# Service Re-routing for Service Network Graph: Efficiency, Scalability and Implementation

David Lai and Zhongwei Zhang

University of Southern Queensland
Toowoomba, Queensland, 4350
lai@usq.edu.au zhongwei@usq.edu.au

## Abstract

The key to success in Next Generation Network is service routing in which service requests may need to be redirected as in the case of the INVITE request in Session Initiation Protocol [21]. Service Path (SPath) holds the authentication and server paths along side with service information. As the number of hops in a redirection increases, the length of SPath increases. The overhead for service routing protocols which uses SPath increases with the length of SPath. Hence it is desirable to optimize SPath to ensure efficiency and scalability of protocols involving service routing. In this paper, we propose a re-routing strategy to optimize service routing, and demonstrate how this strategy can be implemented using SPath to enhance the efficiency and scalability of Service Network Graph (SNG).

**Keywords**
Service Routing, Service Path, Service Network Graph, Optimization, Authentication Delegation

## 1. Introduction

As a key feature of service routing, service routing models and architectures such as Semantic Overlay Based service Routing [7] or Session Initiation Protocol (SIP) [21, 22] requires redirection of service requests. The service request redirection can be accomplished with multiple redirections of only one hop each. Service Network Graph (SNG), a remote authentication protocol, requires redirection of a service request using single redirection via multiple hops.

During service redirection, SPath was proposed to hold the service path and service information. As the service network grows and redirection path gets longer, SPath may become unmanageable.

The overhead for establishing authentication and service access will escalate. This makes SPath not scalable. As a result we have to optimize the SPath as the service network grows. In this paper, we introduced a strategy to optimize the SPath for higher efficiency and better scalability.

A formal justification of the optimization using the symbols and approach presented by Lampson in his paper [17] is presented. This paper starts with a review on Service Network Graph in Section 2. In Section 3, we introduce some axioms and theorems established by Lampson in [17]. In Section 4 we briefly introduce the format of SPath and present our proposition. We prove the proposition by proposing and proving four lemmas. In Section 5, a detailed discussion on the implementation of SPath optimization in an SNG environment is presented. In the Conclusion section, we summarize our work in this paper and our work in the future.

## 2. Overview of Service Network Graph

With the onset of Globalization of world economy, geographical location can no longer confine oneself to a particular community. But the reality is we are confined to use services provided by our home network and we cannot access services offered in different autonomous networks may due to the fact that we are not aware of the services; or we do not know how to access them; or we are simply not allowed to access them. It would be desirable if one network can join another network and share their services to their home users. Under this scenario, one of the immediate problems is how to authenticate users of the participating networks. Issues such as information privacy, network platforms and resources make the sharing of user authentication information of all participating networks prohibitively hard or difficult.

To tackle the issues, the use of X.509 certificates [1], trust recommendations [4, 8, 18, 20] trust establishment [6, 19, 2, 3, 5] and Kerberos [9] are developed. Nevertheless, none of them is widely accepted as a viable solution to the problem. We first proposed Service Network Graph (SNG) in 2005 [16, 12, 11] and extended SNG to mobile users in [15]. SNG enables the linking of heterogenous networks in an ad hoc manner to form a Service Network Graph. Within the service network graph, home users of individual networks can share the services provided by other networks within SNG. To enhance the security of the authentication process, SNG can include Dynamic Password [10] as one of its authentication scheme, and thereby forming an authentication protocol suite for heterogenous aggregation of ad hoc networks.

With its service re-routing features, we will use SNG as the environment for our discussion. A brief review of SNG and its mechanism would facilitate our discussion presented later. To participate in an SNG, the authentication server AS1 of Network1 (N1), is required to share a secret key with the authentication server AS2 of Network 2 (N2) which is part of an SNG as shown in Figure 1. A self authenticating encryption channel [14] is set up between two joined networks. Communications between authentication servers are protected by encryption using the shared key. Suppose N2 offers service Srv2. When the service Srv2 is shared with N1, we need to indicate that this service is offered by N2. This can be done with the Service Access Path (SAPath) field in a Service Path (SPath). Obviously, the SAPath in N2 is simply the address of N2 while SAPath of the same service in N1 must include the address of Network 1 and the address of N2. When other networks join in, we may have an SNG as shown in Figure 2. The SAPath field of Srv2 in Network 3 (N3) should include the addresses of N2, N1 and N3.

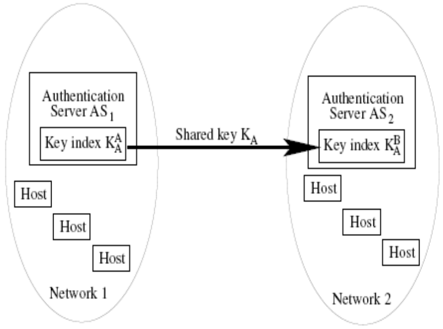


Figure 1. Network 1 joins Network 2 in an SNG

## 3. Basic Axioms and Theorems

In this Section, we will present some of the Axioms and Theorems established in [17]. The symbols used are listed below:

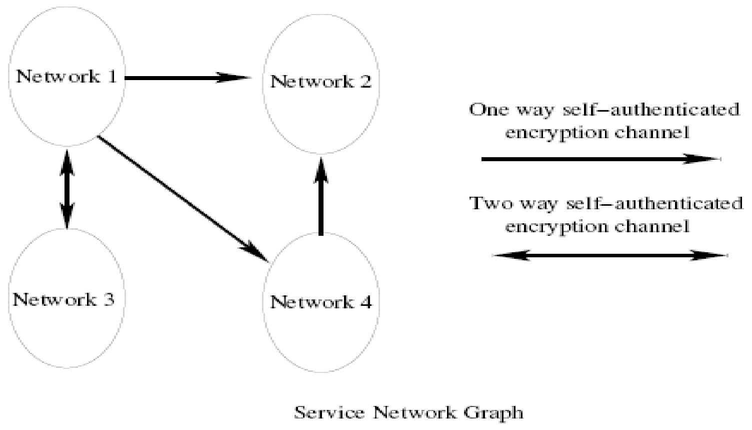

Network 1
Network 2
Network 3
Network 4
One way self–authenticated encryption channel
Two way self–authenticated encryption channel
Service Network Graph

