, and minimum value is $d = 0$.

$$C_i = S_i + D \tag{3}$$

The address generation time and the address waiting start time of the server should begin for time of the reference to the address of the client before the second of $D(1)$ necessary for DAD of the server.

It synchronizes between nodes at time [20], and a safe communication has been achieved by use of a key. The key will be switched for a fixed time, and be used. The key lifetime is provided besides the use switch of this key. The error margin of the time between T and the node is provided e, and the range where W can be taken is decided as follows at W and the key lifetime at update intervals of the key.

$$W \leq \frac{T}{2} - e \tag{4}$$

The validity term of the difference address grows for 108 seconds from the switch interval in the use address so that the proposal technique may consider sending TCP again at switch intervals between validity term and the use address in the address. The minimum value of the address at validity term becomes 216 seconds from expression4.

3.3.3 Supporting Services using Multiple Connections Simultaneously

It is not possible to correspond by Unified Multiplex communication architecture to which one service one address is required and a basic design of this address generation rule common mechanism when two or more sessions are established to the address where the client is the same. Then, ”Single Address Multi Listen” that was the technology to establish two or more sessions to one address to have interchangeability with the communication method that established two or more sessions to such one address was mounted. This generates two or more LISTEN sockets where the same address is allocated, and is the one corresponding to the connection of simultaneous plurals from the client. However, because it becomes possible to connect from the third party when the address that the client connects by this technology leaks, the function to permit only the connection from the same source address is needed for the address that the connection established once.

