VCNF: A Secure Video Conferencing System Based on P2P Technology

Changlai Du, Hao Yin, Chuang Lin, Yada Hu
Tsinghua National Laboratory for Information Science and Technology
Computer Science and Technology Department, Tsinghua University,
Beijing, 100084, P.R.China
{dcl, hyin, clin, yadandaner}@csnet1.cs.tsinghua.edu.cn

Abstract

The goal of a video conferencing system is to deliver multimedia data to the conference participants efficiently and securely. To meet the requirement of a video conferencing system over the internet, many issues have to be addressed, such as security, scalability and heterogeneity. In this paper, we propose a secure and scalable video conferencing system named VCNF (Video Conference Network Foundation), to support large number of conferencing groups simultaneously with each group has limited members. We use a P2PSIP overlay to store and lookup the user and group contact information. SIP protocol is used to setup media sessions and a new kind of session-- VPN session. We adopt a layered data transfer mechanism which traits different kinds of media-streams as of different importance level. We have implemented the system and analyzed its performance and security. An Internet video-based game based on our platform has also been deployed over the Internet.

1. Introduction

Over the last years, with the growth of the Internet, multimedia applications including live streaming and multimedia conferencing over the Internet have been envisioned as the killer applications and also a hot research area.

Unlike live streaming, multimedia conferencing has its own key characteristics [1]:

• Performance requirements: Conferencing applications require low latencies, and need to sustain high bandwidth between the source and receivers. In contrast, live streaming is more tolerant to latency.

• Gracefully degradable: Conferencing applications deal with media streams that can tolerate loss through degradation in application quality.

• Session lengths: Conferences are generally long lived, lasting tens of minutes.

• Group characteristics: Conferences usually involve small groups, consisting of tens to hundreds of participants. Membership can be dynamic. This is in contrast to applications like live streaming, which may have millions of receivers at the same time.

• Source transmission patterns: Typically, conferencing applications have a source that transmits data at a fixed rate. While any member can be the source, there is usually a single source at any point in time. In contrast, large scale broadcasting applications may have a single static source throughout a session.

To meet the requirements and characteristics above and to provide commercial video conferencing services, many key challenges must firstly be addressed.

First, conferencing contents sometimes are private or related to commercial secrets, so security mechanisms must be in place, both for the session setup procedure itself and for the media contents in the session. However, this is quite challenging because of the inherently non-secure attribute of the Internet.

Second, the system has to be scalable to support a large number of people online simultaneously who are divided into size-limited conference groups.

Third, the Internet is a heterogeneous environment, where some people are in corporate LANs which are connected to the Internet via 1000M fiber cable, but a lot of other people are behind an ADSL or even modem. To deal with the heterogeneity feature and provide different services is hard because the network is dynamic and measurement of network condition is not easy.

In order to address the issues talked above, this paper proposes a novel system named VCNF, who organizes users into a P2PSIP [2] overlay network that stores user contact information and conferencing group information into distributed peers. VCNF dynamically setup VPN sessions using extended Session Initiation

978-0-7695-3352-0/08 $25.00 © 2008 IEEE
DOI 10.1109/HPCC.2008.128

Protocol (SIP) [3] to provide security mechanism. We divide different media types in a session as of different importance level and transfer media data according to users' network conditions, in which way we can address the issue of heterogeneity.

The rest of this paper is organized as follows. Section 2 reviews the background and some related work. Section 3 presents the VCNF architecture and specifies each component. Section 4 gives the implementation details and section 5 talks about the experimental results and analyze the performance. Section 6 concludes this paper.

2. Related Work

In this section, we give a brief overview of the related techniques including some related video conferencing architectures, P2PSIP related information and several security related topics.

2.1. Architecture

After years of research, IP multicast [4] has been improved not suitable due to concerns related to scalability and deployment issues. With the development of media-coding method, P2P and Overlay Network techniques, Application Level Protocol (ALM) has been proposed. In ALM, nodes are organized into overlay spanning trees for data delivery.

As for size-limited group video conferencing, the most famous ALM schemes are End System Multicast [5][1] and ALMI[6]. End System Multicast organizes the multicast group members into a mesh, with each member maintaining all the others information. Then, they construct shortest path spanning tree rooted at the corresponding source using well-known routing algorithms. The shortcoming of End System Multicast is that to maintain the spanning trees cost too much and the system is not scalable. ALMI keeps a Minimum Spanning Tree (MST) among group members but it cannot be optimized for all data sources.

