A study on VoIP Security

Chia-Ming Sung, Ching-Cheng Lo, Chung-Hao Peng, Wen-Nung Tsai
Department of Computer Science and Information Engineering,
National Chiao-Tung University
{*chiaming, locc, thpeng, tsaiwn*}*@csie.nctu.edu.tw*

Abstract
In this paper, we proposed a method of secure transmission of voice data in public IP network. Traditional methods used for secure transmission, such as encryption, take up too much time in data encryption/decryption and key processing. We use visual cryptography and data sharing methods to reduce the effects caused by those problems. In addition, we utilize the disjoint multi-path routing algorithm to send voice data, so that we can reduce the occurrence of network hot-spot and decrease the transmission delays.

Keywords: Data sharing, Disjoint multi-path routing, SIP, visual cryptography, VoIP.

1. Introduction

VoIP [1] is an attractive real-time application for Internet service providers. The secure transmission of those voice data is one of the most important issues. We proposed an approach that can ensure the security of voice data and the efficiency of transmission. Although there are already many methods for the application of non-real-time transmission, they may take up much more processing time, especially for cryptography systems requiring higher security. And thus those methods are not efficient enough for real-time application, especially on a small hand-held device.

Applying Data sharing and Visual cryptography [6] to data processing can protect the transmission from being hacked. Compare to the traditional encryption and decryption, it also omits the steps of key establishment, key exchange and key management. Hence, we would like to establish a system model that is more suitable for real-time transmission.

From former study [2], Dynamic multi-path routing can not only reduce network congestion, but also decrease the transmission delays, which is suitable for real-time application data.

The system we proposed is based on the network SIP VoIP phone system. We use some voice codec to develop our data sharing method. We also utilize the disjoint multipath routing algorithm to send data, in order to achieve the ideal of efficient and secure transmission.

2. VoIP (Voice over IP)

Voice over IP, which digitalizes voice into forms of packets, and then transmits them over an IP (Internet Protocol) network to the receiver, can be more adequate in making use of the internet resources comparing to the former Circuit-Switch. This not only can process more telecommunication connections at the same time, but also reduce the cost, so it is becoming more and more popular in the business field. In this paper, the VoIP is based on SIP, which is overviewed in the following.

2.1. Session Initial Protocol (SIP)

Session Initial Protocol, abbreviated as SIP[10]，was proposed in 1999 by the IETF to be an application layer protocol. Its main purpose is to establish, update, and/or hand off multimedia communications. Now, we will introduce SIP according to its architecture and message exchanged.

SIP is a client-server model, the client sends a request, and after receiving it, the server will return an appropriate response. Here are the entities of the SIP architecture:

(1)**Client**：Also known as User Agent Client, is an application, and sends the SIP message to the server or other clients.

(2)**Server**：The main function is to send back a correct response to the client. It can be classified by functions into four categories:

1. *Proxy server*: It transmits the request from the client to the correct server or user. It can be stateful or stateless.
2. *Redirect server*: According to the records in the database of the server, it finds the address which corresponds to the request sent by the client, and returns the address to the client.
3. *Register server*: It allows the user to register on it after it receives the message sent from the client. The registered data includes the user's address, online-name, etc.
4. *User agent server*: Receives the request from the user, and then makes a respond by sending the request to a proper server. It is usually built together with a User Agent Client.

2.2. Voice codec

In order to transmit voice over the IP-network, analog signals must be transformed into digital signals, which require a transforming method.

To make transmission of the voice packet over the Internet more convenient, there should be a limit on the usage of bandwidth, and the quality of the voice must also be considered. ITU-T brought out the idea of MOS (Mean Option Score)[13], which scales the voice quality into 5 levels (1~5, with MOS-5 as the best and MOS-1 as the worst). The standard now used is to calculate the mean of the grade level (according to MOS) given by at least 30 people whom have listened to the voice codec. (Voice codec that can be commercialized are mostly required to be above the level of MOS-4.)

Voice Codecs which are often used in real-time applications are defined in the ITU-T of the G7xx series. The Codecs used in our research are G.711 [11] and G.726 [12].

3. Related Works

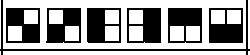
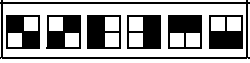
In this section we will bring in some related researches. We will start with Visual cryptography. And lastly will be the relative work of this paper: Disjoint Multi-path Routing Algorithm.

3.1. Visual Cryptography

Visual cryptography [6], proposed by Moni Naor and Adi Shamir in 1995, encrypts documents (writings, notes, pictures, etc.) according to the feature of human vision, and then decrypts the documents by plane human vision system. This kind of cryptography only requires one cipthertext (transmitted by fax or mail delivery) and one printed slide (as the Key). When the receiver puts the slide on top of the ciphertext, he/she will be able to recognize the plaintext (which may contain some random noise). This is a simple and quick method that can encrypt and decrypt the message without complicated processes.