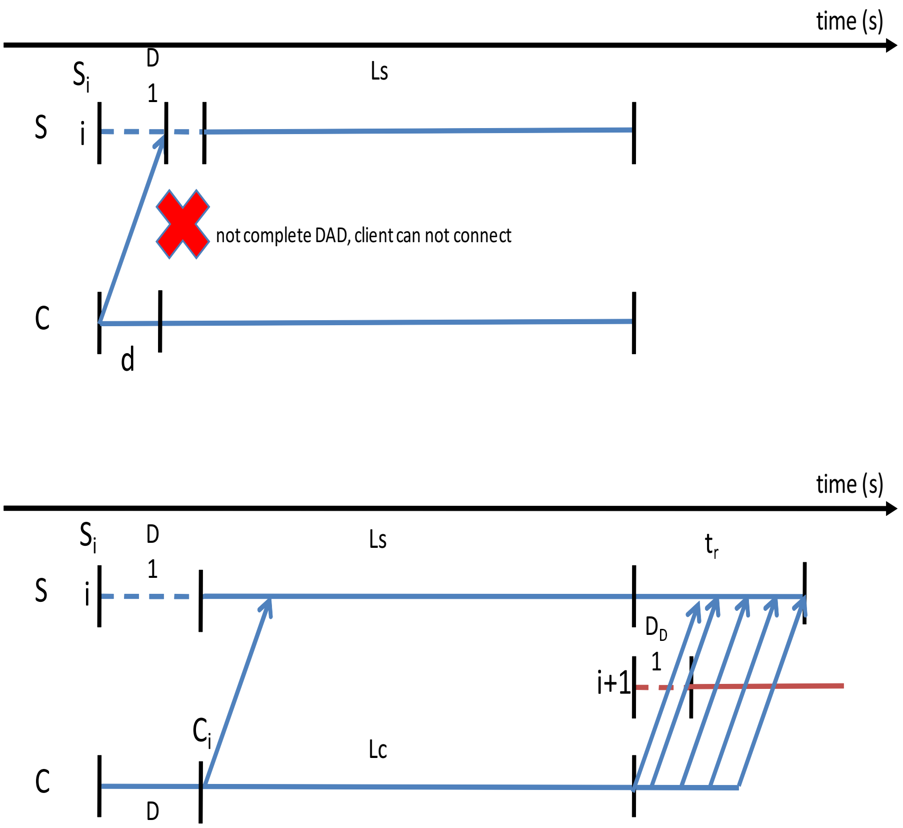


Figure 2: Address update time

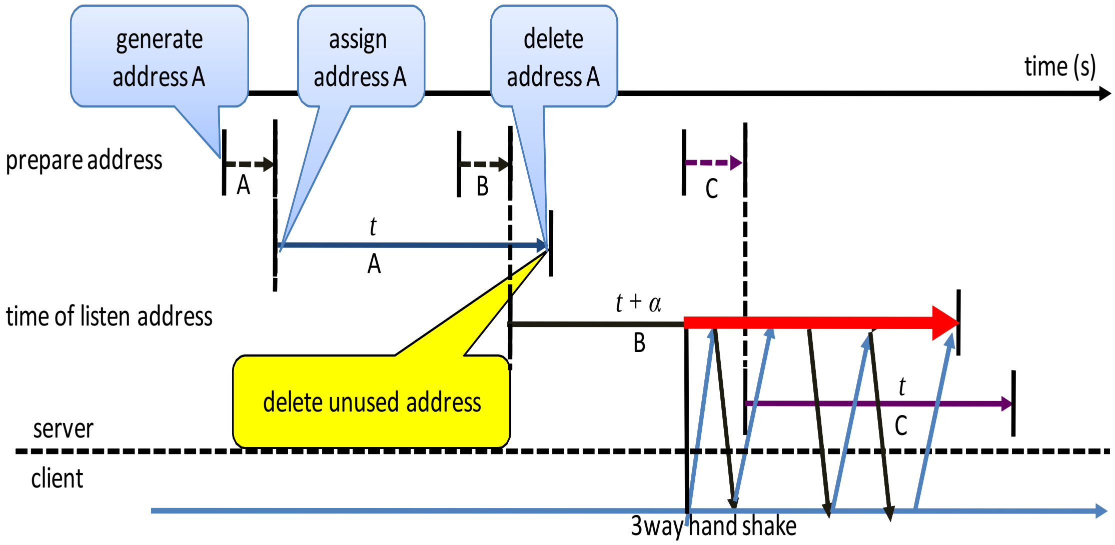


Figure 3: Time-sequence diagram from address generation to deletion

4 Design and Implementation Method

This chapter describes the design and implementation of the proposed method. First, we show that there are two types of implementation of the proposed method will be described using a block diagram of the flow of work for each method. For details of implementation will be described below.

4.1 Block Diagram of Non-negotiable Address Sharing

Describe the design and implementation of the proposed two kinds. Client implementation of this proposed the following two. The goal for each client is updated every time the connection type and constant listening to calculate the addresses for each server you want to connect to client We have proposed two kinds of type to calculate the hourly updates of the server listen address. On the server side also proposed the design of both the design and implementation of the same on the client side, different implementation.

The time is updated every type can be achieved by simply calculating every 200 seconds set by this address as a means to achieve the $/etc/hosts$ will be edited every time. This was designed as a way to solve a static user name $/etc/hosts$ and so different from the original purpose of the $/etc/hosts$ every time you edit and say the preferred implementation. We resolve to be called when the API address `getaddrinfo()` [21] hook, destination services Only when we determine that this method to generate the server's listen address, according to suggestions made in connection with updating every address that connects to calculate waiting.

Using the flow describes a block diagram of the operation(Figure.4、5).

The server works as follows.

- Address the listening process calculation programs, which reads the configuration file that describes the different passphrase for each client to get connected to boot. The client for the server address I should have generated each time a user requests to update the address always comes when you do not know the client is connected, you need to listen. To address the update time, using the control program as the program information for controlling the address previously used per process. It also reads the configuration file, the update timing to calculate the address of standby time from a single address. This is a common client.