**Figure 2. Graphical representation of an SNG**

**Table 1. List of symbols used in this paper**

| Symbol | Meaning |
|---|---|
| $s$ | a statement |
| $\vdash s$ | $s$ is an axiom of the theory or<br>$s$ is provable form the axioms. |
| $\looparrowright$ | speak for |
| $\Rightarrow$ | imply |
| $\wedge$ | and |

**Axiom 1** $\vdash (P_A \text{ says } (P_B \looparrowright P_A)) \Rightarrow (P_B \looparrowright P_A)$

This Axiom says that principal PA can establish a “speak for” relationship with principal PB when he declares PB speaks for him.

**Theorem 1** $\vdash (P_A \looparrowright P_B) \Rightarrow ((P_A \text{ says } s) \Rightarrow (P_B \text{ says } s))$

This theorem tells us that if principal PA speaks for principal PB, then whenever PA says something, PB would have said the same thing.

In this paper, we are concerned with authentication authority of a principal. Hence we will use a qualified “speak for” relation. When we qualify “speak for” relationship with the role “as Authentication Agent” (“as AA” for short), we have a qualified version of Theorem 1 as shown in Theorem 2 below.

**Theorem 2** $\vdash (P_{A\ as\ AA} \looparrowright P_{B\ as\ AA}) \Rightarrow ((P_{A\ as\ AA} \text{ says } s) \Rightarrow (P_{B\ as\ AA} \text{ says } s))$

**Theorem 3** $\vdash ((P_C \looparrowright P_A) \wedge (P_C \text{ says } (P_B \looparrowright P_A))) \Rightarrow (P_B \looparrowright P_A)$

This is called the HandOff rule by Lampson et al in [17]. In this theorem, the first condition requires $(P_C \looparrowright P_A)$. Using Theorem 1, whatever $P_C$ says, $P_A$ would have said the same. Hence the second condition of Theorem 3 can be rewritten as $(P_A \text{ says } (P_B \looparrowright P_A))$. Using Axiom 1, if $P_A$ declares $P_B$ speaks for him, we can arrive at the conclusion $(P_B \looparrowright P_A)$.

## 4. Optimization of Service Path (SPath)

In this section we will prove that Service Paths can be optimized and in the next section, we will show how to implement the optimization of SPaths.

To illustrate our discussion, we will use a freely sharable FTP service provided by network NA for a cost of 215 units as an example. The authentication server for NA has an IP address of 10.1.1.1. The server providing the service is called FTPSer.

### 4.1. Format of SPath

When network NA offers a service, the service is listed as an SPath of the form

$$< SOpt : SAPath/Ser/Srv >:< C >$$

where
SOpt: Sharing Option
SAPath: Service Access Path
Ser: Name of Server
Srv: Name of service
C: Cost for using the service

The SAPath field in this case is simply the network address of the authentication server (AS) of NA.

$$< SOpt : add_{N_A}/Ser/Srv >:< C >$$

So the SPath of our example FTP service listed in the service providing network NA looks like:

< F : 10.1.1.1/FTPSer/FTP >:< 215 >

When network NB joins an SNG by attaching to network NA, NA delegates its authentication authority to NB. NA will also pass the SPath of the FTP service to NB. Home users of NB can now use the FTP service offered by NA if they are authenticated by NB. NB will list all the shared service as SPaths by pre-pending the address of its authentication server to the SAPath fields of all SPaths shared by NA.

< SOption : addNB/addNA/Ser/Srv >:< Cost >
The SPath for FTP service in NB looks like:
< F : 10.1.2.1/10.1.1.1/FTPSer/FTP >:< 215 >

As the SAPath field will be pre-pended with a network address every time it is shared with another network, the SAPath of an SPath gets longer each time the service is shared with another network. When users try to authenticate and access a service, the overhead for authentication and setting up a service gets larger as the SAPath gets longer. It is imperative to keep the SAPath to an optimal length for both efficiency and scalability. To optimize an SPath is the transformation of the SAPath field of an arbitrary SPath to its optimal form. In the next subsection, we will discuss the theoretical basis of optimizing SPaths.

### 4.2. Optimization of SPath

We will start to prove that SPaths (SAPath field) can be optimized with some definitions regarding Authentication Delegation in SNG context.

### Definition 1 Authentication Delegation

If network NA attaches to network NB which is a member of an SNG, then we define NB delegates its authentication authority to network NA.

If a network NB delegates its authentication authority to another network NA, then we represent it as

$$N_{B\ as\ AA}\ \text{says}\ (N_{A\ as\ AA}\ \looparrowright\ N_{B\ as\ AA})$$

This formalized the definition of Authentication Delegation in an SNG. When authentication authority is delegated, we have to keep track of the delegatee and the delegator relationships. They are recorded in Authentication Delegation Paths. Every time when a delegation occurs, the new delegatee address is pre-pended to the Authentication Delegation Path. So an Authentication Delegation Path would have the address of the delegatee network as the leftmost address and the delegator network address as the right most address. In between are intermediate networks which were delegatee networks at certain time in the authentication delegation process.