Due to the drawbacks of easily recognizing the plaintext from the Share data, investigators expand each of the 1 pixels on the plaintext into 4 pixels on the Share data. The corresponding table is shown in Table 1.

Table 1. Visual cryptography, which divides each pixel into 4 bits

Image	Secret pixel (white)	Secret pixel (black)
share1		
share2		
stacking result		

Thus the share data would be recognized as noise, and the security of the transmitting documents would then be enhanced. The disadvantage is that it would take up a space four times the plaintext to save to share data (Figure 1).

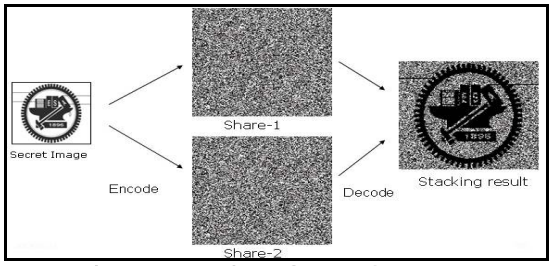


Figure 1. Example for VC with 4 bit/share pixel

3.2. Disjoint Multi-Path Routing Algorithm

In concepts of Share Data, two different Share data must be transmitted on two different paths, if not, the Share data may be intercepted and then recovered into the plaintext, and lose the security of Share data. In addition, according to other studies [2],[3],[4], Dynamic multi-path routing is able to reduce network congestion, and decrease the transmission delays, which is suitable for those time-sensitive applications, such as VoIP, multimedia stream, etc.

Many algorithms have already been proposed to find the Routing path for the Disjoint-set. The related studies [6], [8], [9] have its advantage.

Before introducing the algorithm, we will first explain some notations of Network topology. Network architecture can be modeled into an undirected graph: G=(V, E). Where V stands for a limited set of nodes, E stands for a set of limited links. A link in set E connects a pair of nodes, x and y, written as (x,y). The cost for (x,y) is written as c(x,y). c(x,y) is an integer and c(x,y)=c(y,x).

From set V, give a node t as the destination node. Then give a path **p** that goes from **s** to **t**, (s,i,j, ⋯k,t), which is composed of a group of conjunct link sets (s,i),(i,j),⋯,(k,t). The cost of path **p** is c(**p**), the cost of the sum of going through the all links between it. If there is a path **p**, (s,x,⋯,y,t), then its path id is defined as pid(p)=y. In G there exists a shortest path tree (SPT) with t as its root. The cost of the path from every node in the SPT to the node t is minimized if the path is traced along the SPT. SPT(x,t) means that it is the only path from x to t. If y is a node on STP(x,t), then y is a downtree of x, and x is an uptree of y. As for a node x, if there exists a link (x,y), then y is a neighbor of x. All neighbors of x are written as nbr(x). If y is an uptree of x and $y \in nbr(x)$, then y is the uptree neighbor of x. If y is a downtree of x and $y \in nbr(x)$, then y is the parent of x. If $y \in nbr(x)$ and (x,y) is not a link of the SPT, then y is a horizontal neighbor of x.

It assumes in [8], that all nodes are given its uptree neighbor, parent, and horizontal neighbor, and the topology is not changed in the process. The structure used for message transmission is shown in the following:

msg{mytype, nid, pid, cst, path}: ***mytype*** stands for message type, which can be 0 or 1; ***nid*** is the node id that sent the message; ***pid*** stands for path id; ***cst*** means the cost of passing through the path; ***path*** is the set of all nodes that have been passed through.

There are two steps in finding the algorithm of this kind of disjoint multi-path routing. The first step is to send a message with message type 0, ***msg{0, t, Φ, 0, (t)}***, from the destination node ***t*** to its neighbors. After any node ***x*** in the internet receives a message ***msg{0, nid, pid, cst, path}***, there could be two possibilities:

1. If the message was sent from the parent of ***x***, then ***x*** will be added into the ***path***. After altering the message to ***msg(0, x, (pid==Φ)?x:pid, cst+c(x,parent(x)), path+x)***, ***x*** sends the message to its uptree and horizontal neighbors.
2. If the message was sent from the horizontal neighbors of ***x***, check to see if the ***path*** will overlap with an already given path with ***x*** on it after adding ***x*** to the ***path***. If not, then include it to the transmittable paths. This message could not be sent to the external.

