SECURITY ARCHITECTURE FOR SDR SYSTEM USING OTA DOWNLOAD SEQUENCE

Ankita Taneja[1], Ved. P. Mishra[1], Ajay Kr. Singh[1], G. Singh[2] and S. P. Ghrera[1]
[1] Department of Computer Science and Engineering
Jaypee University of Information Technology, Solan -173 215, India
[2]Department of Electronics and Communication Engineering
Jaypee University of Information Technology, Solan -173 215, India
E-mail: ajay41274@yahoo.com and drghanshyam.singh@yahoo.com

ABSTRACT

Over The Air (OTA) software downloads can provide seamless interoperability without relying on any additional equipment or spectral assets. By using OTA software downloads, arriving first responders can download the interoperability parameters and immediately gain access to the host radio system. Software Defined Radio (SDR) is also expected to solve the compatibility problem among various mobile communication standards so that people can use the same device for different mobile environment. If these mobile communication environment are constructed, integrity and confidentiality of data in between terminal and service provider or each network server authentication become very important. In this paper, we propose a method for authentication and transmission security for OTA software download sequence.

1. INTRODUCTION

Due to recent advancement in high speed digital signal processors and analog to digital converters, the commercial implementation of software radio has become viable. The main advantage of software is its flexibility such that in can be programmed for emerging standards. It also can be dynamically updated with new software without any change in hardware and infrastructure. Rapid deployment is another important feature of the software radio. In wireless applications where different standards might be deployed, user's roaming can be a big issue in existing platform. OTA software downloads provide numerous benefits for the public safety community. Most of these benefits are inherent in SDR technology [1], but are enhanced by OTA software downloads [2]. Two of the key benefits are increased public safety interoperability and a reduction in radio lifecycle costs. OTA software downloads have great potential to enhance interoperability and reduce lifecycle costs in the public safety community [3]. As coordination increases among public safety agencies, the need for interoperable communications grows. Additionally, maintenance and upgrade costs could be reduced through the use of OTA software downloads. By implementing these software changes over the air, radio technicians would no longer be required to physically touch each radio, thereby reducing the implementation cost. Additionally, the maintenance and upgrades could be completed faster because all of the radios could receive the transmission at once [4] in the communication systems. The security and authenticity becomes a major issue.

In this paper, we have proposed the latest cryptographic technique namely Kerberos version 5 developed in MIT [5] for secure transmission, authentication and access control of all forms of communications taking place using over-the-air transmission [6]. Figure 1 explains the OTA download sequence and the proposed position for Kerberos and when the user requests the software download, the SDR terminal attempts connecting to an OTA download server by using Kerberos, and the server authenticates the user by Kerberos authentication. If user authentication succeeds, the encrypted data is transmitted and received on the session established. The SDR terminal transmits the terminal information which includes CPU, specification, digital signal processor specification, operating system information, and vendor information.

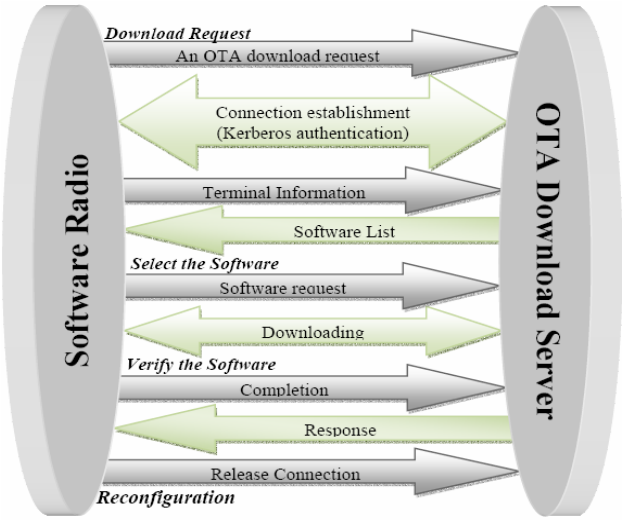

Fig. 1. OTA download sequence.