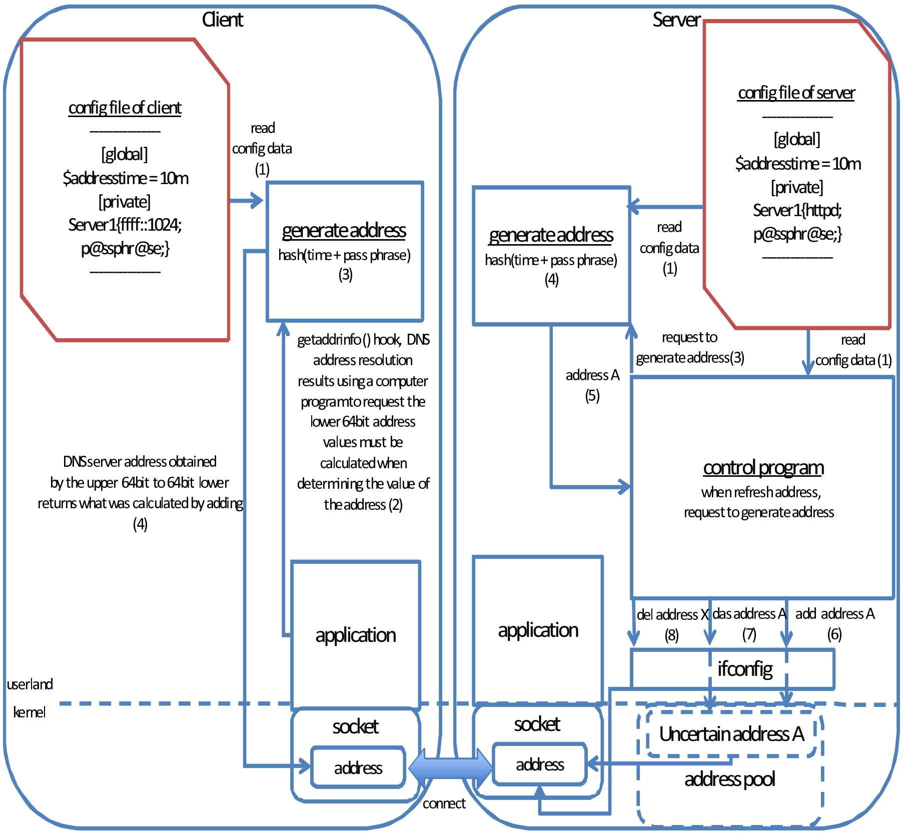


Figure 4: Block diagram of connection-driven approach

- Time to calculate the address update program information and control. Uniquely determined if a single address standby time is decided, as well as the earlier client.
- To assign a date and address to a socket address updates using the command `ifconfig` assigns an address for the process.
- After assigning the address, delete the unused address
- Address used to pre-compute the next address value calculation using the address information for next time updates to be available soon.
- Immediately ready for use in computed address to the Uncertain Address cause a transition.