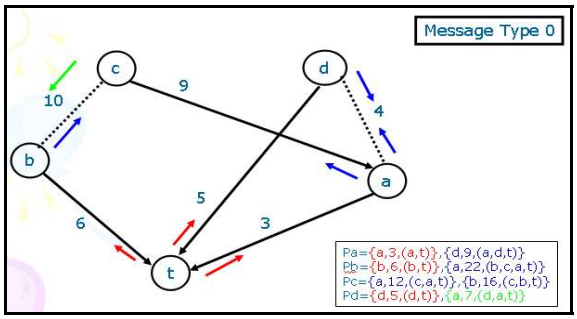


Figure 2. Disjoint multi-path routing with message type 0

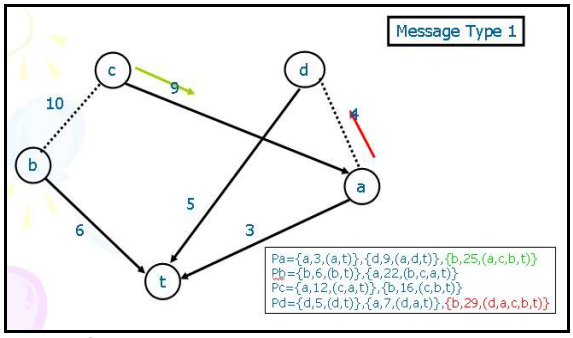


Figure 3. Disjoint multi-path routing with message type 1

The second step is to acquire more sets of disjoint multipath routing by exchanging the message from the message type 1. The message in the message type 1 is sent from each node. As for the alternative path recorded by a node ***x***, ***x*** will produce a message ***msg{1, x, p.pid, p.cst, p}*** and send it to the appropriate neighbors. Appropriate neighbors must follow three conditions: (1) $v \in nbr(x)$; (2) $v \notin p$; (3) ***v*** must not be an uptree neighbor of ***x***. In this way, we can obtain sets of disjoint multipath routing after transmitting, as shown in Figure 2, Figure 3.

4. System Architecture

In this section, we will introduce the system architecture integrating visual encryption concepts and SIP phone. First we briefly introduce the system architecture and some assumptions, then we show the usage of the system and specific signal exchange processes, and lastly we introduce the methods for segregation and reconstruction of voice packets.

This system inherits basically from SIP, using a Client-Server model. To maximize compatibility with current network architecture, we made some changes to the original server functions, as shown in Figure 4.

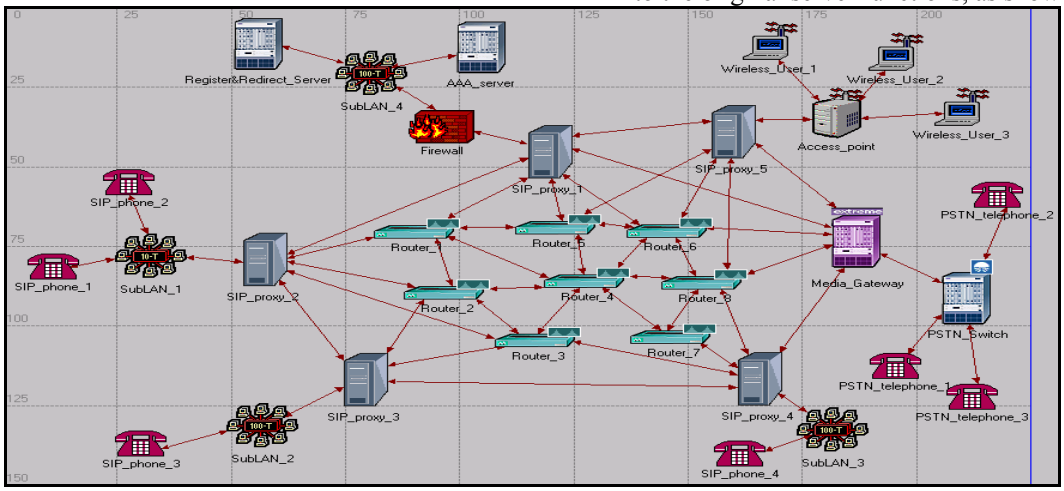


Figure 4. System architecture

In our approach, taking into account that most of the routers currently in use are using the Open Shortest Path First (OSPF) routing protocol, changing everything to use Disjoint multi-path routing won't be plausible and will be very costly. We therefore separate the SIP proxy servers, and expand their functions to more than just the original packet-switching function. This will be covered in

the next section.

