A Secure and Hierarchical Architecture for P2PSIP Session Initiation

Xianghan Zheng, Wenzhong Guo, Shangping Zhong, Zhiyong Yu
[1] College of Mathematics and Computer Science
[2] Fujian Key Laboratory of Scientific and Engineering Computing
Fuzhou University, Fuzhou, P.R. China, 350108
{xianghan.zheng, guowenzhong, spzhong, yuzhiyong}@fzu.edu.cn

Abstract

Recently, both academia and industry have initiated research projects directed on integration of P2PSIP paradigm into communication systems. In this paradigm, P2P network stores most of the network information among participating peers without help of the central servers. The concept of self-configuration, self-establishment greatly improves the robustness of the network system compared with the traditional Client/Server based systems. In this paper, we propose a system architecture for constructing secure P2PSIP session initiation. The proposed approaches include: three-layer hierarchical overlay division, proxy based security, subjective logic based trust enhancement, NAT traversal, message routing, etc. After that, a prototype with 512 P2PSIP peers is implemented. We evaluate the system architecture in several aspects: implementation testing, Number of hops and the protection against malicious or compromised intermediate peers. We take Chord as the P2PSIP overlay asexample. However, this system architecture is independent of Chord overlay and could be extended to the other DHT (Distributed Hash Table) technologies.

Keywords: *Peer-to-Peer (P2P), Session Initiation Protocol (SIP), P2PSIP, Chord, Chord Secure Proxy (CSP), Chord Secure Proxy Gateway (CSPG), Subjective logic.*

1. Introduction

Currently, P2P computing has begun to infiltrate into SIP communication systems. The decentralized nature of P2P might provide distributed peer-to-peer communication system without help of the traditional SIP server.IETF P2PSIP working group defines the motivation of P2PSIP [1]: The concept behind P2PSIP is to leverage the distributed nature of P2P to allow for distributed resource discovery in a SIP network, eliminating (at least reducing) the need for centralized servers.

However, the decentralized nature of P2P comes to the cost of less or decentralized management, which creates security problems. Firstly, due to lack of centralized credential mechanisms for authentication and authorization, all the message flows among participating peers from beginning to the end are distrusted and unsecure.

Let us take a look at a malicious model presented on Fig.1, Peer C who acts as a malicious intermediate peer might understandand preserve sensitive privacy (e.g. peer identity, IP address, Port, etc) via parsing the incoming message. This privacy information can be used to initiate DoS attack on a specific peer or the overlay network. It might also be sold to illegal advertisement parties, which results in SPAM messages or calls.

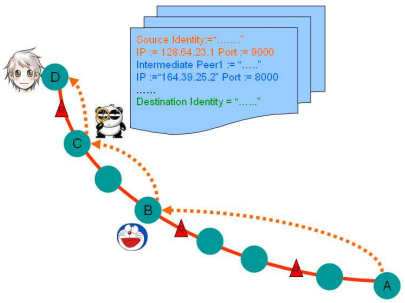

Fig.1. A Malicious Behaviour Model

Besides, portable peers (e.g. mobile phone, PDA, Pad, etc) might cause Churn problem (peer joins and leaves the overlay frequently) due to unreliable connection or other limitations. Therefore, portable peers that act as an intermediate peer might not be reachable and this reduce system availability, which has become one of the most important aspects in security field.

In this paper, we investigate on P2PSIP security issues and propose a system model that is capable to providesecure services during peer session initiation process.Our proposed solution could also provide better system availability through dividing peers with different capability into hierarchical sub-layer.

The paper is organized as follows. In Section 2, we consider the related works. The proposed system

978-0-7695-4676-6/12 $26.00 © 2012 IEEE
DOI 10.1109/IPDPSW.2012.144

architecture and corresponding technical issues are specified in Section 3. After that, we describe use scenarios in Section 4 and illustrate prototype implementation in Section 5. The evaluationwork is presented in Section 5. Finally, we draw the conclusions and future work in Section 6.