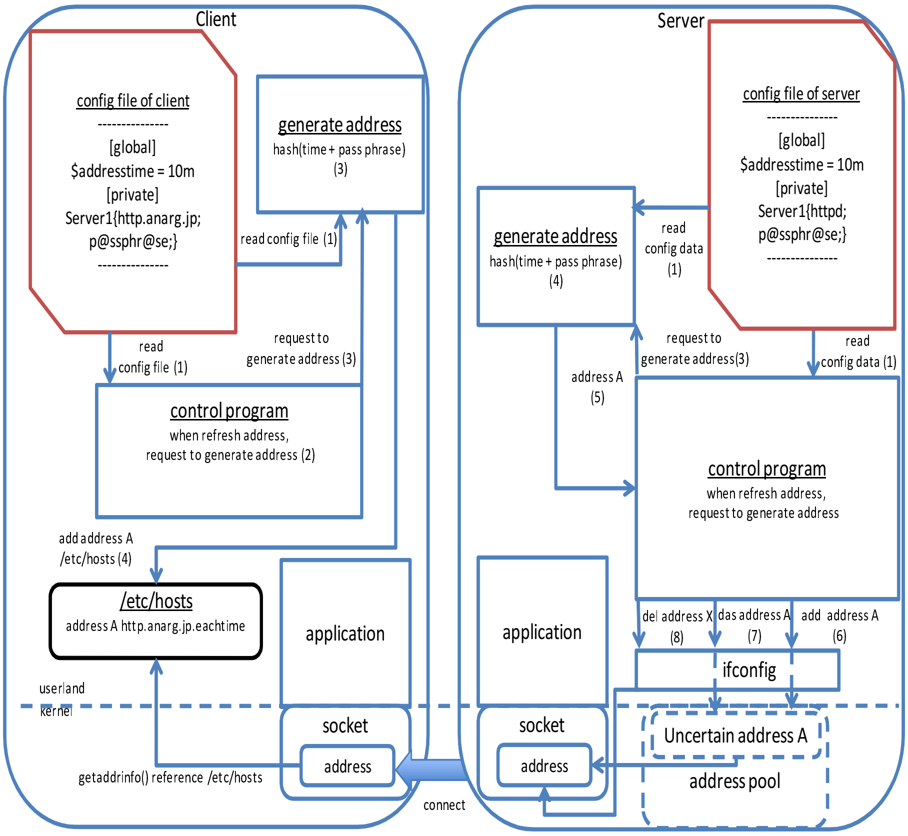


Figure 5: Block diagram of time-driven approach

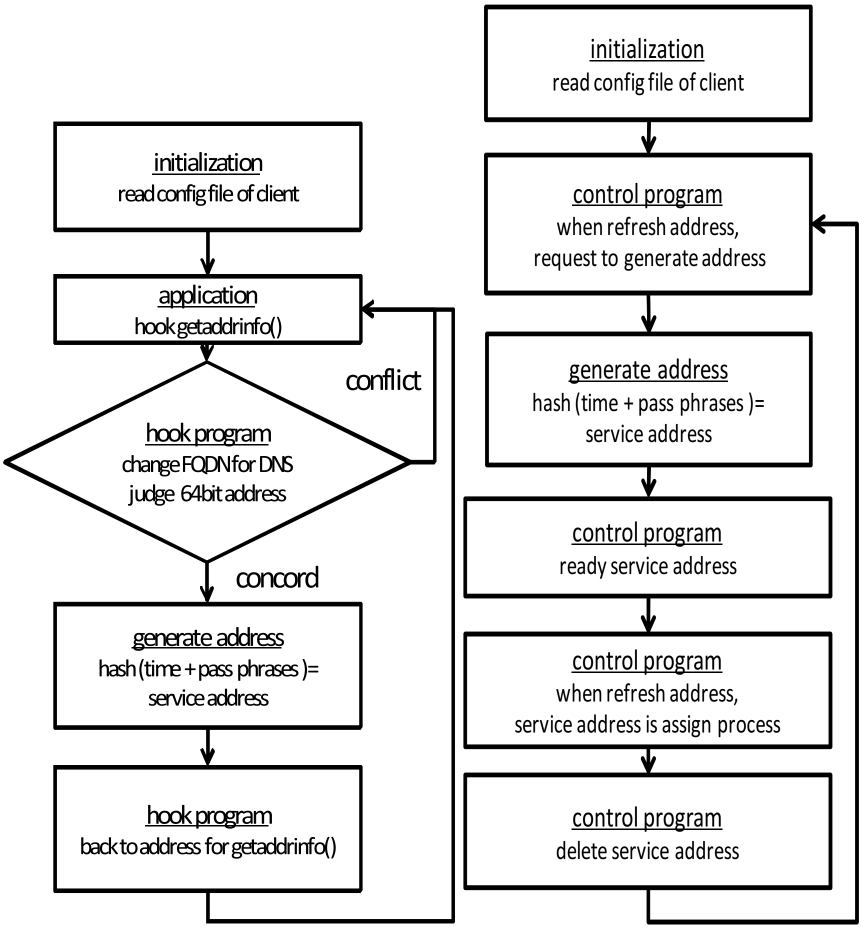
4.2 Approaches for Address Composition

The client works There are two methods to be proposed to explain the behavior of each method, respectively.

4.2.1 Connection-driven Approach

Figure.4、6 shows the connection method for each update type. After the address the client is calculated using the time information and passphrases for listening later determined that the address changes every time the server address resolution when `getaddrinfo()` function return value of the destination socket address information Rewritten. Following the operation of this implementation.

- Reads the configuration file for the client program will address the calculation, to boot. In



(a) Client (b) Server

Figure 6: Flowchart of connection-driven

this case, the configuration file, which describes the continuous standby time in a single address, which clients need to keep shared between servers. The destination server, for generating the address determine a time-based settings for each server process running on IPv6 address lower 64bit, which describes the pass phrase. This is different to listen Different processes for the same destination server. This method is because trying to provide a high level of safety by to listen to a different address in Different listening process even the same server.