4.1. System Assumption

In addition to the changes in the SIP proxy server, this system has some preset conditions and restrictions, listed below:

- There will be no safety mechanics of communication within a LAN. Data segregation on voice is only used when transmitting data across different LAN networks.
- Assume that there are more then two disjoint paths between two nodes.
- Assume that the SIP proxy server can receive most of routing information from nearby routers, thus allowing the SIP proxy server to make a better path choice.
- Assume that if the data goes through a firewall or a Network Address Translator (NAT), there will be a corresponding solution like UDP hole punching.

4.2. SIP Proxy Server

In our planned VoIP security measures, the SIP proxy server will have the greater changes, adding in the following functions:

- Disjoint Multi-path Routing. We implement this on the SIP proxy servers, instead of on the routers.
- Voice Data Segregation. We segregate the voice packets into two shares, and then transmit them.
- Gateway to the PSTN network. The parts linking with PSTN network can be integrated with the original Media gateway, achieving the functions of a SIP proxy server and a PSTN media gateway.
- SIP user management in a region. This will record communication time and user management information, including registration information and accounting information.

Then, we introduce ways used by the SIP proxy server to handle all kinds of incoming data:

- When the SIP proxy server receives a SIP message, everything will go through the normal forwarding actions except for the register data, which will be introduced in the next section.
- After the SIP proxy server receives routing information from general routers, it obtains a current routing table, which facilitates the later process in disjoint multi-path routing.
- After the SIP proxy server receives messages of disjoint multi-path routing, it will proceed with the corresponding algorithm similar to the one in section 3.2. It will be covered in section 4.4 later.
- After SIP proxy receives a RTP packet, it will make a series of processes:

  1. Check to see whether the source of the RTP packet is a SIP phone controlled by its SIP proxy server or not. If yes, go on to step 2. If not, step 3.

  2. Pack the data into two RTP packets with the same header of the original packet. Then acquire information of disjoint multi-path routing. Take the two shortest paths, and write it individually into the IP header of two Share packets. Last, send the packet according to the path, and leave the process.

  3. Check to see if the destination address of the RTP packet is an SIP phone controlled by its SIP proxy server. If yes, then check the temporary saver and see if another Share date already exists. If it does, combine it and send it to the SIP phone. If it does not, then save it in the temporary saver. If the destination address is not an SIP phone controlled by its SIP proxy server, send the data to the next destination, according to the IP header,.

  4. In step3, if the connection of the next destination and the SIP proxy server is broken off, obtain a disjoint path for transmission after comparing the routing information of the SIP proxy server and the option field or Hop by Hop field in the RTP header.

4.3. System Processing and Message Exchange

Afterwards, there may be different kinds of situations. According to different scenarios, we will now introduce the process of massage interchange and transmission. To send messages in a communication is similar to the process of message transmission in the original SIP. The biggest difference is the transmission media.

When an SIP phone is connected with the Internet, it sends the Register message to the Register server over the proxy server.

Scenario 1 –User register into the system

Step 1: SIP_phone_1 sends the message of register to SIP_proxy_2.

Step 2: After SIP_proxy_2 receives the message of register, it temporarily saves the user's data. Then it sends the register message of SIP_phone_1 to the register server.

Step 3: After the register server passes the confirmation, and records the data of SIP_phone_1 user and its SIP_proxy_2, it sends an OK message to SIP_proxy_2.

Step 4: When SIP_proxy_2 receives the OK message, it moves SIP_phone_1 data from the temporary saver to a regular one, which makes the establishing of a communication in the future more convenient. Then it sends the OK message to SIP_phone_1, and the register finishes its process.

Step 5： This is an optional step. In order to protect personal Internet transmission security, we can use a simple cryptography between the SIP phone and the SIP proxy (such as XOR plaintext with Key). For such communication, the two sides can discus whether there is a need of the key or not.

Scenario 2 –Establish call in local area network

As mentioned in the former section (4.1), the SIP call security within a LAN will not be covered in this paper since we assume it is secured in a LAN.

Scenario 3 –Establish call across network

Step 1: If SIP_phone_1 wants to establish a call with SIP_phone_3, it will send a request to query the address of SIP_phone_3. When SIP_phoxy_2 receives the message, it can't find related data of SIP_phone_3 in the SIP_phoxy_2 record table. So it turns to the Register&Redirect_server to search. The Register&Redirect_server would then send back the search result to SIP_proxy_2 and SIP_phone_1.

Step 2: According to the search results, SIP_phone_1 sends an INVITE message to SIP_phone_3. The communication becomes established after SIP_phone_3 sends back a Ringing and an OK message. And then SIP_proxy_2 starts to record the related communication data. (Normally, the Ringing and OK messages would be sent separated. The Invite sends OK message after the receiver answers the Ringing message sent previously.)

