Towards an Authentication Protocol for Service Outsourcing Over IP Networks

David Lai, Zhongwei Zhang, Hua Wang
Department of Mathematics and Computing
University of Southern Queensland
Toowoomba, Queensland, 4350

***Abstract*— *When an IP network is unable to provide users with demanded services, it usually outsources to other networks. Outsourcing will involve the process of authenticating users for services over IP networks. The process can become complex when different networks have different authentication schemes, in which users are required to provide varying sets of information for service authentication. In this paper, we propose a new service authentication protocol by means of which many networks are linked together forming an* ad hoc *network system. This* ad hoc *network system will provide an aggregated range of services to users. Individual networks within the system retain their own authentication schemes and the result of user authentication can be relayed to other networks within the system. We argue that with this protocol, the burden of collecting authentication information and obtaining authentication for each outsourced service is alleviated. We also demonstrate how to construct a service authentication framework and apply the protocol in two case studies.*

***Keywords*— autonomous network, service authentication, outsource service, authentication propagation, authentication token.**

I. Introduction

Within an IP network community, a user needs to register himself with a local network to enjoy all the services available on the network. The range of available services is often one of the determining factors for a user when choosing a network. However, it is difficult for a single network to provide users with all the services they want due to technological and financial limitations. To overcome these constraints the range of services is supplemented by those provided by other autonomous networks. This process of utilizing services provided by other networks is referred to as service outsourcing.

To establish the identity of a user for an outsourced service, either the user authenticates directly to the target network or server, or the authentication status from a local network can be transmitted to the target network or server. Various approaches to authenticate a user have been suggested. For instance, use of ISO X.509 [12] certificates which require a common third party or trust recommendations [6] [7] and trust establishment [1] [2] [8] [3] [9] which requires a common set of trust agents have been proposed. A service authentication protocol in which a local network authenticates a user and automatically relays the authentication status to its linked networks with minimum initial set up overheads is desirable.

When autonomous networks are linked together for service outsourcing, they form a graph of networks. In a graph of networks, each arc represents an outsourcing relationship which could can be a one-way or a dual relationship between the requesting network and the targeted network. We will refer to such a graph of networks as the *service network graph* (SNG).

When a service is outsourced, the target network must be able to authenticate users and to assign the appropriate authorizations to users' requests. As the service outsourcing may involve many networks and different target networks may have different authentication schemes, users in an SNG may be required to have different sets of authentication information. The outsourcing process will not appear to be seamless if a user has to collect and present the authentication information to the target network itself. While collecting and maintaining many sets of authentication information is not a trivial task, authentication and authorization information should be forwarded to target networks by the local network to facilitate seamless service outsourcing. Practical and secure authentication methods are required to extend the range of services which a network can provide by outsourcing.

Service outsourcing is not effective if the list of available services, both local and outsourced, is not up to date. When an autonomous network links to an SNG, it should update its list of services with the new services made available through the SNG. Depending on the outsourcing relationship, networks in the SNG may have to update their lists of services also. The updating of services should be dynamic as autonomous networks may attach or detach from an SNG at anytime.

In this paper, we will focus on how to make the service outsourcing process seamless. We propose a service authentication protocol and the associated frame work for seamless service outsourcing.

In section II we identify the major issues of service authentication and related work. In section III a frame work for service authentication over IP networks is proposed. The protocol for establishing service authentication over IP networks is also explained. In section IV we will then look at how to apply the service authentication protocol to service outsourcing. Section V concludes the paper with a discussion on future work.

II. Problems of Service Authentication and Related Work

Autonomous networks have their own authentication schemes and sets of authentication information for their local users. Users must register as local users before they can access the services offered by a network. Service outsourcing involves more than one autonomous network. Without the assistance from the host network users would have to register with multiple networks and maintain multiple sets of authentication information. Even if the networks have the same authentication scheme and the same set of authentication information, users would have to login to different servers for different services.

Kerberos [10] presented a user friendly solution in which users authenticate with a central authentication server and the authentication status can be propagated to the required servers. With one set of authentication information and one log-in, users will be able to access services available from servers within the same realm.