**Definition 2 Authentication Delegation Path for Self-Authentication**
The Authentication Delegation Path of network NA in network NA itself is defined as:
addNA/
It simply means NA performs authentication itself.

**Definition 3 Authentication Delegation Path in Remote Networks**
If NA delegates its authentication authority to another network NB, then we define the Authentication Delegation Path for NA in NB to be
addNB/addNA/

Within the SNG context. The delegated authentication authority can further be delegated. That is to say, if NA delegates authentication authority to NB which in turn delegates the authentication authority of NA to another network NC, the Authentication Delegation Path looks like

addNC/(addNB/addNA/) which is equivalent to
addNC/addNB/addNA/
Hence we can generalize our Authentication Delegation Path definition to the following definition.

**Definition 4 Authentication Delegation Path**
The Authentication Delegation Path is defined as the network path which traces the authentication delegation sequence from the delegator network to the final delegatee network in the form of

addNdelegatee/.../addN2/addN1/addNdelegator/

With the definitions in place, we can now make the proposition that SPaths can be optimized.

**Proposition 1 (Optimization of SPath)**

Service Path of the form
< SOpt : SAPath/Ser/Srv >:< Cost >
can always have the SAPath optimized to a two-address format
addNhome/addNservice/
and the resulting SPaths have the form
< SOpt : addNhome/addNservice/Ser/Srv >:< Cost >
We will prove this proposition by working through a sequence of Lemmas.

**Lemma 1 (Transitivity of "speak for" relation)**

$$\vdash (N_{A\ as\ AA} \looparrowright N_{B\ as\ AA}) \wedge (N_{B\ as\ AA} \looparrowright N_{C\ as\ AA}) \Rightarrow (N_{A\ as\ AA} \looparrowright N_{C\ as\ AA})$$

**PROOF:**
From the first condition in the Lemma and Theorem 2, we have

$$(N_{A\ as\ AA}\ \mathsf{says}\ s) \Rightarrow (N_{B\ as\ AA}\ \mathsf{says}\ s)$$

Similarly, the second condition in the Lemma yields

$$(N_{B\ as\ AA}\ \text{says}\ s)\ \Rightarrow\ (N_{C\ as\ AA}\ \text{says}\ s)$$

So the predicate of the logic becomes:

$$((N_{A\ as\ AA}\ \text{says}\ s)\ \Rightarrow\ (N_{B\ as\ AA}\ \text{says}\ s))\ \wedge\ ((N_{B\ as\ AA}\ \text{says}\ s)\ \Rightarrow\ (N_{C\ as\ AA}\ \text{says}\ s))$$

Transitive property of the “)” relation allows us to replace the predicate with

$$((N_{A\ as\ AA}\ \text{says}\ s)\ \Rightarrow\ (N_{C\ as\ AA}\ \text{says}\ s))$$

which is precisely what we will get when we apply Theorem 2 to the conclusion of the Lemma.

**PROOF**

When NA delegates authentication authority to NB, by Definition 1 and Axiom 1, we have

$$(N_{B\ as\ AA}\ \looparrowright\ N_{A\ as\ AA})$$

When NB delegates authentication authority to NC, by Definition 1 and Axiom 1, we have

$$(N_{C\ as\ AA}\ \looparrowright\ N_{B\ as\ AA})$$

These two authentication delegations satisfied the conditions of Lemma 1 and so we can conclude from Lemma 1 that

$$(N_{C\ as\ AA}\ \looparrowright\ N_{A\ as\ AA})\ \blacksquare$$

**Lemma 3 (Authentication Delegation Path)**

If NA as AA delegates its authentication authority to another network NB as AA, then NB will have the Authentication Delegation Path for NA and all Authentication Delegation Paths NA has with the address of NB pre-pended:

addNB/addNA/.../addN3/addN2/addN1/

**PROOF**

When N1 delegates its authentication authority to N2, by Definition 3, N2 will have a Authentication Delegation Path

addN2/addN1/

Similarly, when N2 delegates its authentication authority to N3, from Lemma 2, Definition 3 and Definition 4, N3 will have two Authentication Delegation Paths

addN3/addN2/

addN3/addN2/addN1/

We keep on applying Lemma 2 and Definition 3 and 4 every time a network delegates its authentication authority to another network, until, finally NA delegates its authentication authority to network NB. From Lemma 2, all authentication authority already delegated to NA, and the authentication authority of NA itself, will be delegated to NB. When authentication authorities are delegated to NB all the Authentication Delegation Paths will have the address of NA pre-pended by Definition 3.

**Lemma 4 (Equivalence of Authentication Delegation)**

Authentication Delegation Path of the form

addNhome/.../addN3/addN2/addN1/addNservice/

is equivalent to

addNhome/addNservice/

**PROOF**

Authentication Delegation Path

addNhome/.../addN3/addN2/addN1/addNservice/
indicates Nservice delegates its authentication authority to N1 (Definition 4) which in turn delegates its authentication authority to N2. From Lemma 2, the authentication authority for Nsevice will also be delegated to N2. By Definition 4 the Authentication Delegation Path of Nservice in N2 is
addN2/addN1/addNservice/
When the authentication authority for Nsevice is delegated to N2, using Definition 1, we have

$$N_{service\ as\ AA}\ \mathrm{says}\ (N_{2\ as\ AA} \looparrowright N_{service\ as\ AA})$$

By Axiom 1, we have

$$(N_{2\ as\ AA} \looparrowright N_{service\ as\ AA})$$

And by Definition 3, the Authentication Delegation Path for the delegation listed above is

addN2/addNservice/
Hence we can transform Authentication Delegation Path from
addN2/addN1/addNservice/
to a shorter form
addN2/addNservice/
Every time we apply the argument to the address triplets on the left hand side of an Authentication Delegation Path, we will get one address less. By repeating the process just described to an Authentication Delegation Path argument, we can arrive at its optimized form:
addNhome/addNservice/

**PROOF of Proposition 1**
SAPath inside a SPath is the authentication Delegation Path of the service to the user's home network. Hence by Lemma 4, all Authentication Delegation Path can be reduced to the form
addNhome/addNservice/
and hence we have the optimized form of an SPath.