Step 3: After the communication begins, when SIP_proxy_2 receives a voice packet from SIP_phone_1, it would divide the packet into two parts. Then according to the current condition of the Internet (resulted from adjusting the information received by the router), and using the two shortest path obtained from the Disjoint multi-path routing algorithm presented in section 3, it transmits these two packets separately to SIP_proxy_4. The packets are recovered in SIP_proxy_4, and then sent to SIP_phone_3.

Step 4: When a BYE message is sent from either end, the end that receives the BYE message will reply an OK message and the communication ends. The SIP_proxy_2 will then record the starting and ending time of the communication to facilitate queries of IDs and records from the telecommunication company.

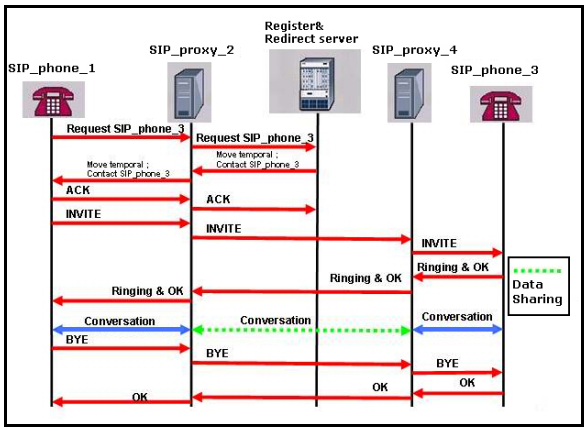


Figure 5. Scenario 3 in SIP VoIP

Scenario 4 – Establish call between IP network and PSTN network

In this scenario, the message transmission is similar as that in scenario 3. The difference is, in step 3 of scenario 3, the INVITE message of communicating with SIP_proxy_4 must be sent to the Media Gateway Controllers (MGCs), and communicate with PSTN switch over the MGC to establish a call. It also reserves the resources on the Media Gateway Controllers (MGCs) for this call. Then, when transmitting the data of the voice packet, SIP_proxy_2 also segregates the packet. As proceeding, the functions of the SIP proxy server, Media Gateway recovers the packet and transmits it on the PSTN network.

4.4. Voice Packet Segregation

To minimize the increase in data space, the easiest way is to only segregate two bits from one bit of the original data. We only need to consider the fair operations used in the segregating.

To make the probability of 0s and 1s both one-half respectively, we use XOR operation. The data segregating method is very straightforward. When the original data is 0, the corresponding bits of share data 1 and share data 2 can be either 0-0 or 1-1; when the original data is 1, the corresponding bits in share data 1 and share data 2 can be either 1-0 or 0-1. If a sniffer gets only one share of the data, the chance for the original data to be 0 or 1 is both ½.

If we segregate all bits of the sound packet using the above method, we will get two share of the same size as the original packet, thus increasing the size by twofold. We wish to use different techniques according to different sound coding, in order to lessen the size increase. Now we use the encoding methods of G.711 as an example, using the data segregating methods shown in figure 7. Only the first four bits in every byte of the original date are segregated, achieving a size increase of 1.5 times of the original one. (Similar methods can also be used in the encoding methods of G.726 32Kbps, just take two bits ahead of the original segregating method.)

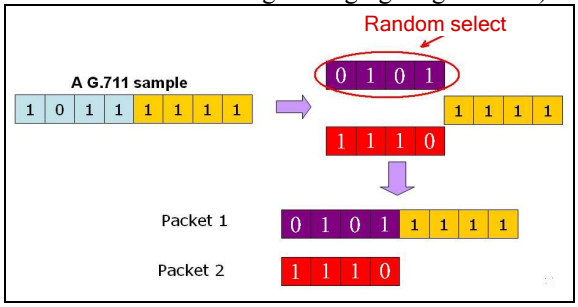


Figure 6. Data sharing method by example

As shown in Figure 6, the left most 4-bit data is XORed with a 4-bit random number to form packet 2. (4 bits only.) And then replace the left most 4-bit of the original one with that random number to form another share packet 1.

4.5. Disjoint Multipath algorithm

In this section, we will use some examples to show you how to find disjoint multi-paths. The network topologies used are shown in Figure 7,Figure 8, and Figure 9. The procedure includes 3 steps.

Step 1：Every SIP proxy server tries to obtain enough information from routers. Then it will find out all the intermediate routers and calculate the costs to the destination.