The problem with this approach is that users have to keep track of the availability of services as they have to specify which server and service they want to access. Users must maintain an updated list of services available by themselves.

The administrative work involved in setting up realms makes Kerberos semi-static. If autonomous networks attach and detach from an SNG in a dynamic fashion, we need a more responsive solution.

When many autonomous networks form an SNG, the network administrators face a problem of authenticating users from other networks which have various authentication schemes and authentication information sets. It is obvious that enforcing a common authentication scheme is not feasible and involves substantial administrative overheads. For instance, when a network links to an SNG and subsequently detaches from the graph, switching to the common authentication scheme and reverting back to the original authentication scheme requires all users of the network to collect and present a different set of authentication information. On the other hand, maintaining a global set of authentication data will fail as networks may link to the graph or detach from the graph at any time. Even worse, some networks may be reluctant to disclose the authentication data for security reasons. As a direct result of this, other networks may not have the authentication data required by the global authentication set. A typical example is the X.500 [11] plan which has never succeeded in producing a global database of named entities.

A solution to this problem is the ISO X.509 [12] recommendation which was published in 1993. Authentication in X.509 is based on the secrecy of the private key and the binding of the public key to a user name. For instance, a Certificate Authority (CA) authenticates a user and binds its name and public key in a digital certificate. If a user demonstrates (s)he is the owner of the private key with a corresponding X.509 certificate, then (s)he is the user named in the certificate. This authentication mechanism is built upon the unanimous trust for the Certificate Authority.

Note that an administrator of an autonomous network may decide to set up a CA for the network or empower a third party to run the CA. However, when many autonomous networks form an SNG, they must agree on a common CA to issue all certificates or on CA certificate chaining. We envisage that the workload increases with the number of users involved.

Another approach is to establish a *trust* [1] [2] [8] [3] [9]. *Trust* is the result of an assessment of an entity relative to a domain of action [5] by an observer. When an observer is authorized by a network administrator to give trust recommendations [6] [7], the observer becomes a trust agent. The trust is represented by a token and each trust token is signed by the trust agent.

It is reasonable for each autonomous network to have its own set of independent trust agents. A user will be asked to provide trust tokens from a few trust agents. By using the aggregated result [4] of the trust tokens, the server can determine the authentication and authorization status of the user for the requested service.

This way of authentication works fine for individual networks. However, for an SNG, each autonomous network will have its own set of trust agents. Either all the networks adopt the same common set of trust agents or users have to collect trust tokens from different sets of trust agents for services outsourced by different networks.

An approach is to design a service authentication framework which allow autonomous networks to link and detach from an SNG with minimum administrative overhead while retaining their own autonomy, independence and integrity. At the same time, forming an SNG should not introduce extra requirements for users.

Our research aims at a service authentication protocol and the corresponding framework to allow service authentication to be performed at the local network and the authentication status to be propagated automatically to other networks in the SNG.

III. Proposed Protocol and Framework

In this section, we propose a Distributed Networks Service Authentication Protocol (DNSA) and a framework for the protocol which allows seamless service outsourcing in *ad hoc* SNG. The concept of SNG is followed by the DNSA protocol.

A. Service Graph

The Distributed Networks Service Authentication framework is based on autonomous networks and SNG. We state our assumptions and define some terms used in the framework in the following paragraphs. An autonomous network is assumed to have a number of local users (U) and the following service providing hosts (S) as shown in Figure 1:

- Authentication Server (AS) which authenticates local users,
- Server (S) which provides services,
- Service Locating Server (SLS) which stores information about local services and outsourced services.

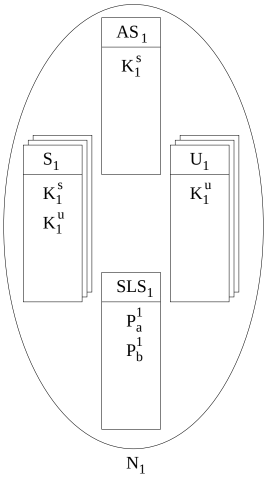


Fig. 1. An autonomous network

Next we assume that an encrypted channel authenticates statements transmitted via the channel [13]. All communications among autonomous networks and between hosts within the same network are assumed to be encrypted using the symmetric encryption and symmetric encryption and decryption keys. For instance, Server Key (K^s) is the encryption and decryption key shared between AS and S while Session Key (K^u) is the encryption and decryption key shared between S and U and generated for each log-in session.

