

SIP-Based Multimedia Services Provision in Ad Hoc Networks

Y. Rebahi, D. Sisalem, U. Depirianto

Fraunhofer Institut Fokus
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
{rebahi, sisalem, depirianto}@fokus.fraunhofer.de

Contact details of the first author

Yacine Rebahi
Fraunhofer Institut Fokus
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany

Tel: +49 30 3463 7141

Fax: +49 30 3463 8000

Email: rebahi@fokus.fraunhofer.de

Abstract

This paper provides an approach of how SIP-based multimedia services can be ported to ad hoc networks. This technique achieves two main results: first, it overcomes the lack of existing infrastructure, which is the major handicap for service provision in ad hoc networks. Second, it endows the network with a robust security mechanism to protect the privacy of the undertaken communications.

Keywords

SIP, ad hoc networks, multimedia services, security, digital certificates

SIP-Based Multimedia Services Provision In Ad Hoc Networks

Yacine Rebahi, Dorgham Sisalem, Unggul Depirianto
Fraunhofer Fokus, Kaiserin Augusta Allee 31, 10589, Germany
{rebahi, sisalem, depirianto}@fokus.fraunhofer.de

Abstract—This paper provides an approach of how SIP-based multimedia services can be ported to ad hoc networks. This technique achieves two main results: first, it overcomes the lack of existing infrastructure, which is the major handicap for service provision in ad hoc networks. Second, it endows the network with a robust security mechanism to protect the privacy of the undertaken communications.

I. INTRODUCTION

As mentioned earlier, Session Initiation Protocol (SIP) [1] is a protocol that is used for managing multimedia communication between users. It works in concert with other protocols to deliver multimedia data such as text, audio or video. SIP was designed to work in a fixed network where a connection to a SIP proxy server is available. On the other hand, an ad hoc network is a networking system that allows nodes to form a temporary network without any help from preexisting infrastructures. Each node acts as a router to forward data between users. Such a network can be built easily and automatically [17]. Unfortunately, as there might be no support from a fixed network, a connection to a server cannot be guaranteed.

The goal of this work is to find a mechanism to implement SIP in ad hoc networks. Within this context, it is important to compensate the lack of support from the fixed network. Furthermore, a security mechanism and a protocol for realizing the mechanism to be deployed in the ad hoc networks environment will be studied. The security mechanism must be distributed, in the sense that no special node is needed to be contacted.

The rest of this paper is organized as follows: section II gives a brief overview of SIP. Section III describes how security is achieved in ad hoc network. Section IV presents an architecture of SIP in an ad hoc network, section V presents the security mechanism and

its communication protocol and section VIII concludes the paper.

II. THE SESSION INITIATION PROTOCOL

The Session Initiation Protocol (SIP) [1] is an application-layer protocol for managing multimedia sessions in the Internet. A session can be established between two end-users or more and can involve IP phone calls, conferencing and messaging. SIP is handled through messages such as INVITE for initiating a session and BYE to terminate it. The main components in a SIP network are briefly described below,

SIP User Agents (SUAs): these are the end devices in a SIP network. They can originate SIP requests to establish a media session and send and receive media. A user agent can be either a SIP phone or a SIP client software running on a PC.

Servers: these are intermediate SIP entities in a SIP network that assist the user agents in establishing media sessions and some other functions. SIP servers are three categories: proxies, redirect servers and registrars.

Location servers: a location server is a database where users information such URLs, IP addresses, scripts and other preferences are stored. A location server may also contain routing information such as locations of proxies, gateways and other location servers.

III. SECURITY IN AD HOC NETWORKS

In general, security in ad hoc networks uses cryptography mechanisms based on symmetric and asymmetric key encryption. In the former, a shared secret is distributed to all the nodes in the network. However, in the latter, each node is assumed to have a pair of keys: a private one which is kept secret and used to decrypt the received messages, and a public one which is distributed to the other nodes in the network

and which is utilized to encrypt the messages towards the processor of the key material.