The same as End System Multicast, VCNF proposed in this paper is suitable for size-limited conferencing environments. The distance learning and remote education application with a static single source and large number of receivers is more like a live streaming system with some latency limitation.

Different from End System Multicast and all the other ALM architectures, VCNF is a whole solution including not only data delivery mechanism but also user and group management functions. As presented in Section 3 and 4, VCNF data delivery scheme is a mixed using of a central server named MCU and a mesh structure. However, it can easily adopt other schemes such as an ALM mesh-tree structure by including ALM algorithms in VCNF clients.

2.2. P2PSIP

P2PSIP is an emerging research area including a set of protocol standards and mechanisms for using SIP in settings where the service of establishing and managing sessions is principally handled by a collection of intelligent endpoints, rather than by centralized servers as in SIP currently deployed.

P2PSIP overlay is used not only to transfer session initiation messages but also to store user contact information. The overlay acts as a distributed database using some hash algorithms to reflect a node's address or a user name into a hash namespace.

Research on P2PSIP is now in progress and is expected to form Internet standards next year [2].

2.3. Security

VPN can be used to create a secure, policy-based overlay network within the Internet. VPN uses the Internet as a transport while creating a secured tunnel within it. IPSec vpn is the most popular remote access solution with high security performance, but it is designed for site-to-site connectivity and is costly to manage. SSL vpn is much easier to manage and suitable for movable users to access private resources. OpenVPN[7] is the first and most popular SSL VPN products.

Setting up a vpn tunnel includes at least two steps: authentication and session key negotiation. PKI is the mostly used technique for authentication. Each user has a public/private key pair generated by Certificate Authority (CA). This key pair can then be used for encryption and signature.

Polycom[8] has a solutions guide talking about how to deploy vpn within a corporate enterprise for video conferencing. However, the scheme mentioned in that guide focuses on enterprise applications, and only supports site-to-site vpn.

In VCNF, we explore a username/password scheme for user login, a public/private key pair for authentication and to negotiate session key. A symmetric encryption algorithm is adopted to encrypt media data.

3. VCNF Architecture

3.1. Control plane architecture

The responsibility of the control plane is to start up vpn and media sessions and to maintain these sessions.

We use Chord [9] as the underlying P2P overlay to store user contact and video group information, and to deliver sip messages.

Chord is a ring-based distributed hash table (DHT) for structured P2P systems. Chord is selected because of its simplicity, convergence properties, and general familiarity within the P2P community. In Chord, peers and resources are hashed into peer-ID and resource-ID respectively in the same namespace. Resource with resource-ID k will be stored by the first peer with peer-ID equal to or greater (mod the size of the namespace) than k, ensuring that every resource-ID is associated with some peer.

In VCNF, we define three kinds of IDs:

• PID is the unique ID for a peer in the overlay, which is calculated using a Hash algorithm:

$$PID = Hash(peer\text{-}IP{:}port) \quad (1)$$

A peer with a PID is responsible to hold resource data assigned to it with the algorithm in Chord.

• UID is the unique user ID calculated using the same Hash algorithm:

$$UID = Hash(username) \quad (2)$$

Username is a SIP URI, and its definition can be found in Section 4.1. A <*username, IP:port*> which is called the user contact information, is stored in responsible peers according to Chord algorithm.

• GID is the unique video conference group ID calculated using the same Hash algorithm:

$$GID = Hash(groupname) \quad (3)$$

The definition of *groupname* can also be found in Section 4. A < *groupname, IP:port*> is stored in responsible peers according to Chord algorithm.

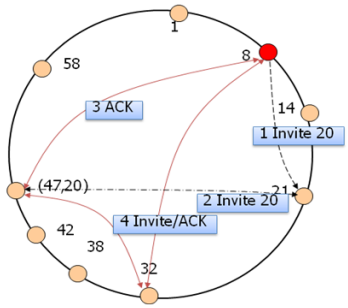

Figure 1. **VCNF network architecture**

Figure 1 illustrates a basic VCNF network structure. Peers are organized into a Chord ring, where each one has a unique PID. In this figure, peer with PID 47 is responsible for storing contact information of user 20. This figure shows a simple setting-up procedure for a Video Conferencing Group (VCG). We now talk about the VCG session set-up process in detail.

User registration: A user must first join the VCNF overlay and register his location. Actually, before the user can join the overlay, he/she must first login to a global *VCNF Login Server*, which is not illuminated in Figure 1. If the user passes the login check, he then gets a list of *VCNF Bootstrap Server* addresses which will help him join into the overlay following the rule in Chord.