Note that the distributed network from which an outsourced service request is initiated is referred to as the *request network* while the network which provides the actual service is the *target network*.

Let us have a look at the relationship among the n autonomous networks in Distributed Networks Service Authentication.

We say that N_1 attaches to N_2 when:

- N_2 delegates its authentication authority to N_1. In which case AS_2 generates and shares ATK^2 with AS^1.
- N_2 provides services to N_1 as outsourced services of N_1.

Consequently, this is a one-way relationship. As shown in Figure 2, N_1 attaches to N_2 and N_2 is the delegator network while N_1 is a delegatee network. In other words, a delegatee network attaches to a delegator network and this is a target network of the delegatee network.

Moreover, N_1 and N_2 are defined to be mutually linked when N_1 attaches to N_2 and N_2 also attaches to N_1. Mutual linking is therefore a dual-way relationship.

A *Service Network Graph* is a representation of many networks attached to or mutually linked with each other in which an attachment is represented by a single-ended arrow and a mutual-link is represented by a double-end arrow as depicted in Figure 2.

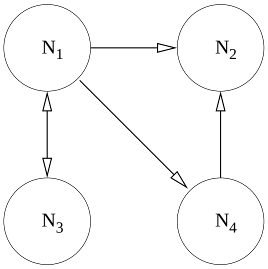


Fig. 2. A Service Graph

For instance, within Figure 2, each node stands for an autonomous network (N_1, N_2, N_3 and N_4) where

- N_1 attaches to N_2 and N_4.
- N_4 attaches to N_2.
- N_1 and N_3 are mutually linked.
- N_2 does not attach or mutually link to any node.

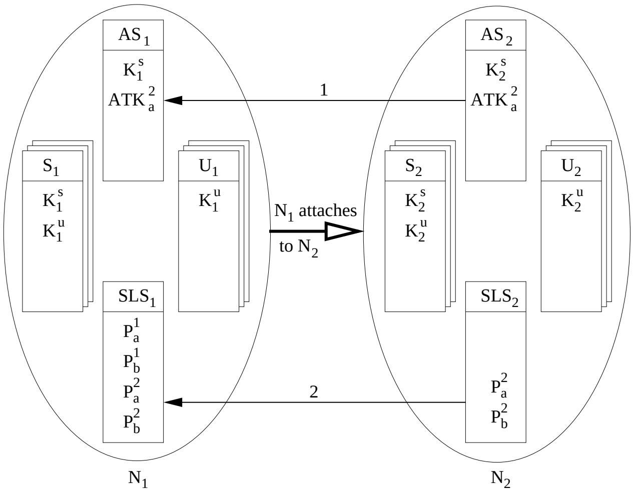
With the Distributed Networks Service Authentication framework in place, we can now proceed to look at the Distributed Networks Service Authentication Protocol.

B. Distributed Networks Service Authentication Protocol

The Distributed Networks Service Authentication Protocol has two distinct operation modes. One is the *Network Participation mode* (NP mode) in which a network links to another network in an SNG. Another is the *User Service mode* (US mode) in which a user accesses a local or outsourced service. We will discuss them in the following sections.

1) Protocol in NP Mode: Let us assume N_1 and N_2 are two separate autonomous networks as shown in Figure 3.

Upon receiving the request from N_1 to attach directly to N_2, AS_2 generates and sends ATK_a^2 to AS_1. SLS_2 also sends information about services available for outsourcing to SLS_1 in the form of a Service Path (P). A Service Path is a service locater similar to a URL and represented by a string of network path and costing metrics. An example is
< [Network Path/]Target Network/Server/Service>
:<Cost Metrics>
where [Network Path/] is optional. Services available for outsourcing include local services in N_2 and those outsourced by N_2. Both AS_1 and SLS_1 will acknowledge the receipt of information to AS_2 and SLS_2 respectively. Note that N_2 can also request to attach to N_1 and form a mutual link with N_1.