- Function is called to perform name resolution for the client user-specified address `getaddrinfo()` hooks, DNS address resolution results using a lower 64bit configuration files are described in the earlier If a match to those that request to the address value calculation program. Kono Tame, `getaddrinfo()` should read the file that the program also sets the hook.

- Lower address value calculation program upon receiving a request to the server 64bit shared passphrase that is configured for each address, the address value calculation based on time information. Time and time information to update the address, which is uniquely determined regardless of the time the program starts to determine the address of standby time. Thus, with just the lower 64bit DNS 64 was calculated using the obtained upper address bit is `getaddrinfo ()` and the destination address of the return value of the socket structure.

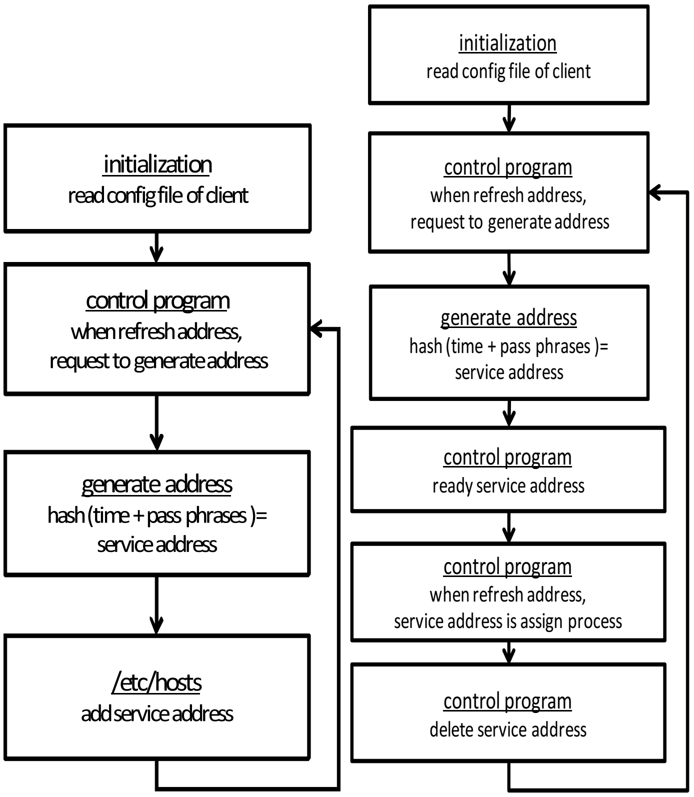
After the address the client is calculated using the time information and passphrases for listening later determined that the address changes every time the server address resolution when `getaddrinfo()` function return value of the destination socket address information rewritten. This approach to calculate the destination address only if the target client user connects to the server, unlike the method described to calculate the destination address is always followed by, say efficient implementation because it minimizes the computational cost. However, `getaddrinfo()` because it connects to a different destination address and destination address obtained in applications such as `getaddrinfo()` as it is when the output value of the log is actually a different destination address is output to the log.

4.2.2 Time-driven Approach

Figure.5,7 shows the hourly update type. The client listens to calculate the address of the server every time changes every time, every time you edit $/etc/hosts$.

Implementation of the type shown below updated every hour. The client listens to calculate the address of the server every time changes every hour, every $/etc/hosts$ edit. In this implementation is not always calculated, even if you do not have to connect to the server. How to implement the following actions.

- Reads the configuration file for the client program will address the calculation, to boot. In this case, the configuration file, which describes the continuous standby time in a single address, which clients need to keep shared between servers. Unlike the previous one and the method, we calculate the destination address in every address update clients on time.



(a) Client (b) Server

Figure 7: Flowchart of time-driven

- Method.1 calculate the address using the update timing information and control programs as well as servers, to keep.
- Address the request to the computer program will address the update time.
- Calculation program calculates the address of the address using the time information and passphrases, and, DNS host to get the advantage of 64bit.
- Address calculated $/etc/hosts$ add an entry with the server.
- When the correct address information to connect to the address that is updated every time a client $/etc/hosts$

4.3 Detailed Implementation

The following update procedure for each connection type implementations. Client-side implementation.