## 5. Implementing SPath Optimization

In this section, we will discuss how to achieve the SPath optimization in an SNG context.

For heterogeneous aggregation of networks in an SNG, it is common to have a service shared with many neighboring networks. The shared service may, in turn, shared with more next neighbors. When a shared service is made available to a network, it may come with a different Service Path even though the service is provided by same unique server.

When the Service Path is optimized, information about the Authentication Path is lost. Further more, if a network withdraws from the SNG, all the Service Paths it shared with other networks will be invalidated. The optimized form of SPath does not reveal the intermediate networks involved. The same problem occurs when a networks re-joins an SNG, members of the SNG have to determine which SPaths will be validated.

We will discuss these issues along with the implementation methodology using Figure 3.

### 5.1. Selecting Service Paths

Figure 3 shows three possible ways to obtain a service. The SPath are:
< SOpt : $N_H/N_3/N_1/N_S$/Ser/Srv >:< C1 >
< SOpt : $N_H/N_3/N_4/N_1/N_S$/Ser/Srv >:< C2 >
< SOpt : $N_H/N_2/N_S$/Ser/Srv >:< C3 >
Obviously the three SPath will optimized to the same SPath:
< SOpt : $N_H/N_S$/Ser/Srv >:< C4 >

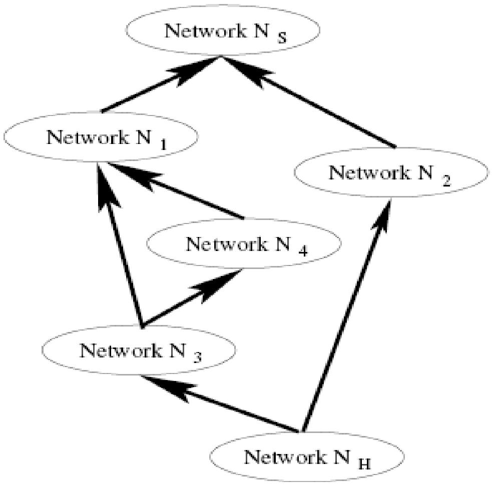


Figure 3. Sharing of key before SPath optimization

Note that the choice for the Cost id the minimum of all SPath Costs. In this case, C4 is the minimum of C1, C2 and C3.

Optimization summarizes SPaths of the same service with different service access paths to a single SPath. This helps to minimize the resources required to store and process the SPaths for service routing and provides a more scalable way of deployment.

### 5.2. Service Optimization Table and Authentication Path Network Lookup Table

We have seen that three distinct Spath for the same service can be optimized and form a single SPath for the service. It is true that by optimizing the SPath, the SPath is more efficient and scalable. On the other hand, the lost of detail SPath information results in two problems.

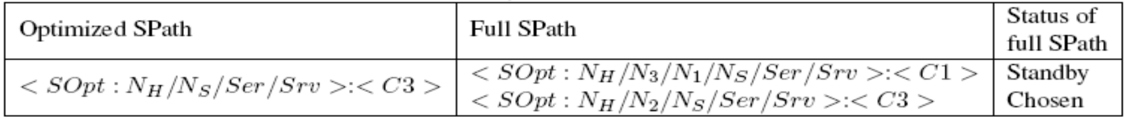
As the SPath serves as the path for the user home network to get service information, the optimized SPath does not have enough information for the user home network to reach the server for service information. Secondly, if a network stops to participate in an SNG, the Authentication Delegation chain is broken and all the SPaths that involve the network will no longer be valid and should be removed from the Service List. With the SPaths optimized, there is no way to tell which SPath is to be removed from the Service List. The problems can be solved if

- the SPath is shared in full form
- the SPath in full form and in optimized form are recorded in a SPath Optimization Table (SOT)
- the optimized form is listed in the Authentication Path Network Lookup Table (APNLT).

Consider the two SPaths in network NH:

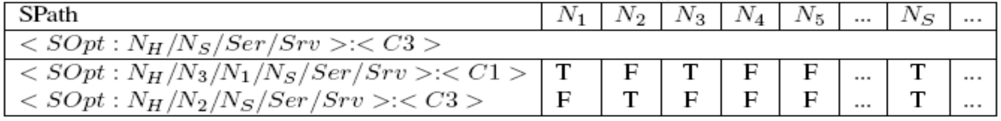
< SOpt : NH/N3/N1/NS/Ser/Srv >:< C1 > and
< SOpt : NH/N2/NS/Ser/Srv >:< C3 > which is optimized to the form
< SOpt : NH/NS/Ser/Srv >:< C3 >
Here we assume that C3 is less than C1 and C3 is chosen as the cost for the optimized SPath.
The SOT for network NH will look like Table 2 and Table 3.
The associated APNLT is:

Table 2. SPath Optimization Table for $N_H$



| Optimized SPath | Full SPath | Status of full SPath |
|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C3 >$ | $< SOpt : N_H/N_3/N_1/N_S/Ser/Srv >:< C1 >$<br>$< SOpt : N_H/N_2/N_S/Ser/Srv >:< C3 >$ | Standby<br>Chosen |