Though asymmetric key cryptography is computationally expensive, it provides a more robust security mechanism. Within this category, several solutions are suggested in the literature, one can mention Partially Distributed Certification Authority [3], Fully Distributed Certification Authority [4] and Self-Organized PKI (also known as Web-of-Trust for Ad Hoc Networks) [5].

In the sequel, we will provide an overview of the Fully Distributed Certification Authority protocol, which we choose to achieve security in our architecture since it reflects better the ad hoc networks characteristics as it will be clarified.

A. The Fully Distributed Certification Authority

The Fully Distribution Certification Authority (FDCA) [4] is a protocol built on the public key infrastructure mechanism and uses a threshold scheme [6]. It assumes the existence of an entity, called Certification Authority (CA), which can issue certificates and maintain the certificates database. In general, a digital certificate is a statement containing information such as, user ID, user’s public key and the certificate’s validity time. In the FDCA protocol, the certification authority is distributed between all the nodes. The CA has a secret and public key. The latter is known to every network member. In this architecture, each node carries a certificate signed by the CA’s private key. The latter is shared between nodes in the network, each node holds a partial private key. Any group of nodes cannot reconstruct the secret key even with collaboration, and any coalition of nodes can collaboratively sign a certificate. The FDCA protocol assumes that each node has a certificate signed by the private key of the CA before joining the network. A valid certificate is used by a node as a means for authenticating itself and joining the network. On the other hand, when a node’s certificate expires, the node requests a new one from any coalition of nodes. To enhance the network security, FDCA also proposes on the one hand, a certificate revocation procedure, and on the other hand, an update of the private key shares algorithm to recalculate new shares for each node after some time. This will challenge adversaries to compromise nodes in only short periods. Although, this mechanism suffers from some drawbacks, which are

out of the scope of this paper, this technique is suitable for ad hoc networks in the sense that it does not depend in any case on one or some special nodes to provide service.

IV. SYSTEM ARCHITECTURE

In some scenarios, ad hoc networks may not have any support from fixed networks. Entities such as DNS server, DHCP server, or CA may not be available in ad hoc networks. Nodes intend to communicate in an ad hoc network must rely solely on their own capabilities to build the network. In this paper, we propose an architecture to deploy SIP in ad hoc networks. In our architecture, we assume that each user has a unique and non-zero ID, and there is a mechanism to accomplish the address distribution tasks. Furthermore, we assume that the lower layers are secure and/or have their own security protection mechanisms. More precisely, in network layer, we assume that routing information are valid. Thus, despite message forwarding mechanism in ad hoc network, a message arrives at node X when we do send the message to node X and no other node receives the message. The proposed architecture for implementing SIP in ad hoc networks is depicted in the figure below.

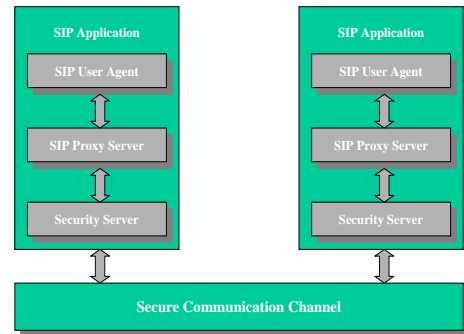


Fig. 1. System Architecture

Figure 1 shows two SIP applications reside in two different nodes, connected by a secure communication channel. Each SIP application consists of a SIP user agent (SUA), SIP proxy server (SPS) and Security Server (SS). The SUA is used by the user as an interface to

communicate. The SPS is used to route request or reply SIP messages to or from a local SUA. An SUA also has a registrar and location service database part as its components. The SS is used to secure the communication channel between users. As explained earlier, in ad hoc networks there is no guarantee that a node can be contacted by other nodes at any time. Furthermore, the nodes move arbitrarily and the membership of the networks change over time. Thus, we cannot rely on one or several nodes to be the special entities to facilitate the communication, e.g. acts as a SIP proxy server, because those nodes can not always be contacted. In our architecture, each node has its own SIP proxy server (SPS). In this way, we remove the need of a connection to a central SIP proxy server. Moreover, the SPS has its own user location database, so any incoming or outgoing messages can be routed to the intended recipients. We use certificates in the security mechanism. The certificate is also used for distributing user address. Thus, to have a distributed system, we need a Security Server in each node to act as a distributed Certification Authority (CA) and to secure messages from the user.