- Config file of client

  ```
  [global]
  $Address_Refresh_Time=200
  [private]
  ffff::8080;p@ssphr@se,
  ffff::22;pAssw0rd,
  ```

- Hook application
  - Initialize
    1. Read config data
  - Loop
    1. Hook `getaddrinfo()`,FQDN for DNS address resolution results, determine the need for lower-value calculation 64bit address.
    2. If we need to address value calculation, `gettimeofday` time information, the mod takes the address of standby time, and time will update the address 0, IPv6 address of the computer program to pass the address 128bit.
    3. Listening to the return address from the address of the computer program `getaddrinfo()` return as a return value.
- Generate address
  - Initialize
    1. Read config data
  - Loop

1. Get IPv6 address and address refresh time from hook application
2. Obtained in the above time and address updates 64bit 128bit lower bound matches the passphrase.
3. During the high value of the SHA-1 64bit use.
4. Top uses a 64bit server address obtained from a hook program, the lower the calculated address to listen 64bit server in conjunction.
5. Listen address to return the hook program.

We discuss the implementation of server'side as follow.

- Config file of server

  ```
  [global]
  Address_Refresh_Time=200
  [private]
  httpd http.anarg.jp p@ssphr@se
  sshd ssh.anarg.jp pAssw0rD
  ```

- Control program
  - Initialize
    1. Read config data
  - Loop
    1. Time information on `gettimeofday`, the mod takes standby time address, the time will hold the information.
    2. 2s address update time passes the updated time address before the address of the computer program, called.
    3. 64bit address of the computer program obtained from the address of the FQDN of the upper 64bit address to be coupled.
    4. Available state to address the resulting `uncertain` issues a command to prepare the address.

    5. At the time address updates `das address` issuing commands to add addresses to the process.
    6. Address was added to the process control program to keep information.
    7. Time has passed to address valid address is `del address` commands to delete.

- Generate address
  - Initialize
    1. Read config file.
  - Loop
    1. Get information from control program.
    2. Add address refresh time and pass phrase.
    3. Hash the value.
    4. Return the value.

We discuss these steps of time-driven approach.Client-side.

- Config file of client.

```
[global]
Address_Refresh_Time=200
[private]
http.anarg.jp p@ssphr@se
http.ist.osaka-u.ac.jp pAssw0rd
```

- Control program
  - Initialize
    1. Read config file.
  - Loop

1. Time information on `gettimeofday`, the mod takes standby time address, the time will hold the information.
2. 2s address update time passes the updated time address before the address of the computer program, called.

- Generate address
  - Initialize
    1. Read config file.
  - Loop
    1. Get information from control program.
    2. Add address refresh time and pass phrase.
    3. Hash the value.
    4. Obtain DNS server address is the top referring to 64bit, 64bit and just calculated in conjunction with the address of the server listens.
    5. When refresh address,edit $/etc/hosts$.
- $/etc/hosts$

```
2001:380:500d:140:xxxx:xxxx:xxxx:xxxx http.anarg.jp
2001:2f8:23:110::xxxx:xxxx:xxxx:xxxx http.ist.osaka-u.ac.jp
```

Server'side is same connection-driven approach.

5 Validation and Numerical Evaluation

This chapter describes about the adequacy of this method.

5.1 Architectural Validation

I show how this suggestion technique can be satisfied with two chapters for the problem that I spoke. The anonymity which I spoke in two chapters is as follows. I speak how much can satisfy these.

- **Uni-directionality:** The server can not connect to the client using the client's source address in the packet originated the clinet. Since this states the anonymity of the clinet, we do not describe in detail in this thesis.
- **Discontinuity:** By grouping the same addresses recorded in the communication session, multiple communications are associated so that it is difficult to guess the behavior of the node.Unified Multiplex listen address of the server by using this method will be able to take seemingly random values. This grouping could not address value, these conditions are met.
- **Dissimilarity:** Even if the server address is known to a third party the server can not be communicated using the address or some other addresses inferred from the original address.By changing the address of the server that can be achieved by Unified Multiplex used. The method is considered viable by this method to notify the client that address. So, say that these conditions are met.

5.2 Numerical Evaluation

In this proposed method is 216 seconds and the effective time of the server listen address. Time to examine the validity of the address for this stand.