Once the user joins the overlay successfully, he gets a PID using formula (1) for the node and a UID using formula (2) for his name. The peer also has now a finger table including some <*PID, IP:Port*> pairs called finger table entry. The user then sends a sip REGISTER message to another peer whose PID is just larger than the user's UID. The REGISTER message contains the user's contact information, mainly a <*username, IP:port*> pair, which is defined in section 4.1.

User location: Given that two users, Alice and Bob, have joined the VCNF overlay, now Alice wants to invite Bob to start a private video session. The first thing Alice has to do is to find Bob and sends him a sip INVITE message.

Assume that VCNF peers act as stateless sip proxies. As shown in Figure 1, Alice's PID is 8; Bob's PID is 47 and his UID is 20. Alice knows Bob's username, and gets his UID using (1). She then sends an INVITE message to peer 21 whom she thinks is responsible for storing Bob's contact.

Peer 21 has Bob's contact, and transfers the INVITE message to Bob on peer 47. The INVETE message contains Alice contact and Bob answers Alice with an ACK message. A session now can be started.

User Authentication: VCNF has a global *Login Server*. The Login server generates its own public/private key pair, (S_S, V_S) using RSA algorithm. V_S is distributed to all VCNF clients at build time so that every VCNF client can verify he is really talking to the Login Server. Each client generates its own key pair, Alice with (S_A, V_A) and Bob with (S_B, V_B). When Alice and Bob login to Login Server for the first time, Login Server generates an IC_A by signing Alice's username and V_A using its own signing key S_A. Bob also gets his IC_B.

Alice and Bob can now trust each other using a challenge response mechanism.

VPN Session Setup: Alice sends an INVITE message to Bob. This message takes a SDP [10] description that tells Bob the session type is a vpn session. Alice selects a 256-bit AES key, G_K, as the session key. Alice signs to this key with S_A, and sends it to Bob together with IC_A after encrypted by Bob's V_B, that is $E_{VB}(E_{SA}(G_K), IC_A)$.

Bob receives the message and decrypts it to get G_K. Bob gets the other vpn parameters including its virtual IP address and connects to Alice to setup a VPN

session.

Register The Group To VCNF: Once a vpn session sets up, all members in the session together are called a Video Conference Group (VCG), and the member who starts the session is called the VCG owner.VCG owner is responsible for registering the VCG to VCNF overlay.

Each VCG has a unique *groupname* which is defined in section 4.1. Group registration is more or less the same as user registration, as long as VCNF makes sure that *username*s and *groupname*s are all different.

New User Join the Session: New user sends an INVITE message to a registered group to join a session. The INVITE message is actually sent to the VCG owner. The VCG owner keeps the new member's contact and sends it all the other members' contact information.

3.2. Data plane architecture

As well as a vpn session has been set up, a group of users are organized together and each member knows the others. In another word, these group members are now in a full connected mesh graph.

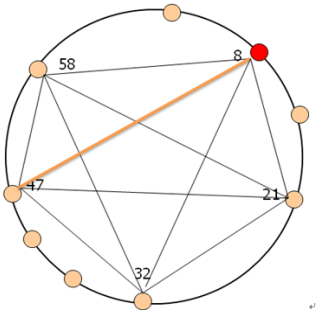

Figure 2. **Data plane architecture**

As shown in Figure 2, five users setup a vpn session and are now in a virtual LAN. Note that the vpn set up here is different from traditional site-to-site vpn, where all tunnel data are transferred by a central vpn server. As all members know about the others, they have all the other members' real IP address and thus can send and receive data peer-to-peer.

However, in VCNF we adopt a mixed structure. When a user start VCNF client for the first time, he/she is asked to set the connection type. VCNF will remember whether the connection type is 100M lan or ADSL or behind a 56kbps modem. If there is a member who has a network bandwidth up a certain threshold, it will be selected as an MCU (Multipoint Control Unit), or else this group should ask VCNF for a MCU server. We will talk about VCNF servers in detail in section 4.

A video conference session may have several data streams. In VCNF, there are four types actually: active user video stream, inactive user video streams, audio streams and text streams. VCNF delivers these four types by different ways: active video streams with high definition and high frame rates as well as audio streams are delivered first to MCU, and MCU transfer these streams to all group members; inactive video streams and text data are sent to all group members directly in a mesh way.