Table 3. Authentication Path Network Lookup Table for $N_H$



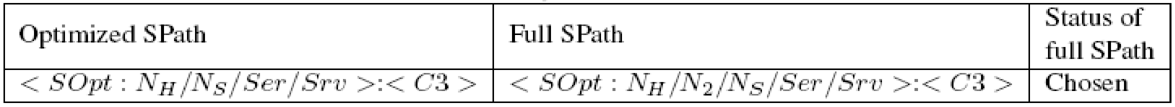
| SPath | $N_1$ | $N_2$ | $N_3$ | $N_4$ | $N_5$ | ... | $N_S$ | ... |
|---|---|---|---|---|---|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C3 >$ | | | | | | | | |
| $< SOpt : N_H/N_3/N_1/N_S/Ser/Srv >:< C1 >$ | T | F | T | F | F | ... | T | ... |
| $< SOpt : N_H/N_2/N_S/Ser/Srv >:< C3 >$ | F | T | F | F | F | ... | T | ... |

We can easily see that the Authentication Path for the optimized SPath is NH/N2/NS/ from SOT. APNLT provides us with a quick way to check if an optimized SPath is affected or not when a network withdraws from an SNG.

Suppose N1 withdraws from the SNG. From APNLT, column N1 we see that the entry at the row for

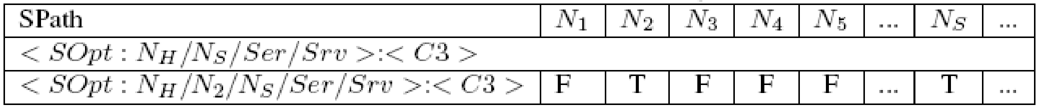
< SOpt : NH/N3/N1/NS/Ser/Srv >:< C1 >
is "T" and the entry at the row for
< SOpt : NH/N2/NS/Ser/Srv >:< C3 >
is "F". It means that
< SOpt : NH/N3/N1/NS/Ser/Srv >:< C1 >
is invalid now as it uses N1 as part of the SAPath while
< SOpt : NH/N2/NS/Ser/Srv >:< C3 >
is not affected as it does not use N1 in its SAPath. So SPath
< SOpt : NH/N3/N1/NS/Ser/Srv >:< C1 >
will be removed from the APNLT and SOT as shown in Table 4 and Table 5.

Table 4. SPath Optimization Table for $N_H$



| Optimized SPath | Full SPath | Status of full SPath |
|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C3 >$ | $< SOpt : N_H/N_2/N_S/Ser/Srv >:< C3 >$ | Chosen |

Table 5. Authentication Path Network Lookup Table for $N_H$

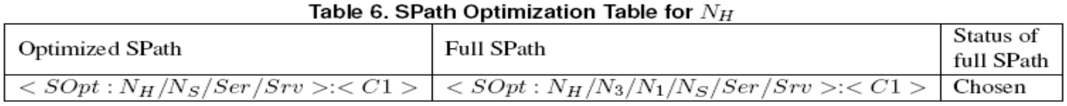


| SPath | $N_1$ | $N_2$ | $N_3$ | $N_4$ | $N_5$ | ... | $N_S$ | ... |
|---|---|---|---|---|---|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C3 >$ | | | | | | | | |
| $< SOpt : N_H/N_2/N_S/Ser/Srv >:< C3 >$ | F | T | F | F | F | ... | T | ... |

Now let us assume that N1 still stays in the SNG but N2 withdraws. Column N2 of APNLT has a "F" at the row for
< SOpt : NH/N3/N1/NS/Ser/Srv >:< C1 >
and "T" at the row for
< SOpt : NH/N2/NS/Ser/Srv >:< C3 >.
Now it is
< SOpt : NH/N2/NS/Ser/Srv >:< C3 >
which is invalid, and
< SOpt : NH/N3/N1/NS/Ser/Srv >:< C1 >
is not affected.

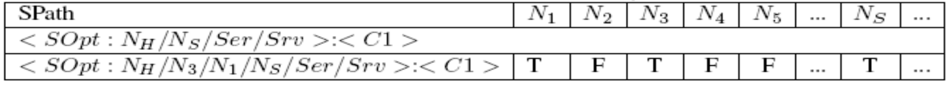
< SOpt : NH/N2/NS/Ser/Srv >:< C3 >
will be removed from the APNLT and SOT while
< SOpt : NH/N3/N1/NS/Ser/Srv >:< C1 >
in SOT will be promoted to Chosen as it provides the SPath with minimum cost. Note the the Cost for the optimized SPath changes to C1 as the SPath which has a Cost of C3 is no longer available. the updated SOT and APNLT are shown in Table 6 and Table 7.

**Table 6. SPath Optimization Table for $N_H$**



| Optimized SPath | Full SPath | Status of full SPath |
|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C1 >$ | $< SOpt : N_H/N_3/N_1/N_S/Ser/Srv >:< C1 >$ | Chosen |

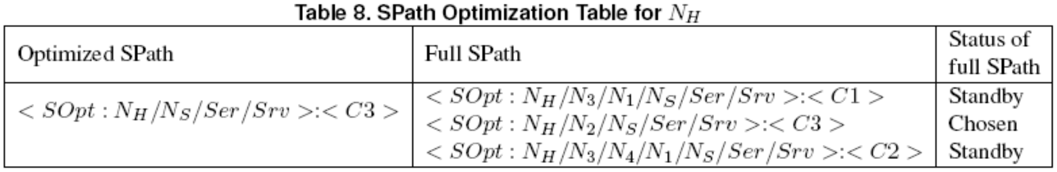
**Table 7. Authentication Path Network Lookup Table for $N_H$**



| SPath | $N_1$ | $N_2$ | $N_3$ | $N_4$ | $N_5$ | ... | $N_S$ | ... |
|---|---|---|---|---|---|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C1 >$ | | | | | | | | |
| $< SOpt : N_H/N_3/N_1/N_S/Ser/Srv >:< C1 >$ | T | F | T | F | F | ... | T | ... |