Step 1 = AS_2 sends athentication token key to AS $_1$
Step 2 = SLS_2 sends service paths to SLS_1

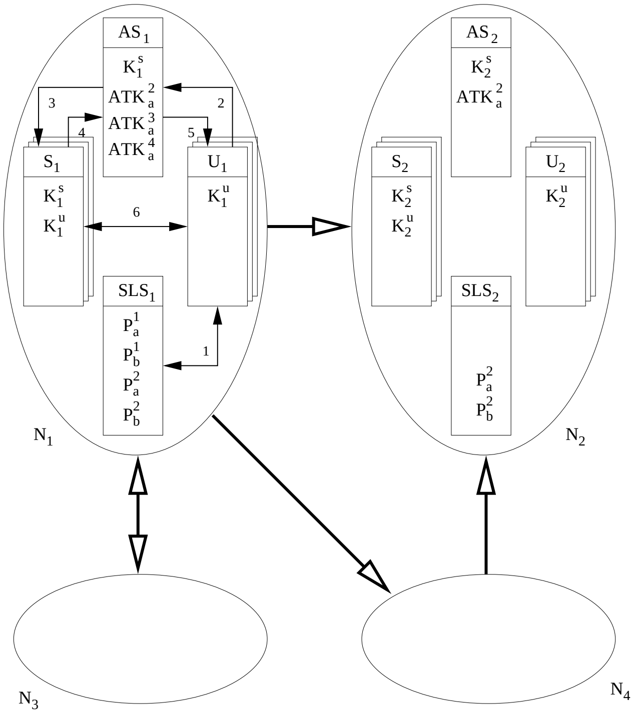
Fig. 3. Attaching one network to another network

We will explain how to handle a service request in section III-B.2.

2) Protocol in US Mode: When a local user U_1 in N_1 requests a service, it will first query SLS_1 whether it is available or not. SLS_1 then returns a message containing a valid service path (P_a^1 or P_a^2 for example) plus its cost metrics or "Service not available" to U_1. If U_1 is comfortable with the cost metrics, the user will authenticate itself to AS_1 and pass along the service path and cost metrics which AS_1 will use to determine the path to reach the target server.

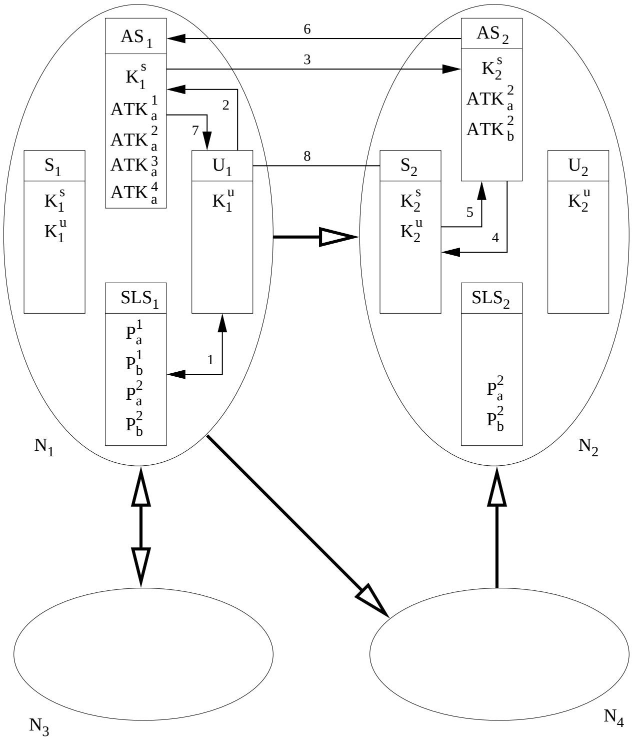
If the authentication is successful, AS_1 will generate a session key K_1^u. If the service is available on a local server S_1 as indicated by the Service Path, AS_1 will encrypt K_1^u using encryption key K_1^s of server S_1 and send it to S_1. S_1 acknowledges the session key K_1^u and returns AS_1 all service information for the request. AS_1 relays the service information and K_1^u to U_1 as shown in Figure 4. If the local server in N_1 does not provide the service, N_1 needs to be outsourced to another network, say N_2 as shown in Figure 5. In this case, AS_1 retrieves the authentication token key of AS_2, ATK_a^2 and uses it to encrypt the session key K_1^u instead of using a server encryption key K_1^s. The encrypted session key K_1^u together with the service path forms an *authentication token*.