The reasons why VCNF adopt this scheme are as follows:

First, active user video stream consume too much network bandwidth if delivered to many other members.

Second, although audio streams are not bandwidth consuming, audios from different members are required to be mixed.

Third, inactive video streams with low definition and low frame rate are selectable. In some scenarios, they are not necessary.

Text messages need little bandwidth, and in addition, sometimes members have the requirement to talk to each other privately. This is why text messages are sent in a mash manner.

4. Implementation

4.1. Definitions

This part defines the items mentioned in section 2 and section 3.

Username: Username is a SIP URI. A user's email address is chosen as it is globally unique, e.g. *sip:alice@thu.edu.cn*. Using a valid email address as the username has other advantages. For example, it allows the system to generate a random password and email it to the user for authentication.

Groupname: Groupname is also a SIP URI. But we include an 'isfocus' feature parameter in the groupname contact header field to express that the SIP dialog belongs to a conference, e.g.,

<sip:alice@thu.edu.cn>;isfocus

User contact: User contact information is a key/value pair stored in the overlays distributed database. User contact in VCNF is defined in XML format, as is shown below

```
<key>sip:alice@thu.edu.cn</key>
<value >
<contacts>
    <contact displayName="Alice">
    sip:alice@166.111.1.2:5060
</contact>
</contacts>
</value>
```

A user can have more than one contact. There is an added contact property "displayName" implying the user's human friendly name.

4.2. System implementation framework

Our system implementation framework is shown in figure 3. The system includes one global login server for user administration; one or more bootstrap server(s) to help peers join the VCNF overlay; several MCU servers to help video groups delivering video content; and a global group management server to maintain and manage conference groups.

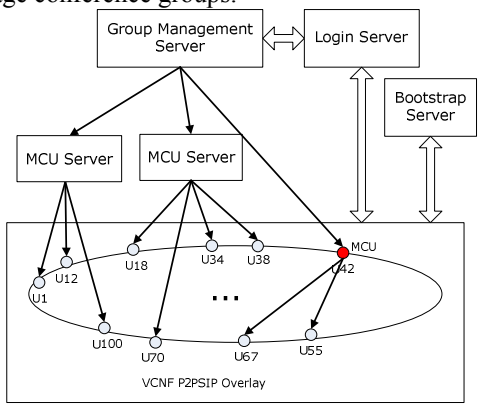

Figure 3. VCNF Implementation Framework

Login server also acts as a global CA, which is responsible for signing and generating user ICs.

Conference group can apply to Group Management Server for an MCU server to help deliver media content. An MCU server receives video streams from an active user and sends them to all other members in the same group. Suppose that the video stream bitrates is 128Kbps, them a MCU with a gigabyte network connection can support about 600 users simultaneously.

4.3. VCNF client framework

Figure 4 is the client software architecture of VCNF, which contains several components including P2P protocol implementation, a SIP protocol stack, VPN server and client, a video codec and a user interface.

We use the openssl [11] library to setup ssl vpn tunnels. The client work flow is as follows:

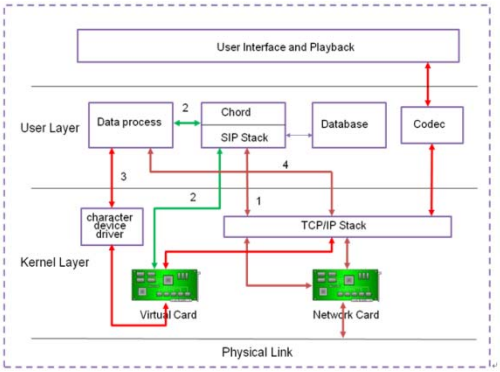

Figure 4. VCNF Client Framework

First, after user login to login server, Chord starts a process of joining the overlay, as is shown in figure 4 step 1.

Then the user selects one of his friends to start a video session, SIP stack gets the friends name and starts a User Location process. The user invites his friends to start a vpn session first and they negotiate for a session key and their virtual IP addresses.

Data process component gets the session key and the virtual IP as the green line shows in Figure 4 step 2.

The user then starts a video camera to capture videos, and start to receive media data to playback. The media data are first sent to the virtual card. VPN data process component gets the data from character device driver, encrypts the data with the session key and sends the data to the real network card. A converse process takes place when data are received.