2. Related Works

Research efforts on P2PSIP security are mostly based on the PKI-based certificate approaches that have been proposed in the literature[2, 3]. In these solutions, Certificate is issued by Certification Authority (CA) and used to handle encryption and decryption. Since the certificate is bounded to a specific peer, it is undeniable and unfalsifiable. However, certificates are not enough in decentralized P2PSIP network. The main problem is that peers are capable to perform malicious behavior after receiving legal certificate. For example, a malicious peer might pretend to be non-malicious and get legal certificate. After that, it could join the overlay and expose malicious behaviors (as shown in Fig.1).

In closed or ephemerals network, pre-shared key (PSK) approach [4] can be more convenient. Pre-shared keys are symmetric keys shared among the peers in advance to establish secure connection. It can be a password like “hElLo#QWoRld”, a passphrase like “Woaini”, or a hexadecimal string like “AUS30209-DOP745”. Using pre-shared keys can help to avoid the need for public key operations. However, pre-shared key can only provide limited security. For instance, an attacker could initiate a DoS attack by sending a larger amount of exchange key request to a peer. Also, it lacks efficient mechanisms to prevent Man-in-the-Middle (MiM) and replay attacks.

To solve the security problem that peers in the overlay are distrusted each other, a proxy-based security approach is proposed in[5, 6]. In this solution, a secure proxy is proposed to relay the data between source and destination peers without being interrupted by malicious intermediate peers.

Another enhancement might be based on subjective trust, proposed in[7]. The subjective logic deals with three parameters (trust, distrust, and uncertainty) to describe the trust value, which gives more adequate trust model of real world.

Paper [8] proposes a hierarchical virtualization model, in which a P2PSIP system is logically divided into*N*sub-layers according to peer capabilities (e.g. CPU processing power, bandwidth, storage, etc). According to performance analysis, hierarchical division increases overall system capability (via reducing the delay) and is capable to enhance system availabilitywhen setting up connection.

The system architecture and corresponding solutions proposed in this paperis the combination of above approaches. In the next Section we describe the system model and then use it in the following sections for the implementation and evaluation.

3. System architecture

In this Section, we introduce the proposed system architecture. After that, we specify the corresponding approaches in detail, including three-layer hierarchical division, peer identifier assignment, cache mechanism, CSP based security, and subjective logic based trust enhancement, etc.

3.1. Architecture overview

The proposed system architecture includes six main units: P2PSIP peer, Chord Secure Proxy (CSP), Chord Secure Proxy Gateway (CSPG), Enrollment and Authentication (E&A) Server, Secure Opinion Server (SOS), and STUN[1]/TURN[2]/ICE[3] server, as shown in Figure 1.

P2PSIP peer, which can be a mobile phone, laptop, PC, etc., is connected to the Internet. CSP is the secure proxy that helps source peer to locate the destination peer. E&A server is the secure server that handles the enrollment and authentication task when P2PSIP peer joins overlay. Secure Opinion Server (SOS) is the trust server that stores and computes dynamic opinion for each P2PSIP peer. STUN/TURN/ICE server is responsible to provide NAT traversal for those peers behind NAT protection.

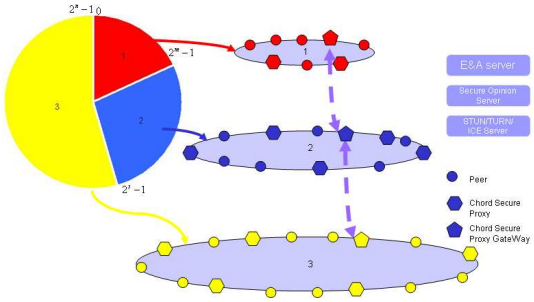

Fig.2. System model

In the system, the overlay is divided into three sub-layers. If the destination peer is in the same layer as the source peer, request would be sent to CSP that is clockwisely nearest to the destination peer. Otherwise, the messages are directed to CSPG in source layer, then CSPG in destination layer, and finally to the corresponding CSP that is clockwisely nearest to the destination peer. CSP is responsible for search and location of destination peer via

[1] STUN: Simple Traversal of UDP through NATs
[2] TURN: Traversal Using Relay NAT
[3] ICE: Interactive Connectivity Establishment

"pingRequest" multicast mechanism (described in Section 3.3). After that, the session between source and destination peers may be established.