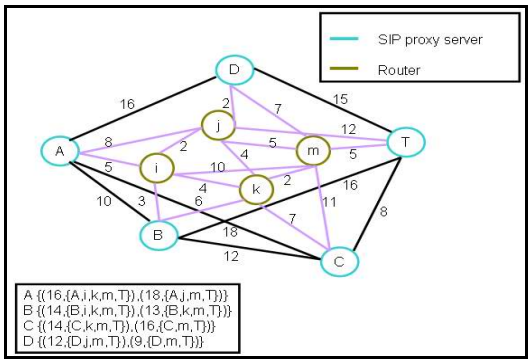


Figure 7. Disjoint multi-path routing step 1

Step 2：As mentioned in section 3, SIP proxy server T sends message type-0 to its neighbor SIP proxy server. Then the neighbor proxy server will try to figure out whether a new path to T exists or not, as shown in Figure 8.

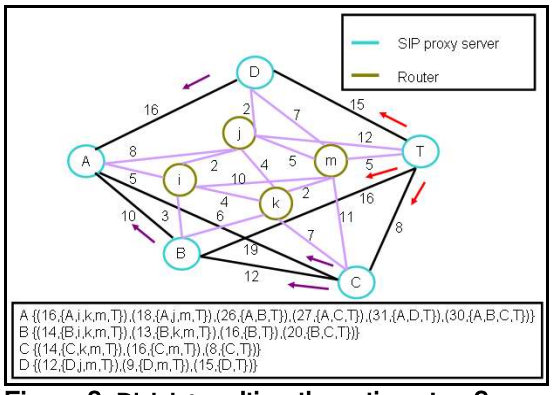


Figure 8. Disjoint multi-path routing step 2

Step 3：In this step, SIP proxy server will decide if it should send message type-1 to its neighbor nodes.

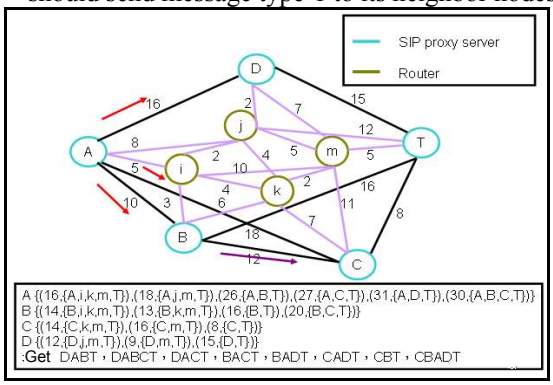


Figure 9. Disjoint multi-path routing step 3

After these 3 steps, the SIP proxy server will get its disjoint multi-path to each of its destination as long as the proxy server can obtain enough routing information. Consider RIP(Routing information protocol) as an example, routing information can be traced back to its previous 16 hops. Other outing protocols allow more hops.

5. Simulation

In this section, we will show some experiments to simulate the methods described formerly. We will illustrate the simulation environment, the reliability of voice segregation, comparisons on time handling with other cryptography methods, and simulations of data transmission over the network.

5.1. Environment

Before introducing our simulations, we will first state the simulation environment used. The hardware for the simulation contains a notebook with a Pentium-M 1.8 GHz CPU, and a 1GB sized SDRAM. The software includes Microsoft Windows 2000 server, Visual C++, and some other software that would be shown in the following.

5.2. Experiments on Voice Segregation

In this section we will simulate the security of the voice segregation method mentioned in section 4. We want to verify that it is unable to recognize the original voice content from only one of the Share packet. We use the recording software, Vox Studio 3, to record a 5-second G.711 A-law voice. The wave of the original voices is shown in the Figure 10.

After using the algorithm proposed in section 4, the waves of Share data 1 and Share data2 (Figure 11, Figure 12) are similar to regular noise, and the content cannot be recognized.

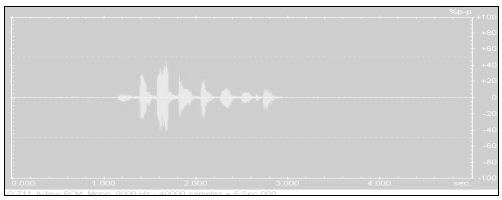


Figure 10. Voice sample with G.711 A-law (5 seconds)

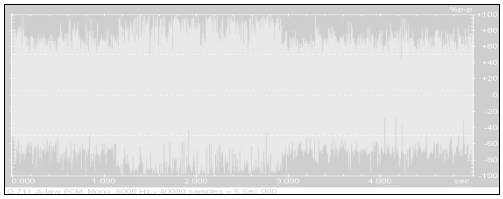


Figure 11. Share data 1

The end that receives Share data 1 and Share data 2 will recover the original voice data after it applies

the XOR operation on all the Samples of Share data 1 with the four corresponding bits in Share data 2 (see Figure 13).