The download server makes a list of appropriate software modules based on the terminal information, and transmits it to the SDR terminal. The user selects the desired module and requests it to the download server. Before or during the OTA software download SDR terminal will check for any virus or bug. If any virus or bug is found it will acknowledge and stop downloading. After the SDR terminal downloads, it verifies the software module. When verification succeeds, the SDR terminal notifies the download server of download completion. Finally, the SDR terminal releases the connection that was established. Authentication and access control need to satisfy various use field of communication terminal. Authentication is required to allow service use that offer within domain, and access control need to restrict user's service access extent. Encryption is required for data integrity and confidentiality. Access control prevents unlawfulness access of software, and offer integrity, confidentiality and availability of software service. Access competence is established in terminal's state. Access control manage technology is divided by identity-based policy, rule-based policy and role-based policy. Access control of identity-based policy consists on subject or sub group's position, rule-based policy consists based on rule that is established on permission grade for object. Role-based policy is access control technology that use together with identity-based and rule-based policy [7].

2. ILLUSTRATIVE APPROACH

2.1 SSL-Based Security

Secure Sockets Layer (SSL) [8] is a protocol for information security developed for internet applications and is currently widely used in the internet. A recent SDR [9, 10] prototype uses over-the-air download over TCP/IP but there is some problems such as:

1. It does not take into account certain differences between SDR secure download requests and conventional internet secure download.
2. SSL has recently been shown to have weakness if inappropriate cryptographic components are used [9, 10]. Also, there are certain security problems related to widely used encryption algorithms, RC4 [11].
3. Only OTA system is considered, therefore, the download file cannot be stored on media and distributed in other forms.
4. The system requires the use of TCP/IP in the terminal devices.

2.2 Kerberos 5

Normally, when most users make a connection over a network to a remote host they use protocols like telnet or rlogin and then authenticate themselves either with a password or .rhosts file. Both of these methods are vulnerable to attack from a malicious party out on the network. By typing their password in a normal remote login session with any of the standard network utilities, the characters of the password are all sent over the network in the clear. It is easy for a third party who has access to the network to intercept these characters and get the user's password and record it for later use. By using a .rhosts file a user's password doesn't go over the network, but the .rhosts mechanism relies on the securing of the internet protocol (IP) layer. This is the layer of the network that gets packets from one host to another and unfortunately the IP layer has very little in the way of security and this makes the user vulnerable to an attacker using an attack called "IP spoofing". Kerberos 5 solves these problems by providing a means of authenticating the user where the user's password never goes over the network. Kerberos bases it's authentication on electronic "tickets" that are send encrypted over the network. A central server, called a key distribution center (KDC), authenticates the user and gives them tickets that allow them to access hosts and services (like popper) as shown in Figure 2. Kerberos calls every user, host or service that uses Kerberos a principal and each principal has a key.

For users, the key is created from their password and for hosts and services it is a long random number. These keys, like passwords, are kept secret and are known only to the principal the key belongs to and the KDC. By using a form of encryption called "symmetric encryption", these keys can be used to do authentication and send secure messages. Symmetric encryption means that if a message is encrypted with a key, it can only be decrypted with the same key. Since keys are only known to two parties, the principal they belong to and the KDC, this allows secure messages to be sent between principals and the KDC.

3. PROPOSED MECHANISM

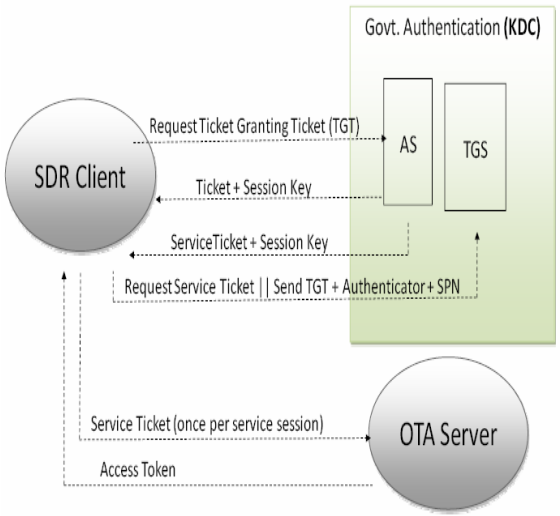

Fig. 2. Proposed Kerberos mechanism for authentication and access control.