3.2. Three layer hierarchical division

We suggest divide the overlay into three sub-overlays according to peer capabilities, as shown in Fig.2. The first sub-overlay consists of stable peers that have public IP addresses, more powerful CPU, and stable connection. Such typical device can be a web server. Peers in the second sub-overlay are those who have enough stability and processing power, e.g. normal PC with Internet connection. Peers in this layer do not own public IP address, and might relay on STUN/TURN/ICE for NAT traversal. The lowest sub-overlay is those with unstable connection (e.g. mobile phones, PDA, laptops with wireless connection). Note that each sub-overlay contains a few CSPs for handling security services in intra-layer, and at least one CSPG (Chord Secure Proxy Gateway) for handling secure inter-layer communication. Both of CSP and CSPG are assumed to be stable P2PSIP peers.

It is expected that many legacy P2PSIP peers in the future are unstable peers (e.g. a large amount of mobile phones, PDA, laptops, etc) with wireless connections. Therefore, the division of three sub-overlay guarantees peer/resource lookup efficiency in the top two layers.

3.3. Peer identifier assignment

IETF P2PSIP WG is still discussing the assignment of peer identity in the overlay. Some researchers suggest use conventional SHA-1 hash mechanism to produce 128/160 bits peer identifier. However, this solution might cause efficiency problems. For example, geographically close peers might be assigned with identifiers that are far away from each other in the overlay, and this causes long delay during connection establishment.

We advocate the idea that geographically close peers should be assigned close peer identifiers in the overlay because the most frequently communicated peers are those who are geometrically related to each other[9]. We propose to combine this idea into our hierarchical system. In the beginning of enrollment, P2PSIP peer should contact an Enrollment and Authentication (E&A) server (which is a central server), submit information about peer capabilities (e.g. connection type, CPU processing power, bandwidth, storage, etc) and geometry information (e.g. public IP, etc), etc. Based on peer capabilities, E&A server allocates specific sub-overlay; based on geometry information, E&A server assigns specific peer identifier attached in specific sub-overlay.

3.3. Proxy-based Security

Fig.3 shows how CSP provides secure session initiation services. CSP acts as a proxy server for receiving P2PSIP request from source peer (Step 1), and on the other side, for probing destination peer via multicasting "PingRequest" messages to a few of its successors that are in the anti-clockwise direction of the destination peer (Step 2). The multicast messages are forwarded step by step until the destination peer through standard Chord routing algorithm. When destination peer receives multiple "PingRequest" messages, it chooses one of them for handling (based on calculated trust value described in next section) and replies with a "PingResponse". After that, CSP forwards the original P2PSIP request (Step 3) and the connection between source and destination peers can be securely established (Step 4). Note that all connections are SSL/TLS secured.

The use of "Ping" message (in Step 2) makes sure that intermediate peers are incapable to understand original privacy sensitive P2PSIP request. The proposed multicast mechanism (Step 2) guarantees on some level that "Ping" message is capable to arrive at the destination peer. In conclusion, security is guaranteed as long as CSP is trusted and secure.

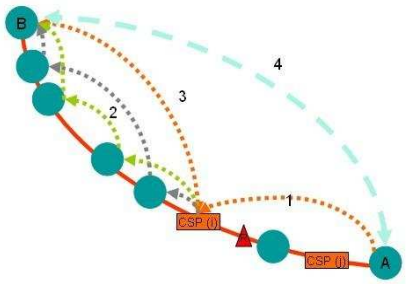

Fig.3.CSP based Security

3.4. Subjective Logic based Trust

The subjective logic[10] defines the term "opinion", which is a triple $\omega = \{t, d, u\}$, where t, d and u correspond to trust, distrust, and uncertainty respectively. Expressing trust by using three parameters instead of one simple trust level gives more adequate trust model of real world. Subjective logic also defines logical operators for combining opinions. For example, the recommendation operator $\otimes$ can be introduced to evaluate the trust worthiness of p which might be a statement like "the message traverse B via A is unchanged result of measurement", as following:

$$\omega_p^{AB} = \omega_B^A \otimes \omega_p^B = \{t_p^{AB}, d_p^{AB}, u_p^{AB}\}$$

Where

$$t_p^{AB} = t_B^A t_p^B,\ d_p^{AB} = t_B^A d_p^B$$

and $u_p^{AB} = d_B^A + u_B^A + t_B^A u_p^B$.