On receiving the authentication token from AS_1, the authentication server AS_2 in N_2 extracts the K_1^u from the authentication token. From the service path that was embedded in the authentication token, AS_2 determines if the service is provided locally on N_2 or by networks other than N_1 and N_2. If the service



Step 1 = SLS$_1$ returns P_a^1 to U$_1$ upon request from U $_1$
Step 2 = U$_1$ sends P_a^1 and authentication information to AS $_1$
Step 3 = AS$_1$ generates and sends K_1^u to S$_1$ using K_1^s as the encryption key
Step 4 = S$_1$ sends service information to AS $_1$
Step 5 = AS$_1$ sends K_1^u and service information to U $_1$
Step 6 = Service traffic
Service Path = <./S $_1$/service>:<1>

Fig. 4. User requesting a local service



Step 1 = SLS$_1$ returns P_a^2 to U$_1$ upon request from U $_1$
Step 2 = U$_1$ sends P_a^2 and authentication information to AS $_1$
Step 3 = AS$_1$ generates and sends K_1^u to AS$_2$ using ATK_a^2 as encryption key
Step 4 = AS$_2$ sends K_1^u to S$_2$ using K_2^s as encryption key
Step 5 = S$_2$ returns service information to AS $_2$
Step 6 = AS$_2$ relays service information to AS $_1$
Step 7 = AS$_1$ sends service information and K $_1^u$ to U$_1$
Step 8 = Service traffic
Service Path = <N $_2$/S$_2$/service>:<3>

Fig. 5. User requesting an outsourced service

is a local service provided by S_2, AS_2 encrypts K_1^u with K_2^s and sends it to S_2 as explained before.

In the case when N_1 does not attach directly to N_2 as shown in Figure 6 the service path would indicate that a target network is only reachable via N_4. In this case K_1^u is passed on from AS_1 to AS_2 via authentication server AS_4 in N_4. AS_1 will encrypt K_1^u with ATK_a^4 while AS_4 will encrypt K_1^u with ATK_a^2. Service information returned from S_2 will follow a similar path but in the reverse order.

IV. Applications of the Distributed Networks Service Authentication Protocol

In this section, we illustrate the feasibility of the protocol with two case studies as follows.

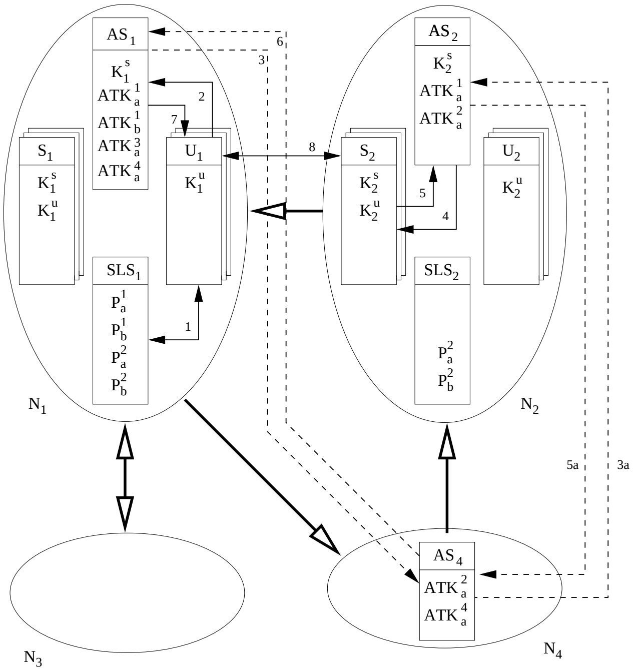
A. Subnets to Subnets

The autonomous network of domain *usq.edu.au* belongs to University of Southern Queensland (USQ). Under this network, a number of subnets are created for individual Faculties. For instance, a subnet of sub-domain *sci.usq.edu.au* was created for the Faculty of Sciences (FoS) in which *probus.sci.usq.edu.au* (Probus) serves as a web server and a file server.