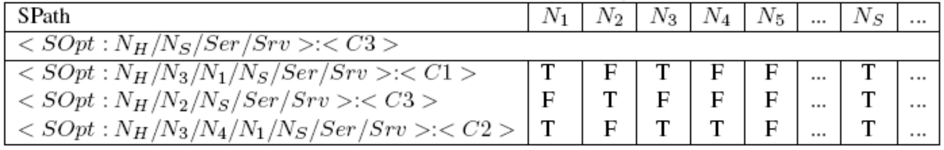
Consider another case when both N1 and N2 are staying in the SNG and a new SPath for the same
service NS/Ser/Srv is shared with NH:
< SOpt : NH/N3/N4/N1/NS/Ser/Srv >:< C2 >
If C2 is more than C3, the new SPath will become a backup SPath and assume a status of Standby as shown in Table 8 and Table 9.

**Table 8. SPath Optimization Table for $N_H$**



| Optimized SPath | Full SPath | Status of full SPath |
|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C3 >$ | $< SOpt : N_H/N_3/N_1/N_S/Ser/Srv >:< C1 >$ | Standby |
| | $< SOpt : N_H/N_2/N_S/Ser/Srv >:< C3 >$ | Chosen |
| | $< SOpt : N_H/N_3/N_4/N_1/N_S/Ser/Srv >:< C2 >$ | Standby |

**Table 9. Authentication Path Network Lookup Table for $N_H$**

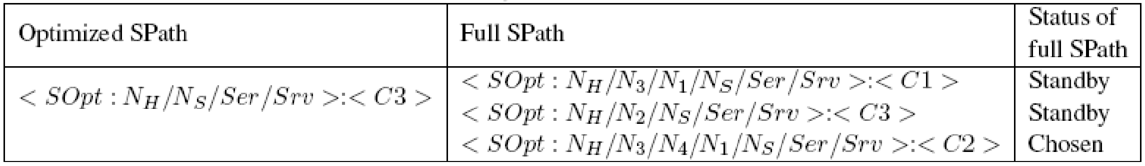


| SPath | $N_1$ | $N_2$ | $N_3$ | $N_4$ | $N_5$ | ... | $N_S$ | ... |
|---|---|---|---|---|---|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C3 >$ | | | | | | | | |
| $< SOpt : N_H/N_3/N_1/N_S/Ser/Srv >:< C1 >$ | T | F | T | F | F | ... | T | ... |
| $< SOpt : N_H/N_2/N_S/Ser/Srv >:< C3 >$ | F | T | F | F | F | ... | T | ... |
| $< SOpt : N_H/N_3/N_4/N_1/N_S/Ser/Srv >:< C2 >$ | T | F | T | T | F | ... | T | ... |

On the other hand, if C2 is less than C3, the new SPath will become the preferred SPath and assume a status of Chosen while

< SOpt : NH/N2/NS/Ser/Srv >:< C3 > will be down graded to Standby. The APNLT
remains the same as before, but SOT will be changed as shown in Table 10 and Table 11.

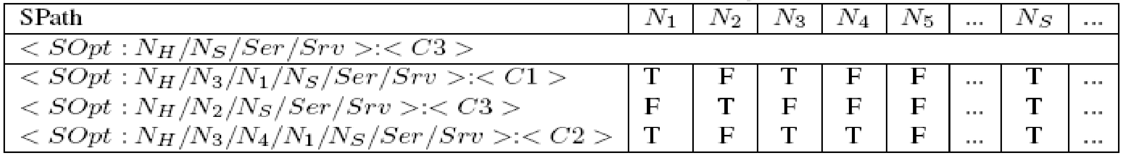
**Table 10. SPath Optimization Table for $N_H$**



| Optimized SPath | Full SPath | Status of full SPath |
|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C3 >$ | $< SOpt : N_H/N_3/N_1/N_S/Ser/Srv >:< C1 >$ | Standby |
| | $< SOpt : N_H/N_2/N_S/Ser/Srv >:< C3 >$ | Standby |
| | $< SOpt : N_H/N_3/N_4/N_1/N_S/Ser/Srv >:< C2 >$ | Chosen |

In the extreme case when an optimized SPath has no valid full SPath in the SOT, the optimized SPath is no longer valid and can be removed from the SOT, APNLT and service list.

Table 11. Authentication Path Network Lookup Table for $N_H$



| SPath | $N_1$ | $N_2$ | $N_3$ | $N_4$ | $N_5$ | ... | $N_S$ | ... |
|---|---|---|---|---|---|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C3 >$ | | | | | | | | |
| $< SOpt : N_H/N_3/N_1/N_S/Ser/Srv >:< C1 >$ | **T** | F | T | F | F | ... | **T** | ... |
| $< SOpt : N_H/N_2/N_S/Ser/Srv >:< C3 >$ | F | **T** | F | F | F | ... | **T** | ... |
| $< SOpt : N_H/N_3/N_4/N_1/N_S/Ser/Srv >:< C2 >$ | T | F | T | T | F | ... | **T** | ... |

With SOT and APNLT in place, we can optimize SPath without loss of Authentication Path information and can check how the optimized SPaths are affected when a network withdraws from an SNG.