It is a pair from Figure8 to 10000-per second random selection attack for these 216 seconds. It is thought that it has security strength that can endure practical use because it can suppress the collision rate to 0.000001% or less by doing. Moreover, it is usually 32 at constant intervals of 60 seconds based on the random numbers key generated when it clocks and the factory is shipped building into RSA SecureID [22] [23] that is the one-time password attestation mechanism that uses the token that generates the validation code of bit is introduced into the enterprise that will

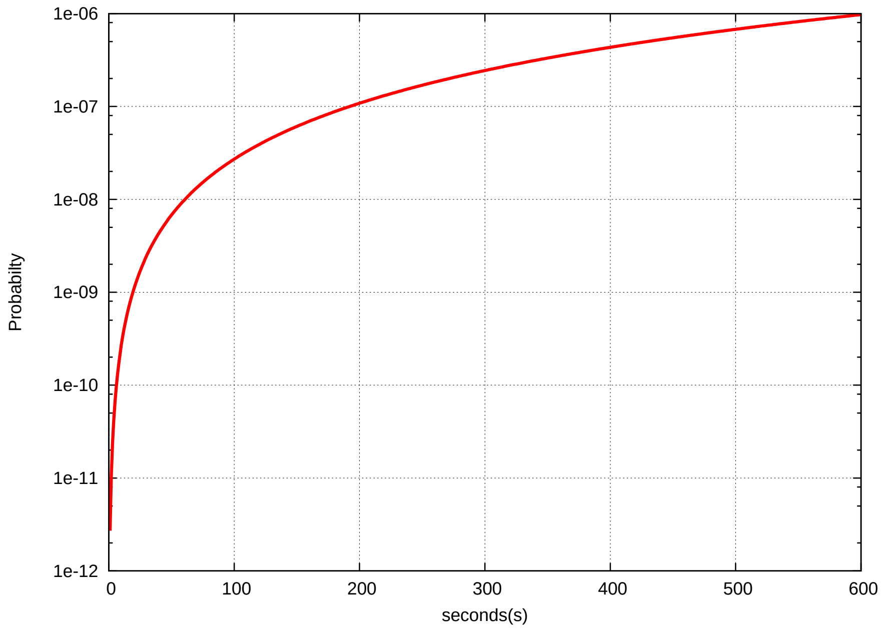


Figure 8: Collision probability at a random selection

exceed 20,000 by present. It uses it by this proposal technique, and changing information is bit row twice 32bit subordinate position 64bit of IPv6, and is thought to have high security strength enough even if it keeps using information on 64bit for 216 seconds because it becomes 2^{32}twice as volume of information when thinking the number of reported cases of the security breach is 0 for 15 years. Therefore, it is thought this waiting address effective time is appropriate.

6 Conclusion and Future Work

In this thesis, it proposed the mechanism that was able to be peeled off to practical development of Unified Multiplex. Whether it notifies becomes important for the client ..waiting address of the server.. very because it changes it in Unified Multiplex in each session of the address. This time, the waiting address of this server was assumed to be a method of notifying the client and it proposed the mechanism of the notification of the address of non-negotiation type. Because this technique shares the passphrase between the client and the server beforehand, the client becomes possible to connect with the server without acquiring the waiting address of the server whenever connecting to the server it. It designed to achieve this proposal technique, and the mounting specification was settled on.

Acknowledgements

First, I would like to express my sincere gratitude to my supervisor, Professor Masayuki Murata of Osaka University, for his continuous support and expensive advice throughout my studies, and providing me this precious study opportunity in his laboratory.

I especially would like to express my deepest appreciation to Associate Professor Shingo Ata of Osaka City University, for his elaborated guidance and invaluable firsthand advice. All works of this thesis would not been possible without his support.

My deepest thanks also to Dr. Hiroshi Kitamura of NEC Corporation who I sincerely wish to thank for his advice and assistance. He made an effort for acceptance of the internship to NEC, and did instruction which cannot be received in school education called mental attitude as a member of society.

Furthermore, I must acknowledge Associate Professor Naoki Wakamiya, Assistant Professor Shin'ichi Arakawa, and Assistant Professor Yuichi Ohsita of Osaka University for their valuable comments and suggestions on this study.

I heartily thank Ms. Morimoto, secretary's doing a lot of help, and the encouragement at times.