ω_B^A is the opinion of A about trustworthiness of recommendation of B, and ω_p^B is the opinion of B about trustworthiness of p.

We propose a Secure Opinion Server (SOS) to store and manage dynamic opinion for each peer. Therefore, destination peer should contact SOS in order to get the corresponding opinion value (as shown in Fig.4). Suppose that a request goes from source peer A, intermediate peers B_1, B_2, B_{n-1}, and ended in the destination peer B_n. By applying the rules of subjective logic, the trustworthiness of this data flow is:

$$\omega_p^{AB_2...B_{n-1}B_n} = \omega_{B_1}^A \otimes \omega_{B_2}^{B_1} \otimes \omega_{B_3}^{B_2} \otimes .....\otimes \omega_{B_n}^{B_{n-1}} \otimes \omega_p^{B_n}$$

Where

$$t_p^{AB_1B_2...B_{n-1}B_n} = t_{B_1}^A t_{B_2}^{B_1} t_{B_3}^{B_2} ....t_{B_n}^{B_{n-1}} t_p^{B_n} = t_{B_1}^S t_{B_2}^S t_{B_3}^S ....t_{B_n}^S t_{B_n}^S$$

$$d_p^{AB_1B_2...B_{n-1}B_n} = t_{B_1}^A t_{B_2}^{B_1} t_{B_3}^{B_2} ....t_{B_n}^{B_{n-1}} d_p^{B_n} = t_{B_1}^S t_{B_2}^S t_{B_3}^S ....t_{B_n}^S d_{B_n}^S$$

$$u_p^{AB_1B_2...B_{n-1}B_n} = 1 - t_{B_1}^A t_{B_2}^{B_1} t_{B_3}^{B_2} ....t_{B_n}^{B_{n-1}} t_p^{B_n} (1 - u_p^{B_n})$$
$$= 1 - t_{B_1}^S t_{B_2}^S t_{B_3}^S ....t_{B_n}^S t_{B_n}^S (1 - u_{B_n}^S)$$

Based on the opinion result, we introduce another parameter v that represents the final score of a specific message transaction. The higher value of v, the higher trust of this message flow. We define v in the following:

$$v = t + (1/2)u - 2d$$

Peer opinion inside SOS is dynamically updated according to the behavior. It either improves in some rate according to the contribution to the overlay (e.g. act as the intermediate peer to relay the traffic, etc) or degrades when do not make contribution or experience malicious behavior.

We define the encouraging rule: when the peer acts as an intermediate peer that relays the data traffic, the trust t increases and the uncertainty u decreases:

$$t = t_{prev} + (1/200)d_{prev} + (1/200)u_{prev}$$
$$d = d_{prev} * (199/200) \text{ and}$$
$$u = u_{prev} * (199/200)$$

We also define the punishment rule: if the peer does not contributes in the time period (e.g. 10 minutes, etc), the opinion value should be degraded as:

$$d = d_{prev} + (1/50)t_{prev} + (1/50)u_{prev}$$
$$t = t_{prev} * (49/50) \text{ and}$$
$$u = u_{prev} * (49/50)$$

Note that $\{t_{prev}, d_{prev}, u_{prev}\}$ denotes previous opinion and $\{t, d, u\}$ represents newly opinion.