4.4. Source coding

We use xvid [12] codec library to encode and decode captured video frames. Xvid is a MPEG-4 video codec library with high compression ratio and fast compression speed. Thus it is adopted in live communications. VCNF can be configured to support different video definitions and frame rates. An active user has a video window with a definition of 320*240, and a 15 fps frame rate. The inactive users' video window is 176*144 with a 5 fps frame rate. As our experimental result shows, the active user video stream bit rate is about 70Kbps after xvid decoding. And an inactive video stream is about 30Kbps. The bit rates are mean values and differ a lot with static or dynamic background.

VCNF adopts G.729 as its audio encoding method, with a constant bit rate, 8Kbps.

A VCNF conference group has an active user and up to 16 inactive users.

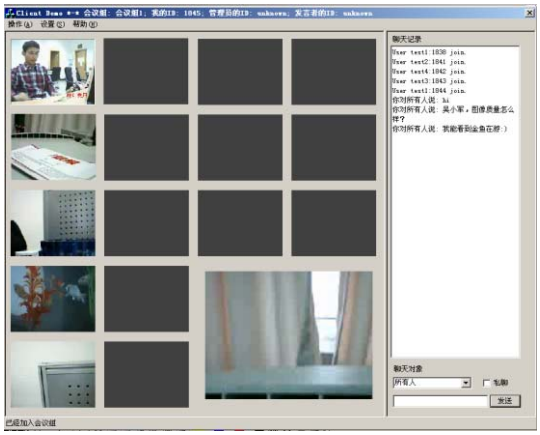

Figure 5. A user interface demo of VCNF

Figure 5 shows a user interface demo of our VCNF system with an active user and up to 16 inactive users in a conference group.

4.5. Authentication and encryption algorithms

We use RSA [13] algorithm to generate user public/private key pairs. The asymmetrical encryption scheme is used for user authentication and session key negotiation.

We use AES-256 for session cryptography. AES-256 provides sufficient security performance and a quite good encryption speed.

Note that when a user joins or leaves a session, MCU is responsible for updating the session key and broadcasting it to all valid members.

4.6. Dealing with heterogeneity

In VCNF, MCU maintains a global permission table (GPT) and each member maintain s local permission table (LPT). Members can control whose streams he wants to receive. Is a member denies to receive another one's video or audio streams, the request will sent to MCU and transferred to the target member. The target member's LPT is updated and he will never send the forbidden stream data to the request user. MCU can control every link between any members. Actually, MCU user acts as a chairman of the conference.

LPT is useful when a user's network can't afford the received bit rate. For example, he can easily disable all the inactive video windows and only receive audio and active video window data.

Network conditions could be auto detected by VCNF, but as it is not easy to measure the network conditions dynamically, we just leave it as a future work.

5. Performance Evaluation

In video or voice communications, two values are considered to be performance metrics: call setup time and media delay time.

As our system pays much attention to security, we also have security performance as an additional metric.

5.1. Session setup time

As presented in sections above, we use Chord for the user lookup algorithm. In Chord overlay, for any user contact information, the node whose range contains the user is reachable from any node in no more than log_2N overlay hops, where N is the size of Chord namespace. Assuming that mean network delay time between overlay hops is *TTL*, then it takes log_2N*TTL for the first INVITE message to reach the target peer.

Session key negotiation involves a 3-way handshake process. Thus the total time taken to setup a session is about:

$$T = log_2N*TTL + 3*TTL$$

Suppose that N = 1,000,000 and TTL < 100ms, then

$$T < 2.3s$$

Our experiment result verified the conclusion—the session setup time seldom extends 3s.

5.2. Delay

As mentioned above, active video stream and audio stream are delivered via MCU, and inactive video data are delivered directly between group members. Audio data also need to be mixed in MCU and thus audio streams have the largest delay. Assume that data encoding or decoding time is Co_T, encryption or decryption time is Cr_T, mean network delay is *TTL*, then the total audio stream delay can be expressed as:

$$D = 2* Co_T + 2* Cr_T + 2* TTL = 2*(Co_T + Cr_T + TTL)$$

Experiment result shows that $Co_T <= 40ms$, $Cr_T < 10ms$, and $TTL < 100ms$, so

$$D < 300ms$$

According to VoIP standard, it is an acceptable delay time.

5.3. Security

As was shown in our previous work [14], SSL vpn has almost equal safety with IPSec whose high security performance is well-known all over the network research area.