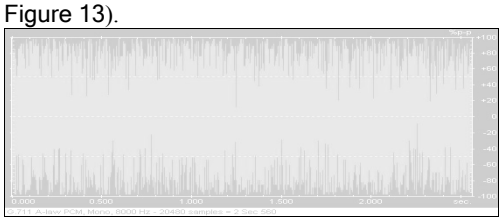


Figure 12. Share data 2

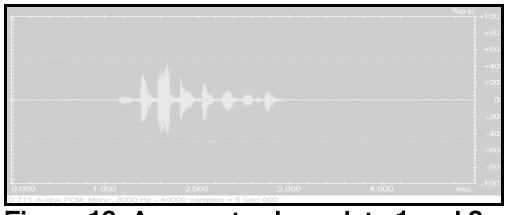


Figure 13. Aggregate share data 1 and 2

If Share data 1 is intercepted, the last four bits remain as the same as those in the original voice packet. So we take off the last four bits of the original packet and fill it in with four 0s to see if it would be able to recover the original voice. The experiments show that we are able to recognize the parts of the human voice but unable to recognize the content (Figure 14).

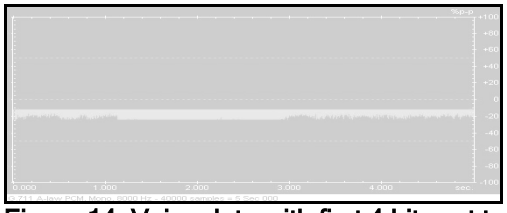


Figure 14. Voice data with first 4 bits set to 0

5.3. Comparisons on Handling time between Different Cryptographic Methods

According to the feature of the VoIP transmission, we know that if we reduce the time on handling with the packets on the two ends, we can improve the voice quality. If we can decrease the time spent in the data packet processing, there will be quite an effect on the quality of the transmission.

We choose the methods of voice codec frequently used and the corresponding size of the data that needs to be encrypted, and then compute the time needed in encryption and decryption. The following table shows the software environment used in the simulation. The environmental parameters used for comparing cryptographies are listed in Table3.

Table 2. Simulation environment

Encryption software	KRYPTOS v1.0, one of GNU educational software toolset
Complier	Microsoft Visual C++ 6.0
Encryption data	40 Kbytes G.711 A-law sample

Table 3. Environmental parameters

Encryption	Key size	Mode	IV size
DES	64 bits	CBC	64 bit
Triple DES	128 bits	CBC	64 bits
Rijndael	128 bits	CBC	128 bits
RC6	128 bits	CBC	128 bits
RSA	Public:1280 bits Private:5064 bits	CBC	2048 bits

Table4 shows the average time required to encrypt/decrypt 1000 data packets of the same voice packet, using different cryptographies.

Table 4. Encryption/decryption time for 40 bytes

Algorithm	DES	3DES	Rijndael	RC 6	RSA	Data sharing
Average enc Time (ms)	9.16	16.45	4.90	8.40	431.22	1.25
Average dec Time (ms)	9.70	17.30	5.33	7.00	14019.04	0.25

In order to be more close to the real world for VoIP, we preset the packet size of the G.711 as 160 bytes according to the data processed in G.711. After taking 160 bytes out from the original voice packet, then do the same simulation under the environment listed in Table 5, using same parameters of cryptographies listed in Table 3, and then compare the results with the previous simulations with the same environment. Table 6 shows the results.

Table 5. data sharing environment -2

Encryption software	KRYPTOS v1.0, one of GNU educational software toolset
Complier	Microsoft Visual C++ 6.0
Encryption data	160 bytes G.711 A-law sample

Table 6. Time to Encrypt/Decrypt 160bytes

Algorithm	DES	3DES	Rijndael	RC 6	RSA	Data sharing
Average enc Time (μs)	99	160	74	71	11121	6.0
Average dec Time (μs)	101	160	71	74	140515	2.8

According to the simulation results, the encoding/decoding time used in the data sharing method is much less than encryption/description time used in all the cryptography methods. Since there is less Mouth-to-Ear delay time, the voice quality would be better.

5.4. Network Delay Simulation

Besides the encoding/decoding time, the packet delay is incurred most by the transmission delay on the network. Because we need an alternative path to transmit share data 2 of the voice packet, and this path is obviously not the shortest path, thus we have to take consideration into the network delay of the voice data packet.

Network delay time includes Propagation delay (5 μ sec/km) and Queuing delay. The propagation delay has nothing to do with the data packet size.

While Queuing delay is the time spent in a router processing and thus is usually affected by the packet size and the network congestion status.

Considering the features of voice data packet, we simulate transmitting data packet of size 240 bytes and 80 bytes and collect the delay statistics.

The environment used in our simulation is shown in Table 7 and Figure 15. Assume the propagation delay between any two nodes is 10ms (except the delay between node 5 and node 6,) the bandwidth is 100Mbps. The number of FTP connections to be observed is 3, 6, 9, 30, and up to 300. The simulation results are shown in Figure 16.