A. Message Processing

When an SUA wants to send a message, it always sends the message to its own SPS, i.e. the SPS residing on the same device. The way an SUA creates and sends a message follows the procedures described in [1]. An SUA has to send a REGISTER message to its own SPS before it can start communicating with other users. An SPS receives and forwards messages from SUA. If the message is a REGISTER message, the registrar processes the message. The registrar can authenticate the SUA using 401 (Unauthorized) response codes. An outgoing request message, i.e. a request message comes from the local SUA, must be forwarded to SS. Note that the term outgoing request message includes REGISTER messages coming from the local SUA. An incoming request message coming from a local SS should be forwarded to the target of the message. An SPS should not receive any incoming message other than from the local SS. An SPS can consult a local location service database when forwarding an incoming message. In most cases, however, the target of the message will be the local SUA. An SPS may receive an incoming REGISTER message from other users, i.e. not from the local user. In this case, the registrar may process the incoming REGISTER message, however, the incoming REGISTER message should not be forwarded back to the local SS. For a reply message, an SPS has to forward the message to

the entity contained in the second Via header of the message. This is a usual SIP proxy behavior, as described in [1], when processing a reply message. An SS receives and forwards two types of messages, SIP messages and security messages. For a SIP message, an SS first classifies the message into outgoing or incoming message. An outgoing message is a message coming from the local SPS and an incoming message is a message comes from another SS residing in another node. For an outgoing SIP message, an SS checks whether or not the message is a REGISTER message. If it is an outgoing REGISTER message, the SS broadcasts the message into its neighboring nodes. This allows other nodes to learn some information regarding this particular node. Before broadcasting a REGISTER message, an SS attaches a certificate of the user in the body of the message. The certificate, which is attached in the body of an outgoing SIP message, must be signed by SK . Furthermore, if the message is not an outgoing REGISTER message, i.e. other outgoing SIP request message or an outgoing SIP reply message, the SS forwards the message to another SS which is responsible for the target of the message, after signing and encrypting the message, . The SS first decrypts and then verifies the signature in an incoming SIP message. If it is verified, the SS checks whether or not the message is a request message. If it is an incoming SIP request message but not an incoming REGISTER message, the SS inserts a Via tag in the header of the SIP request message and forwards the message to the local SPS. This will allow the SPS to forward the reply of this request message back to the SS later. Furthermore, if it is an incoming SIP reply message or an incoming REGISTER message, the SS just forwards the message to the local SPS.

V. SECURITY MECHANISM

We propose a security mechanism which is suitable for ad hoc networks. Our mechanism is based on the work of [4], in the sense that each node has a share of the system key SK and any coalition of k nodes can collaboratively sign a certificate. Our mechanism is suitable for emergency situations or rescue missions uses. In such a scenario, most of the network members know each other before and security is important only to close the network from outside interference. Furthermore, we use a certificate to bind a user's public key, PK , with a user name and an ID. This allows us to distribute addresses by using certificates. In our security mechanism, each user u has a unique non-zero ID, i , denoted as u_i . The ID is the same as the network