We know the subnet of *sci.usq.edu.au* has no independent authentication server. Authentication is performed by individual servers using a centralized Light Weight Access Protocol authentication information repository (LDATA). Staff members of FoS and student enrolled in programs offered by the Faculty are all legitimate users of the subnet. Every user of the subnet of *sci.usq.edu.au* has an entry in LDATA.

If a user U_{new} from another subnet, say, subnet of *eng.usq.edu.au*, wants to create a web site in Probus, U_{new} has to register as a local user of *sci.usq.edu.au*. Administrative work involves both user management and network related administration. Here we demonstrate how to use the Distributive Network Service Authentication Protocol to do so.

A Distributive Network Service Authentication framework requires a service network must have an AS and a SLS. All servers within the networks use the AS for service authentication. Finally all commu-



Step 1 = SLS_1 returns P_a^2 to U_1 upon request from U_1
Step 2 = U_1 sends P_a^2 and authentication information to AS_1
Step 3 = AS_1 generates and sends K_1^u to AS_4 using ATK_a^4 as encryption key
Step 3a = AS_4 extracts K_1^u and forward K_1^u to AS_2 using ATK_a^2
Step 4 = AS_2 sends K_1^u to S_2 using K_2^s as encryption key
Step 5 = S_2 returns service information to AS_2
Step 5a = AS_2 relays service information to AS_4
Step 6 = AS_4 relays service information to AS_1
Step 7 = AS_1 sends service information and K_1^u to U_1
Step 8 = Service traffic
Service Path = <N_4/N_2/S_2/service>:<5>

Fig. 6. User requesting an outsourced service

nication within the network must be encrypted using symmetric key encryption such as Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA) and Rivest Cipher (RC5). Note that communication with the SLS may be in plain text form.

Due to the fact that the servers within the subnets of *sci.usq.edu.au* and *eng.usq.edu.au* perform service authentication on their own, and SLS does not exist in both networks, we have to create an AS and a SLS for each network

The functions of an AS include:

1) extracts user account name, password and service path from an authentication request from a user;
2) uses account name and password for authentication and retrieves the corresponding authorization;
3) prepends user account and authorization to service path to form authorized service path (AP);
4) forms authentication token by generating and encrypting K^u with ATK of AS in next hop along the Service Path;
5) receives and extracts session keys from authentication tokens and passes them on to local server or AS in next hop as determined by the Service Path;
6) relays service information returned from local servers to AS in other networks;
7) forward service information to local servers or AS in next hop as determined by the Service Path;
8) uses the original LDATA for *sci.usq.edu.au* as authentication information repository.

The SLS has the following functions:

1) being a Light Weight Access Protocol server;
2) creating a service path for all local services within the network as <./service>;
3) converting service paths from delegator network from <./net_path_to_S/S/service> to <./delegator_net/net_path_to_S/S/service>;

Servers in the delegator network need to:

1) open up a service connection with every session key passed to it from AS;
2) use session key for the connection to communicate with the client program;
3) close a connection when the service session for the connection ends.

While on the delegatee network, the client program:

1) requests user input for service demanded;
2) requests Service Path from SLS;
3) requests user input for authentication information;
4) passes service path along with authentication information to AS;
5) upon receiving the session key and service information from AS, connects to the server and start the service session.

Note that each subnet may have an AS and an SLS of its own.

B. *Autonomous Networks*

Similar to the subnet case discussed in section IV-A, we have a case in which one of our research projects requires the use of parallel computers for complex system simulation and needs to access the parallel computing facilities at the Australian National University (ANU). To be able to use the service, we have to submit user information as well as service parameters to the ANU network. This problem can be easily solved if our network can attach to the ANU network and outsource the parallel computing service.