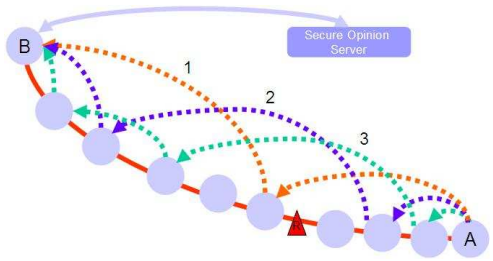

Fig.4.Secure Opinion Server system model

3.5. STUN/TURN/ICE-based NAT Traversal

Network Address Translators (NAT) provides benefits as well as drawbacks. One main drawback is that NAT is not friendly for connection establishment between two endpoints. In order to solve this problem, STUN/TURN/ICE-based approaches[11, 12] are proposed. STUN approach uses a STUN server in the middle of two endpoints to learn the NAT status (e.g. existence of NAT, NAT type, public endpoint address, port, etc). With these information, two endpoints might be able to establish the session directly. However, STUN approach does not work in symmetric NAT where each connection is mapped to a specific IP address and port. One possible solution is to use a TURN server to relay data traffic during the connection and transmission. ICE combines the usage of STUN and TURN approaches. It firstly selects STUN for handling, while turns to TURN if STUN is not available. Besides, ICE supports the negotiation of session establishment (e.g. latency, jitter measurement, error handling, best route, etc) between two endpoints.

3.6. Message Routing

We suggest Semi-recursive routing in our system architecture (illustrated in Fig.5). In this routing style, intermediate peers forward request step by step until the destination peer is reached. The corresponding response is returned directly from the destination to the source. This approach reduces the total number of the message transmitted than the other routing styles (e.g. iterative routing, recursive routing, etc[13]).

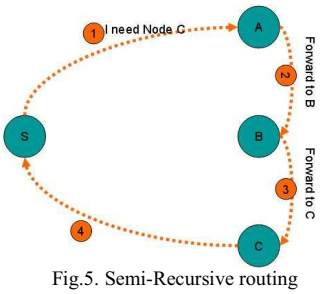

Fig.5. Semi-Recursive routing

4. Use Scenarios

In the following subsections we demonstrate the using of the proposed architecture for two typical use scenarios: intra-layer session initiation and inter-layer session initiation. We use "INVITE" (similar to the traditional SIP message) as the P2PSIP request and "180 Ringing" as the P2PSIP response.

4.1. Use Scenario 1

Use Case 1 (see Fig.6) describes intra-layer P2PSIP session initiation process between source peer A and destination peer B. Possible message flows are:

1) Source peer sends P2PSIP "INVITE" message to the CSP that is clockwise nearest to the destination peer.
2) CSP multicasts a "PingRequest" to a few successors. Intermediate peers forwards the "PingRequest" step by step until the destination.
3) Destination peer receives several identical "PingRequest". It asks SOS server via sending all possible routes. After trust calculation, OS replies with a best route.
4) Destination peer returns a "PingResponse" to CSP.
5) CSP forwards original P2PSIP "INVITE" message to the destination peer.
6) Destination peer returns a P2PSIP "180 Ringing" to source peer.

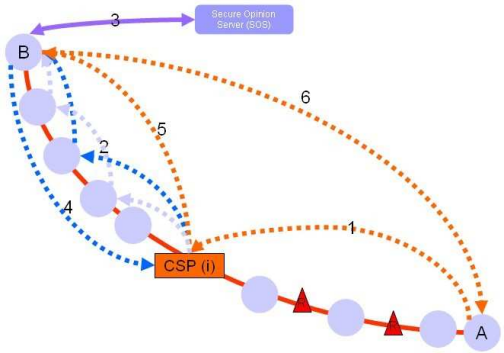

Fig.6. Intra-layer session initiation

4.1. Use Scenario 2

Use case 2 (see Fig.7) describes inter-layer P2PSIP session initiation process between source peer A and destination peer B. Possible messages flows are:

1) Source peer sends P2PSIP "INVITE" message to the CSPG in its sub-overlay.
2) CSPG forwards "INVITE" to another CSPG in destination sub-overlay.
3) The "INVITE" is forwarded to the CSP that is clockwise nearest to the destination peer.
4) CSP multicasts a "PingRequest" to a few successors. Intermediate peers forwards the "PingRequest" step by step until the destination.
5) Destination peer receives several identical "PingRequest". It asks SOS server via sending all possible routes. After trust calculation, OS replies with a best route.
6) Destination peer returns a "PingResponse" to CSP.
7) CSP forwards original P2PSIP "INVITE" message to the destination peer.
8) Destination peer returns a P2PSIP "180 Ringing" to source peer.