address, i.e. IP address. When a user joins a network, he/she can get a unique address using any possible mechanism. We assume that the chosen mechanism guarantees a unique address despite the possibility of two ad hoc networks merging. There are several proposed mechanisms for autonomously configuring network address, such as in [7] and [8]. Furthermore, we assume that it is possible to manually configure the addresses of some nodes. This is required in order to have nodes that have certificates before joining the network. These nodes are needed to start the network. They are denoted as initial nodes. Each of these initial nodes has a share of the system key as well. They get the keys from a dealer, who is also responsible for creating their certificates. After distributing the keys and certificates, the dealer goes offline and erases the system key information, thus no node knows about the complete system key. When a node u joins a network, u can be trusted whether or not it has a certificate signed by SK . If u has a certificate signed by SK , u can use it directly to communicate with others. However, u must send its certificate and address to the other nodes. This can be done by using a network wide broadcast or any other mechanism that is suitable. This is important because a certificate, which is signed by SK , contains the address information of u . Thus, by advertising the certificate, other nodes will be aware of u 's address and its existence. If u does not have a certificate signed by SK , u can create a certificate and sign it using its own key. u then broadcasts a request, asking for a new certificate signed by SK . u must attach the certificate that it has created in the body of the request. Any node, v , which receives the request, tries to verify the certificate in the request body. If v can verify the certificate in the request body, meaning that v knows u 's public key, and v has a partial system key, then v must create a partial certificate for u using its partial system key.

Let's provide now a brief description of the used key system,

Key generation and splitting: In this mechanism, each node u generates a key pair $\langle sk_i, pk_i \rangle$ based on the RSA algorithm [14]. The private key sk_i is known only by node u_i , while the public key pk_i is advertised to other nodes. A system key is generated in the same way as a normal RSA key pair. However, it is generated by a dealer. To create partial system keys, a dealer then uses the threshold secret sharing described in [6].

Certificate issuing: Here we adopt the dynamic coalescing method which simply means that a node u_i requests a certificate from any coalition of k nodes in his neighborhood. Once it receives k certificate shares from k nodes, he can combine them into a certificate that he can use, for more details we refer to [4].

Partial share: After getting its certificate, a node u_i should ask for a partial share of the system key, P_i . It is important that every node have its own share of the partial system key. This will increase the probability of having at least k nodes agree to serve a certificate request/renewal.

VI. IMPLEMENTATION

We have implemented the security mechanism presented previously. The implementation is done under Linux operating system and is written in the C language. In the transport layer, we use the UDP as we need to broadcast some of the messages. Furthermore, we have also designed a communication protocol for the security mechanism. The protocol is used for requesting a certificate, replying a certificate request, requesting a partial key, replying a partial key request, sending an ID list, sending an adder list and advertising a certificate. This protocol is implemented as part of the Security Server (SS). We use the SIP Express Router (SER) [9] for the SIP Proxy Server (SPS) part. SER is extremely configurable, allowing the creation of various routing and admission policies, as well as setting up new and customized services. This feature allows us to easily create a routing script to accomplish the tasks required for our architecture.

VII. FURTHER DEVELOPMENT

We admit that our implementation still requires some improvements. However, we have laid some basic foundation to build such a system. A SIP architecture in ad hoc network has been presented and a security server has been shown. We have identified some parts of the implementation that needed improvements. Here we discuss some of the problems and possible solutions of them.

A. Security Mechanism

The security mechanism that we discussed earlier has proved to be implementable and working. Still, some of the requirements in that mechanism limit the flexibility of the general system. The use of nodes addresses to

calculate certificates requires that each node has a unique address. Furthermore, it is required that each network participant knows the system public key prior to joining the network. While this solution works well when the establishment of the network is governed by an authority, i.e. a group of rescue team starts a network, it is not flexible enough for establishing an arbitrary network. A solution based on Web-of-Trust [5] combined with the solution described in this work may be used to start an arbitrary network. Thus, any user, without any group membership, can join the network. Certificates exchange mechanism can be used to build a trust relationship between a new user and the rest of the network. When the user can be trusted, it can get its own certificate signed by a system key. This certificate is used to communicate with other users. Furthermore, the new user can get a system public key from its neighbor. However, this still leaves the address uniqueness requirement unsolved. Our mechanism requires that a node asks for a new certificate when it changes its address and also changes its own partial system key. These requires a lot of network communication when two ad hoc network merge. A possible solution for this problem is to have a network identity for each ad hoc network and include the network identity in the certificate and calculation of a partial system key, so when two ad hoc networks merge, there is no need to change address and furthermore no need to change certificate and partial system key.