## 5.3. Authentication Re-Delegation

When network NA joins an SNG, it shares a secret key K1 with a member network, NB of the SNG for establishing a self-authenticating encryption channel. In NA the Authentication Delegation Path for NB is
addNA/addNB/
When another network NC links with NA to join the SNG, the key shared between NA and NC is K2. In NC the Authentication Delegation Paths are
addNC/addNA/
addNC/addNA/addNB/
Proposition 1 allows us to optimize the second Authentication Delegation Path to
addNC/addNB/
NC has a shared key with NA. NB has a shared key with NA and NB has no shared key with NC. The optimized SPath addNC/addNB/ works only when there is a shared key between NC and NB. The shared key will be used to establish a self-authenticating encryption channel between NC and NB. So NC must share a key K3 with NB before optimizing any SPath in which the service is provided by NB.

The sharing of a key between NB and NC is called Authentication re-Delegation from NB to NC given that NB has delegated authentication authority to NA and NA has delegated authentication authority to NC.

As the optimized Authentication Delegation Path indicates that NB has delegated the authentication authority to NC, NB would be willing to share a common key with NC and establish an encrypted channel. This can be done via the original encrypted Authentication Delegation Path or simply uses the same procedure as when NC initially links with NA. By sharing a key with NB, NC is now linked directly with NB. Figure 4 shows that NA shares a key K1 with NB and the key indices for K1 are KA 1 and KB 1 in NA and NB respectively. The shared key between NA and NC is K2. The key indices for K2 are KA 2 and KC 2 in NA and NC respectively. The implementation is valid for all SPaths which has the SAPath field optimized to the two-address format. addNC and addNB are now replaced by addNH and addNS . The home network has to initiate the sharing of a key with the service providing network. The addresses which appear in the original SAPath have no affect on the optimization process as shown in Figure 5.

## 5.4. Revocation of Authentication Delegation

At any point of time, if a network wishes to revoke its authentication delegation to a certain network,all it needs to do is to send a revocation message containing its own network identity as the delegator and the network identity of the delegatee network. Each member network of SNG will adjust its SOT and APNLT to reflect the revocation. Starting with SOT and APNLT as shown in Table 10 and Table 11, and assuming NH receives the following revocation message:

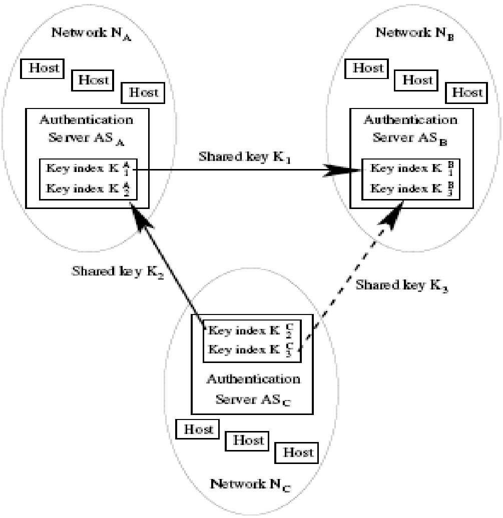


**Figure 4. Sharing of key before SPath optimization**

Delegator : N4
Delegatee : N3
Service : NS/Ser/Srv
NH will immediately look up SOT and APNLT, removing all entries of full SPaths that contains N3/N4 in their SAPaths; and has NS/Ser/Srv as the service and service provider. Note that only SPaths with the pattern N3/N4 in their SAPath are removed, SPaths with the pattern N4/N3 in their SAPath should not be removed as N4/N3 means
Delegator : N3
Delegatee : N4
and is not affected by the revocation of authentication delegation from N4 to N3.
Statuses of the remaining full SPaths in SOT will be updated as shown in Table 12 and Table 13.

**Table 12. SPath Optimization Table for** NH

| Optimized SPath | Full SPath | Status of full SPath |
|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C3 >$ | $< SOpt : N_H/N_3/N_1/N_S/Ser/Srv >:< C1 >$<br>$< SOpt : N_H/N_2/N_S/Ser/Srv >:< C3 >$ | Standby<br>Chosen |

**Table 13. Authentication Path Network Lookup Table for $N_H$**

| SPath | $N_1$ | $N_2$ | $N_3$ | $N_4$ | $N_5$ | ... | $N_S$ | ... |
|---|---|---|---|---|---|---|---|---|
| $< SOpt : N_H/N_S/Ser/Srv >:< C3 >$ | | | | | | | | |
| $< SOpt : N_H/N_3/N_1/N_S/Ser/Srv >:< C1 >$ | T | F | T | F | F | ... | T | ... |
| $< SOpt : N_H/N_2/N_S/Ser/Srv >:< C3 >$ | F | T | F | F | F | ... | T | ... |

## 6. Conclusion

In this paper, we proposed and justified the optimization for Service Routing paths. Use of SPath and hence protocols such as SNG which require service routing are limited by the efficiency and scalability of SPath when applied to ad hoc aggregation of networks. With optimization, not only the scalability of SPath, the performance for service routing will also be improved due to the

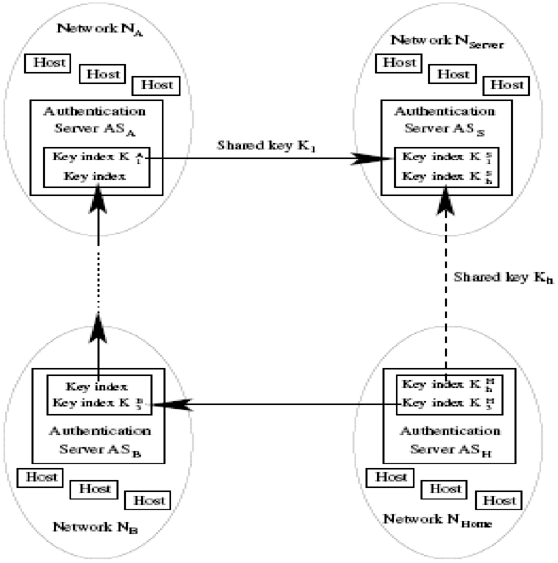


Figure 5. SPath optimization in general

shorter access path and hence less overhead involved. The shorter SAPath of an optimized SPath will make maintenance and network trouble shooting more manageable.