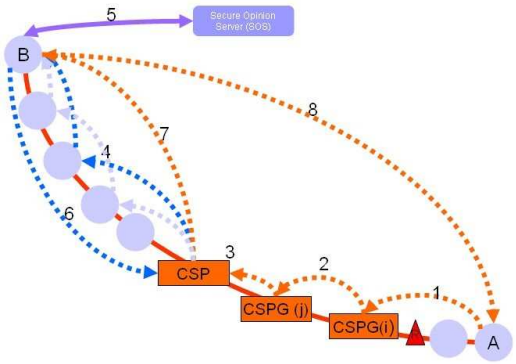

Fig.7. Inter-layer session initiation

5. Evaluation

In this Section, we first describe the prototype implementation and then analyze the system architecture in mainly two aspects (number of hop and delay, security) to show that the proposed system is feasible and secure.

5.1. Prototype Simulation

We simulate the proposed system architecture with corresponding solutions by implementing the prototype in Java. The prototype contains 512 peers (including 496 normal P2PSIP peers, 13 CSP peers, and 3 CSPG peers). We set the overlay space size as 2048, in which [0, 255] represents the top layer sub-overlay; [256,1023] represents the second layer sub-overlay; and [1024,2047] is the lowest layer sub-overlay. Apache Derby is selected as the embedded database implementation for P2PSIP peers, CSPs, and CSPGs.

Besides, we also build a Secure Opinion Server, which is a web server for storing and handling dynamic opinion calculation. The SOS uses Apache Derby as the opinion database, and Apache tomcat as the background HTTP container.

The system is deployed on a platform with Windows XP professional system, 2*2.4G Intel Core CPU, 3G memory, and 100Mbps Ethernet connection. We define "INVITE" as the P2PSIP request and "180 ringing" as the response (See Fig.8 and Fig.9). Note that all the messages sending and receiving are based on TCP.

We use the Wireshirk [14] to monitor the message transmission. The testing shows that the overlay system works well.

```
INVITE 20 P2PSIP/2.0
Max-Forwards:10
From:3
To:20
Call-ID:472721
CSeq: 1 INVITE
Contact:3
Via:3 127.0.0.1:9003;
```

Fig.8. P2PSIP INVITE

```
P2PSIP/2.0 180 Ringing
To:20
From:3
Contact:20
CSeq: 1 Response
Content-Length: 0
Via:3 127.0.0.1:9003;
20 192.168.0.101:9020;
```

Fig.9.P2PSIP 180 Ringing

5.2. Number of hop and delay

We assume that the number of peers and CSPs in the overlay is N and S respectively, where N_1, N_2, N_3 are the number of peers in each sub-overlay from top to bottom and S_1, S_2, S_3 are number of CSPs in each suboverlay from top to bottom. Besides, we assume that peer in communicate with the other peer in the same suboverlay in a probability of p_1, p_2, p_3. Also, we assume that peers and CSPs are evenly distributed in the overlay space.

Based on Chord routing protocol[15], the average num-of-hop of "pingRequest" multicast (for example, Step 2 in Fig.3) is $\frac{1}{2}\log(N_i/S_i)$, where i represents each suboverlay. Therefore, the complexity of intra-suboverlay is $(1/2)\log(N_i/S_i)+1$ due to the addition of one CSP (See Step 1 in Figure 5); the complexity of inter-suboverlay is $(1/2)\log(N_i/S_i)+3$ due to addition of two CSPGs and one CSP (see Step 1-3 in Fig.3).