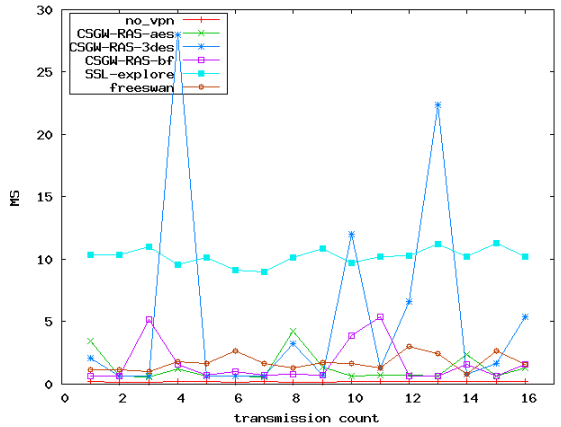

Figure 6. **Session setup time**

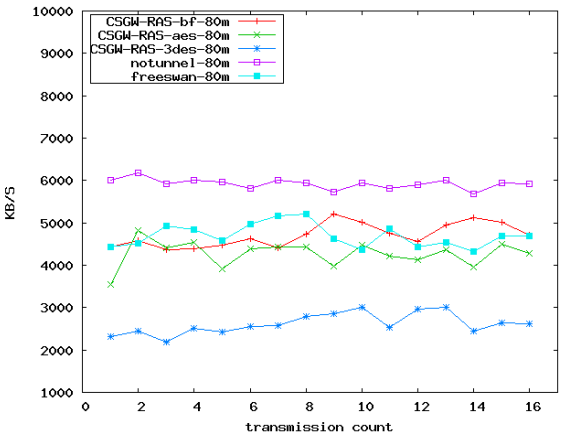

Figure 7. **Data transfer speed**

Figure 6 and Figure 7 from [14] demonstrate that using vpn in an video conferencing system takes a negligible session start time and about 20% data transfer delay.

6. Conclusions

In this paper, we present a scalable and secure video conferencing system called VCNF. We use a P2PSIP overlay to store and lookup the user and group contact information. We explore SIP protocol to setup media sessions and a new kind of session-- VPN session. VPN over P2PSIP here acts as a security infrastructure for our conferencing system. As to the data plane, considering the heterogeneity of the Internet nowadays, we adopt a layered transfer mechanism which traits different kinds of media-streams as of different importance level. Both analysis and experiment results show that VCNF is an efficient and secure Internet video conferencing system.

7. Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 60673184), the National High Technology Research and Development Program of China (No. 2007AA01Z419), and the National Basic Research Program of China (No. 2008CB317101), and ChinaCache-Tsinghua joint research project.

8. References

[1] CHU Y H, RAO S G, SESHAN S, ZHANG H. Enabling conferencing applications on the Internet using an overlay multicast architecture[J]. ACM SIGCOMM Computer Communication Review, 2001, 31(4):55-67.
[2] P2PSIP status pages. http://tools.ietf.org/wg/p2psip
[3] J. Rosenberg and H. Schulzrinne, G. Camarillo, A.R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. RFC 3261, Internet Engineering Task Force, June 2002.
[4] S. Deering. Multicast Routing in Internetworks and Extended Lans. In Proceedings of ACM SIGCOMM, August 1988.
[5] CHU Y H, RAO S G, SESHAN S, ZHANG H. A case for end system multicast[J]. ACM SIGMETRICS Performance Evaluation Review, 2000, 28(1):1-12.
[6] PENDAKARIS D, SHI S. ALMI: an application level multicast infrastructure[A]. Anderson T. The 3rd USENIX Symposium on Internet Technologies and Systems[C]. San Francisco, CA, USA: USENIX Association, 2001. 49-60.
[7] OpenVPN homepage. http://openvpn.net/
[8] Deploying Secure Enterprise Wide IP Videoconferencing Across Virtual Private Networks. http://www.h323forum.org/papers/polycom/DeployingSecureIPVideoNetworks.pdf
[9] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrish-nan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM*, San Diego, CA, USA, Aug 2001.
[10] M. Handley and V. Jacobson. SDP: Session Description protocol. RFC 2327, Internet Engineering Task Force, April 1998.
[11] http://www.openssl.org/
[12] xvid homepage. http://www.xvid.org/
[13] http://www.rsa.com/rsalabs/node.asp?id=2125
[14] Yada Hu, Hao Yin, Chuang Lin, Xin jiang, Ying Ouyang, Chao Li. CSGW-RAS：A Novel Secure Solution for Remote Access Bases on SSL. ISPACS 2007.