There was help of Ms. Haesung Hwang, so I am able to complete this article. I show will of thanks to here.

References

[1] G. Huston, "IPv4 Address Report." available at `http://www.potaroo.net/tools/ipv4/`.

[2] K.Nishida, S.Ata, H.Kitamura, and M.Murata, "An Unified Multiplex Communication Architecture for Simple Security Enhancements in IPv6 Communications," *EuroView*, Aug. 2010.

[3] S.Ata, H.Kitamura, and M.Murata, "Architectural Design of Unified Multiplex Communications for One-Time Use of IP Addresses," *in proseedings of NTMS*, Feb. 2011.

[4] H. Kitamura, S. Ata, and M. Murata, "IPv6 ephemeral addresses," *Internet-Draft, `draft-kitamura-ipv6-ephemeral-address-00`*, Oct. 2008. work in progress.

[5] F. Armknecht, J. Girão, A. Matos, and R. L. Aguiar, "Who said that? privacy at link layer," in *Procedings of INFOCOM*, pp. 2521–2525, May 2007.

[6] P. Srisuresh and M. Holdrege, "IP Network Address Translator (NAT) terminology and considerations," *RFC2663*, Aug. 1999.

[7] G. V. de Velde, T. Hain, R. Droms, B. Carpenter, and E. Klein, "Local network protection for IPv6," *RFC4864*, May 2007.

[8] M. J. Freedman and R. Morris, "Tarzan: A peer-to-peer anonymizing network layer," in *Proceedings of the 9th ACM (CCS 9)*, pp. 193–206, Nov. 2002.

[9] M. J. Freedman, E. Sit, J. Cates, and R. Morris, "Introducing tarzan, a peer-to-peer anonymizing network layer," in *Proceedings of the IPTPS*, Mar. 2002.

[10] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, pp. 66–92, Nov. 1998.

[11] H. T. Kung, C.-M. Cheng, K.-S. Tan, and S. Bradner, "Design and analysis of an IP-layer anonymizing infrastructure," in *Proceedings of the DISCEX-III*, pp. 62–75, Apr. 2003.

[12] C.-M. Cheng, H. T. Kung, K.-S. Tan, and S. Bradner, "Anon: An IP-layer anonymizing infrastructure," in *Proceedings of the DISCEX-III*, pp. 78–80, Apr. 2003.

[13] R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, pp. 303–320, Aug. 2004.

[14] S. Katti, J. Cohen, and D. Katabi, "Information slicing: Anonymity using unreliable overlays," in *Proceedings of the NSDI*, pp. 43–56, Apr. 2007.

[15] D. McCoy, K. S. Bauer, D. Grunwald, T. Kohno, and D. C. Sicker, "Shining light in dark places: Understanding the tor network," in *Proceedings of the 8th International Symposium on PETS*, pp. 63–76, July 2008.

[16] K. Shima and H. Kitamura, "IPv6 global communication architecture build on a mechanism for conveying service dedicated address information (DNSO)," *IEICE Technical Report (IN2008-29)*, vol. 108, pp. 17–22, Mar. 2008.

[17] B. Schneier, "Attack trees," *Dr. Dobb ' s journal*, vol. 24, no. 12, pp. 21–29, 1999.

[18] S. Thomson, T. Narten, and B. Jinmei, "RFC 4862: IPv6 stateless address autoconfiguration," Sept. 2007.

[19] J. Philip and P. Erdelsky, "The birthday paradox," *EFG, at http://www. efgh. com/math/birthday. htm*, Mar. 2002.

[20] G. Xie, C. Irvine, and T. Levin, "Quantifying effect of network latency and clock drift on time-driven key sequencing," in *Proceedings of ADSN*, pp. 35–42, IEEE, Nov. 2002.

[21] G. Wright and W. Stevens, "TCP/IP illustrated, volume 2," 1995.

[22] R. SecureID, "Secure Identity." available at `http://www.rsa.com/node.aspx?id=1156`.

[23] A. Biryukov, J. Lano, and B. Preneel, "Cryptanalysis of the alleged SecurID hash function," in *Selected Areas in Cryptography*, pp. 130–144, Springer, May 2004.