According to the Mean rule, the average hop number is:

$$\sum_{1}^{3}\{p_i*(1+\frac{1}{2}\log(N_i/S_i))+(1-p_i)*(3+\frac{1}{2}\log(N_i/S_i))\}$$
$$=3-\frac{2(p_1N_1+p_2N_2+p_3N_3)}{N}+\frac{1}{2}\log_2(\frac{N}{S})$$

After that, we assume $p_1=p_2=p_3=0.8$, based on the concept that most communication session are geographically related to each other (in Section 3.3). Therefore, the average num-of-hops is:

$$1.4+\frac{1}{2}\log_2(\frac{N}{S})$$

Fig.10 shows the improved result (we set S=16 and S=32 separately) compare with conventional chord-based system. We get the conclusion that our proposed hierarchical division is more efficient than conventional Chord lookup algorithm. Besides, the more S, the better lookup efficiency will be.

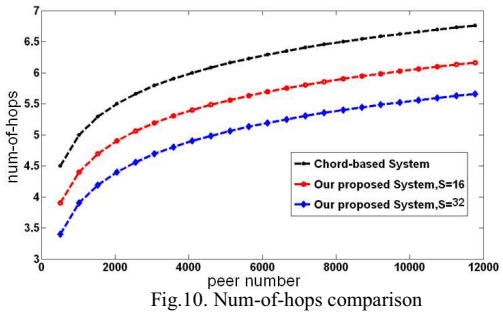

Fig.10. Num-of-hops comparison

5.3. A malicious Attack Analysis

We use a typical malicious use scenario implementation to show both of trust upgrading and the protection of the networks from compromised or malicious intermediate peers.

We initiate a P2PSIP request from peer 80, searching for destination peer 1618. Then, we assume the intermediate peer 1617 is a malicious/compromised intermediate peer that might discard, misroute, revise or temper the data message. This makes the message flow unavailable to reach destination peer (based on the conventional Chord routing: Peer 80 -> peer 1331-> peer 1593 -> peer 1609->peer 1617 ->peer 1618).

However, the situation is different in our system. The request would be directed to CSPG 1, CSPG 1030, and then CSP 1536. After that, "PingRequest" is multicasted and therefore causes several routes. Although one of the routes is interfered by malicious peer 1617 (the read route in Figure 11), two others can still reach the destination peer. Finally, the destination peer asks Secure Opinion Server (SOS) via sending HTTP "asking", asking for the best route.

We assume that in a certain period, the opinion of each related peer are: peer 1593 (0.8, 0.1, 0.1), peer 1600 (0.82, 0.08, 0.08), peer 1609 (0.92, 0.04, 0.04), peer 1618 (0.9, 0.05, 0.05).

We simulate this by manually modifying the opinion database. According to the description of Section 3.6, the opinion of two routes is:

$\omega_p^1=\{0.264,0.012,0.724\}$ with $v=0.605$

$\omega_p^2=\{0.167,0.005,0.826\}$ with $v=0.570$

After the opinion calculation, SOS returns the most trustful route to the destination peer 1618. And the session can be established in the most trustful situation.

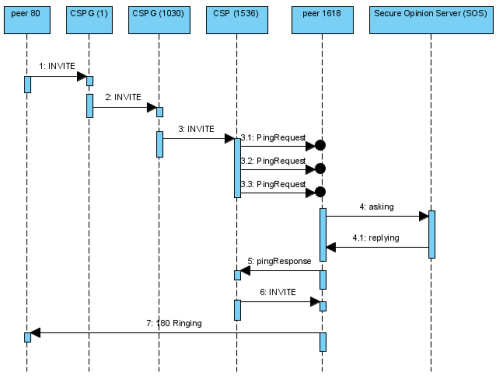

Fig.11. A typical malicious uses scenario

6. Conclusion and Future work

In this paper we propose a possible efficient and secure model for P2PSIP communication systems.The system model resolves several issues including three sub-overlay division, identifier assignment, cache mechanism, proxy based security, subjective logic based trust enhancement, NAT traversal, and message routing. These approaches improve the peer/resource lookup efficiency in P2PSIP session establishment and protect the system from security breaches, such as malicious or faulty intermediate peers.

However, portable peers (in the 3rd sub-layer of proposedmodel) are vulnerable in front of many security breaches (e.g. maleware, trojan, etc)due to lack of security protection or resource limitation, which result of endpoint (platform/system) integrity problem. The future work plans to import the concept of trust computing technology[16], especially TPM module and DAA authentication protocol in the security enhancement of portable peers in the overlay.