Figure 16 shows that the average delay on a node is in proportion to the packet size. Though the delay time will increase as long as the congestion is getting worse, the proportion remains almost the same.

Table 7. Environment for delay simulation

Simulation software	Network Simulation 2(Version 2.27)
Packet size	80 bytes UDP/IP 240 bytes UPD/IP
Link state	Delay：10 *ms*, Bandwidth：100Mb， Queue：droptail

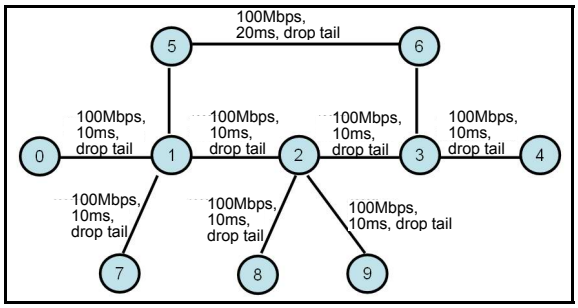


Figure 15. Topology used in the delay simulation

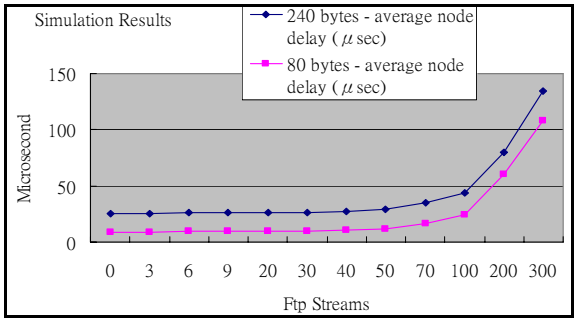


Figure 16. Network delay simulation

6. Conclusion

In this paper, we apply the visual cryptographic concepts, which original used in image procession, to secure SIP VoIP. We segregate the voice data into two shares, and then transmit them on two disjoint paths over the network. Hackers cannot recover the voice data if they only intercept any one share. We use an alternative disjoint path to transmit the shorter share of the segregated voice data since the alternative path is not the shortest path. Since we implement the disjoint multi-path algorithm in the SIP proxy servers, it is not necessary to change the network environment and thus the deployment procedure should not be difficult.

References

[1] D. Collins, *Carrier Grade Voice over IP*, McGraw-Hill, New York, 2003.

[2] Swades De, Sajal K. Das, “Dynamic Multipath Routing (DMPR): An Approach to Improve Resource Utilization in Network for Real-Time Traffic,” *International Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2001, pp.23-32.

[3] P.-J. Chuang, H.-Y. Tu, "Dynamic Scheme for Reducing Hot Spot Effects in Multipath Networks," *IEE Proceedings - Computers and Digital Techniques*, Vol. 146, No. 4, 1999, pp. 179-184.

[4] Swades De, Chunming Qiao, "Does packet replication along multipath really help?," *IEEE International Conference on Communications,* Vol. 26, No. 1, 2003, pp. 1069-1073.

[5] M. K. Ranganathan and L. Kilmartin, "Investigations into the impact of security protocols in Session Initiation Protocol (SIP) based VoIP networks," In *Proceedings of the Irish Signals and Systems Conference*, 2001.

[6] Moni Naor, Adi Shamir, “Visual Cryptography,” *Advances in Cryptology - Eurocrypt'94 Proceeding*, Vol. 950, 1995, pp.1-12.

[7] Deepinder Sidhu, Raj Nair, Shukri Abdallah, “Finding Disjoint Paths in Networks,” *ACM SIGCOMM Computer Communication Review*, Vol. 21, no 4, 1991, pp.43-51.

[8] W. Lou and Y. Fang, “A multipath routing approach for secure data delivery,” in *Proceedings of IEEE MILCOM*, Vol. 2, 2001, pp. 1467-1473.

[9] Bao Hong Shen, Bin Hao, Arunabha Sen, “On Multipath Routing using Widest Pair of Disjoint sets,” Workshop on *High Performance Switching and Routing*, 2004, pp.134-140.

[10] J. Rosenberg, et al, “RFC 3261 SIP : Session Initial Protocol,” 2002.

[11] ITU-T Recommendation G.711, “Pulse code modulation (PCM) of voice frequencies,” International Telecommunication Union, 1988.

[12] ITU-T Recommendation G.726, “40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM) ,” International Telecommunication Union, 1990.

[13] ITU-T Recommendation P.800 “Methods for subjective determination of transmission quality,” International Telecommunication Union, 1996.