Our work in the future includes an analysis of the correctness of the Service Network Graph which can provide user authentication across heterogeneous networks of different administrative domain without sharing user authentication information.

## References

[1] X.509 (03/00). International Telecommunication Union ITU-T Recommendations X series, 9 2003.

[2] A. Abdul-Rahman and S. Hailes. Using recommendations for managing trust in distributed systems. Proceedings of IEEE Malaysia International Conference on Communication ’97 (MICC’97), Kuala Lumpur, Malaysia, 1997.

[3] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. Hawaii Int. Conference on System Sciences 33 , Maui, Hawaii, January 2000, January 2000.

[4] A. Abdul-Rahman and S. Halles. A distributed trust model. Proceedings of New Security Paradigms Workshops 1997, 1997.

[5] A. R. Au, M. Looi, and P. Ashley. Automated cross-organisational trust establishment on extranets. Proceedings of the workshop on information technology for virtual enterprises, 2001, (7):3–11, January 2001.

[6] T. Beth, M. Borcherding, and B. Klien. Valuation of trust in open networks. Proceedings of the Conference on Computer Security 1994, 1994.

[7] C. Cao, J. Yang, and G. Zhang. Semantic overlay based services routing between mpls domains. Proceedings of 7th International Workshop on Distributed Computing, IWDC 2005, Kharagpur, India, 2005.

[8] D. Denning. A new paradigm for trusted systems. Proceedings of 1992-1993 ACM SIGSAC New Security Paradigms Workshop, 1993.

[9] IETF and IESG. The kerberos network authentication service (v5). Porposed Standard, RFC1510, 9 1993.

[10] D. Lai and Z. Zhang. Integrated key exchange protocol capable of revealing spoofing and resisting dictionary attacks. Technical Track Proceedings, 2nd International Conference, Applied Cryptography and Network Security, Yellow Mountain, June 2004.

[11] D. Lai and Z. Zhang. An infrastructure for service authentication and authorization revocation in a dynamic aggregation of networks. WSEAS Transactions on Communications, 4(8):537–547, August 2005.

[12] D. Lai and Z. Zhang. Network service sharing infrastructure: Service authentication and authorization revocation. Proceedings of the 9thWSEAS International Conference on Communications, July 2005.

[13] D. Lai and Z. Zhang. Secure service sharing over networks for mobile users using service network graphs. Proceedings, Wireless Telecommunication Syposium 2006, April 2006.

[14] D. Lai and Z. Zhang. Self-authentication of encrypted channels in servive network graph. Proceedings, 2008 IFIP International Conference on Network and Parallel Computing, (NPC 2008), October 2008.

[15] D. Lai, Z. Zhang, and C. Shen. Achieving secure service sharing over ip networks. Proceedings, ASEE Mid-Atlantic Section Spring 2006 Conference, April 2006.
[16] D. Lai, Z. Zhang, and H. Wang. Towards an authentication protocol for service outsourcing over ip networks. Proceedings of the 2005 International Conference on Security and Management, (7), June 2005.
[17] B. Lampson, M. Abadi, M. Burrows, and E.Wobber. Authentication in distributed systems: Theory and practice. ACM Transactions on Computer Systems, 10(4):265–310, 1992.
[18] M. Montaner, B. Lopez, and J. L. Rosa. Developing trust in recommender agents. Proceedings of the first international joint conference on Autonomous agents and multi-agent systems, 2002.
[19] M. Reiter and S. Stubblebine. Authentication metric analysis and design. ACM Transactions on Information and System Security, 2(2), January 1999.
[20] S. Robles, J. Borrell, J. Bigham, L. Tokarchuk, and L. Cuthbert. Design of a trust model for a secure multi-agent marketplace. Proceedings of the fifth international conference on Autonomous agents, 2001.
[21] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. Sip: Session initiation protocol. RFC 3261, June 2002.
[22] D. Willis and B. Hoeneisen. Session initiation protocol (sip) extension header field for registering non-adjacent contacts. RFC 3608, October 2003.

## Authors

David Lai received his Bachelor degree in Physics from the Chinese University of Hong Kong. He completed MPhil in Physics and moved on to Education while he worked as a high school teacher for some time. To challenge his own capacity, he managed to change his carrier path from education to IT and worked as a communication engineer, a senior IT consultant and eventually an IT security specialist. He then decided that he would like to have a bit of both education and IT. So he joined the Sciences Faculty of University of Southern Queensland in 2002 and started his IT education and research carrier. He enjoyed both the teaching and research components of his academic life. One of his favorite classes is about switching, wireless and WAN technologies. He has been the examiner for a number of courses ranging from programming; algorithms and data structures; to networking and wireless technology. His research interests include wireless technology, network security, service routing, authentication and password schemes.

Zhongwei Zhang(S’97-M’00) received the B.Sc degree in Applied Mathematics from Harbin Institute of Technology, China, in 1986, and the Ph.D degree in computing From Monash University, Victoria, Australia. He has been a Senior lecturer at the University of Southern Queensland, Australia since 2003. In 2003, he was a visiting professor at the University of North Carolina at Greensboro, USA. His current research include wireless communication networks, wireless sensor network, modeling and optimisation in TCP/IP networks, and E-Commerce technology.