Besides, It is also necessary to study the extension function of CSP for legacy portable devices (e.g. mobile phone, etc) that lacks the capability to access P2PSIP overlay due to limited protocol support or other limitation in device capabilities (e.g. available computing, bandwidth, etc). A thin-client based system architecture proposed in [17]might be a possible solution.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61103175, the Key Project Development Foundation of Education Committee of Fujian province under Grand No. JA11011, the Technology Innovation Platform Project of Fujian Province under Grant No.2009J1007.

References

1. *P2PSIP.* p. http://www.p2psip.org (accessed Nov 2011).
2. Bryan, D.A., B.B. Lowekamp, and M. Zangrilli, *The Design of a versatile, secure P2PSIP communications architecture for the public internet*, in *IEEE international Symposium on Parallel and Distributed Processing*April, 2008, IEEE: Miami, USA. p. 1-8.
3. Jennings, C., et al., *REsource LOcation And Discovery (RELOAD) base Protocol.* IETF Internet Draft (draft-ietf-p2psip-base-19), Oct, 2011: p. http://tools.ietf.org/html/draft-ietf-p2psip-base-19 (accessed Oct 2011).
4. Eronen, P. and H. Tschofenig, *Pre-Shared Key Ciphersuites for Transport Layer Security (TLS).* IETF RFC 4279, Dec, 2005.
5. Zheng, X. and V. Oleshchuk, *Secure Interworking with P2PSIP and IMS*, in *The 2010 International Symposium on Collaborative Technologies and Systems (CTS 2010)*May, 2010, IEEE: Chicago, USA. p. 481-488.
6. Zheng, X. and V. Oleshchuk, *Providing Privacy Service for P2PSIP based Communication Systems*, in *Norsk informasjonssikkerhetkonferanse (NISK)*Nov, 2008: Kristiansand, Norway.
7. Zheng, X. and V. Oleshchuk, *Trust-based Framework for Security Enhancement of P2PSIP Communication Systems*, in *The 4th International Conference for Internet Technology and Secured Transactions (ICITST-2009)*Nov, 2009, IEEE: London, UK. p. 1-6.
8. Le, L. and G.-S. Kuo, *Hierarchical and Breathing Peer-to-Peer SIP System*, in *IEEE International Conference on Communcations (ICC'07)*2007, IEEE: Glasgow, Scotland. p. 1887-1892.
9. Shi, G., et al., *T2MC: A Peer-to-Peer Mismatch Reduction Technique by Traceroute and 2-Means Classification Algorithm.* Networking 2008Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, 2009. **4982**: p. 366-374.
10. Josang, A., R. Hayward, and S. Pope, *Trust network analysis with subjective logic*, in *Proceedings of the 29th Australasian Computer Science Conference*2006, Australian Computer Society, Inc.: Hobart, Australia. p. 85-96.
11. Rosenberg, J., et al., *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs).* IETF RFC 3489, March 2003.
12. Stukas, M. and D.C. Sicker, *An Evaluation of VoIP Traversal of Firewalls and NATs within an Enterprise Environment* Information Systems Frontiers, Sept, 2004. **6**(3): p. 219-228.
13. Zong, N., et al., *An Extension to RELOAD to support Direct Response Routing.* IETF Internet Draft (draft-ietf-p2psip-drr-01), Nov, 2011.
14. *Wireshark . Go deep.* p. http://www.wireshark.org/ (accessed Nov 2011).
15. Stoica, I., et al., *Chord: a scalable peer-to-peer lookup protocol for internet applications.* IEEE/ACM Transactions on Networking, 2003. **11**(1): p. 17-32.
16. Kagal, L., T. Finin, and A. Joshi, *Trust-based security in pervasive computing environments.* Computer, Dec 2001. **34**(12): p. 154-157.
17. Zheng, X., V. Oleshchuk, and H. Jiao, *A System Architecture for SIP/IMS-based Multimedia Services* in *Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics*Dec, 2007, Springer. p. 543-548.