Let us assume the network *sci.usq.edu.au* has already been using the Distributed Network Service Authentication Protocol with its own AS and SLS. We describe here how to use the Distributive Networks Service Authentication Protocol to outsource the service from the ANU network. The procedure is similar to the previous case, i.e. installing an AS and a SLS on the subnet of *sci.usq.edu.au*.

Note that the AS will:

1) receive and extract session keys from authentication tokens and pass them on to servers within the local network;
2) relay service information returned from local servers to the AS in other networks;

The SLS will:

1) create a service path for all local services within the network as <./service>;
2) convert the service paths from the delegator network from <./net_path_to_S/S/service> to <./delegator_net/net_path_to_S/S/service>;

Finally the client program on our subnet of *sci.usq.edu.au* will:

1) request user input for simulation parameters;
2) request Service Path for parallel computing service from SLS;
3) request user input for authentication information;
4) pass service path along with authentication information to AS;
5) connect to the server and start the parallel computing service session upon receiving the session key and service information from AS.

V. Conclusion

A user interested in a service available in a distributed network environment has to establish a trust relationship with a local network first. Each time a service is requested, the network has to authenticate the user before granting the user access to any local or outsourced service. A service authentication protocol which relays authentication status from local network to target network is required for outsourced services. In this paper, we have designed the DNSA protocol which allows autonomous networks to link together, forming an SNG. With this protocol, authentication status is relayed from a request network to a target network without any extra input from the user. Local users of networks adopting the DNSA protocol therefore need only authenticate with its local authentication server. This makes the protocol user-friendly and efficient and alleviates the burden of collecting authentication information and obtaining authentication for each outsourced service. The extra entity required in the DNSA protocol is only a service locating server which is capable of listing services using service path.

Note that we have not included the granting and revocation of service authorization in DNSA. We will focus on DNSA authorization and security issues in the future.

References

[1] Beth, T., Borcherding, M., Klien, B. (1994) *Valuation of Trust in Open Networks.* Proceedings of the Conference on Computer Security 1994.

[2] Reiter, M. (1999). *Authentication Metric Analysis and Design.* ACM Transactions on Information and System Security, Vol. 2, No.2.

[3] Kohlas, R., Maurer, U. (2000). *Confidence Valuation in a Public-Key Infrastructure based on Uncertainty Evidence.* Proceedings of Public Key Cryptography 00, Lecture Notes in Computer Science, Vol. 1751.

[4] Abdul-Rahman, A., Halles, S. (1997). *A Distributed Trust Model.* Proceedings of New Security Paradigms Workshops 1997.

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

[6] Montaner, M., Lopez, B., Rosa, J. L. (2002) *Developing Trust in Recommender Agents.* Proceedings of the first international joint conference on Autonomous agents and multi-agent systems.

[7] Robles, S., Borrell, J., Bigham, J., Tokarchuk, L., Cuthbert, L. (2001) *Design of a Trust Model for a Secure Multi-Agent Marketplace.* Proceedings of the fifth international conference on Autonomous agents.

[8] Abdul-Rahman, A., Hailes, S. (1997). *Using Recommendations for Managing Trust in Distributed Systems.* Proceedings of IEEE Malaysia International Conference on Communication '97 (MICC'97), Kuala Lumpur, Malaysia

[9] Abdul-Rahman A., Hailes S. (2000). *Supporting Trust in Virtual Communities.* Hawaii Int. Conference on System Sciences 33 , Maui, Hawaii, January 2000.

[10] *The Kerberos Network Authentication Service (V5).* Internet Engineering Task Force (IETF) and the Internet Engineering Steering Group (IESG) Porpoised Standard, RFC1510.

[11] *X.500 (02/01).* International Telecommunication Union ITU-T Recommendations X series. http://www.itu.int/rec/recommendation.asp

[12] *X.509 (03/00).* International Telecommunication Union ITU-T Recommendations X series. http://www.itu.int/rec/recommendation.asp

[13] *Authentication in Distributed Systems: Theory and Practice* Lampson B., Abadi M., Burrows M., Wobber E. (1992). ACM Transactions on Computer Systems, vol. 10, no. 4.

Acknowledgment: Special thanks to our colleague Walter Spunde for his constructive suggestions.