B. Address Resolution

Peer-to-peer networks have their own mechanisms for locating resource. Such as in gnutella [10] network, resource location is used to find data in other computers. This mechanism can be used to find a user in an ad hoc network. Furthermore, protocols such as Scribe [11], Nom [12] and Pastry [13] can also be used to locate a user in the ad hoc network. The user address, A , can be converted into a hash and saved in a node, B , which is chosen using an algorithm. When another user, C , needs to find the network address of A , C can calculate the hash of A 's address. Then using the same algorithm, C finds node B to ask for A 's network address. This address resolution mechanism should be incorporated in the system architecture.

VIII. CONCLUSION

We have presented an architecture for implementing SIP in an ad hoc network. The architecture is flexible and can be used as a basic model for implementing SIP in an ad hoc network. Furthermore, we have also

presented a security mechanism which can be used in our architecture. The lack of a connection to a SIP server in an ad hoc network, can be counter balanced by adding a sip proxy layer in the user application. Furthermore, to provide a secure communication, a security server can be added as well. Our technique is suitable for a group communication using an ad hoc network, such as in emergency team communication. Our system might be improved, this is the reason of identifying some problems with the current implementation and their possible solutions for future developments.

REFERENCES

- [1] J. Rosenberg et al, "SIP: Session Initiation Protocol", RFC 3261, June 2002
- [2] <http://www.ietf.org/html.charters/manet-charter.html>
- [3] L. Zhou and Z. Haas, "Securing Ad Hoc Networks", IEEE Network Magazine, 13(6), Nov/Dec 1999
- [4] H. Luo and S. Lu, "Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks", Technical Report 200030, UCLA Computer Science Department, October 2000
- [5] S. Capkun et al, "Self-Organized Public -Key Management for Mobile Ad Hoc Networks", Technical Report EPFL/IC/200234, June 2002
- [6] A. Shamir, "How to share a secret", Communications of ACM, 1979
- [7] J. Boleng, "Efficient Network Layer Addressing for mobile Ad Hoc Networks", in ICWN'02, June 2002.
- [8] C.E. Perkins, "Mobile-IP, Ad Hoc Networking and Nomadicity", in Proc. of COMPSAC'96
- [9] "The SIP Express Router (SER)", <http://www.iptel.org>
- [10] "The Gnutella Protocol Specification v0.4", <http://www.clip2.com>
- [11] P. Druschel et al, "Scribe: The Design of a Large-scale Event Notification Infrastructure", Proc. NGC'01, November 2001.
- [12] D. doval and D. O'Mahony, "Nom: Resource Location and Discovery for Ad Hoc Mobile Networks"
- [13] P. Druschel and A. Rowstron, "Pastry: Scalable, Decentralized Object Location and Routing for Large-scale Peer-to-peer Systems", Proc. Middleware '01, 2001.
- [14] Y. Frankel et al, "Proactive RSA", CRYPTO, 1997
- [15] P. Hoffman, "Enhanced Security Services for S/MIME", RFC 2634, June 1999
- [16] J. Franks, et al, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999
- [17] <http://www.ietf.org/html.charters/manet-charter.html>
- [18] J. Mackar and S. Corson, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, January 1999
- [19] C. Ellison et al, "SPKI Certificate Theory", RFC 2693, September 1999
- [20] A. O. Freier et al, "The SSL Protocol Version 3", <http://home.netscape.com/eng/ssl3/ssl-toc.html>
- [21] P. Zimmermann, "The Official PGP User's Guide", MIT Press, 1995
- [22] L. Venkatraman and D. P. Agrawal, "A Novel Authentication Scheme for Ad Hoc Networks", University of Cincinnati