The standard Kerberos version 5 authentication protocol communication sequences consist of six (five required and one optional) messages. These messages compose three types of exchanges (also known as sub-protocols), which are examined more closely later in this section.

3.1 Authentication Service Exchange

The SDR client contacts the key distribution center's authentication service (AS) for a short-lived ticket (a message containing the client's identity) called a ticket-granting ticket (TGT). This happens at logon. The authentication service (AS) constructs the TGT and creates a session key the client can use to encrypt communication with the ticket-granting service (TGS). The TGT has a limited lifetime. At the point that the client has received the TGT, the client has not been granted access to any resources, even to resources on the local computer. The reason for using a TGT instead of AS simply issuing a ticket for the target server is that if the AS issued tickets directly, the user would have to enter a password for every new server/service connection. Issuing a TGT with a short lifespan (typically 10 hours) gives the user a valid ticket for the ticket-granting service, which in turn issues target-server tickets. The TGT's main benefit is that the user only has to enter a password once, at logon.

3.2 Ticket-Granting Service Exchange

The SDR client wants access to local and network resources. As it is using network we can use software to remove any virus or bugs. To gain access, the client sends a request to the TGS for a ticket for some network server or service. This ticket is referred to as the service ticket or service ticket. To get the ticket, the client presents the TGT, an authenticator, and the name of the target Software Radio Server (the Server Principal Name or SPN). The TGS examines the TGT and the authenticator. If these are acceptable, the TGS creates a service ticket. The SDR client's identity is taken from the TGT and copied to the service ticket. Then the ticket is sent to the SDR client.

The point to be noted here is that the TGS cannot determine if the user will be able to get access to the target server. It simply returns a valid ticket. Authentication does not imply authorization.

3.3 Client/Server Exchange

After the SDR client has the service ticket, the client sends the ticket and a new authenticator to the target server, requesting access. The SDR server will decrypt the ticket, validate the authenticator, and for communication services, create an access token for the user based on the Security identifier (SID)'s in the ticket. Optionally, the client might request that the target server verify its own identity. This is called mutual authentication. If mutual authentication is requested, the target SDR server will take the client computer's timestamp from the authenticator, encrypt it with the session key the TGS provided for client-target server messages, and send it to the client.

3.4 Shortcomings of Kerberos

Once a Kerberos ticket leaves the KDC it is prone to be hacked and used by a third party like a password. Although Kerberos has mechanisms in place to help prevent this from happening, if a machine where a credentials cache is stores is compromised, it is possible tickets can be stolen. To help limit damage from such an event, all tickets have a lifetime encoded into them so that they expire after a set period of time. How long depends on the local site configuration, but is typically on the order of a day. Unfortunately from the user perspective this is a pain. This means they must re-authenticate every so often and requires special support for long-running processes.

Although Kerberos offers some support for preventing user's from choosing bad passwords with dictionary checking, character class counts, and password history, the user still must be responsible for choosing a password that won't be easily guessed by a hacker.

4. CONCLUSION

Kerberos is a widely-deployed network authentication protocol that is being considered for standardization. Many works have analyzed its security, identifying flaws and often suggesting fixes, thus helping the protocol's evolution. In this paper, we have proposed to use a mechanism to control the authentication and access control for OTA transmission of data and signals between the SDR server and client. Mechanism that proposes in this paper can protect terminal from virus or bugs through safe software download, can protect away attack about data forgery and alteration from attacker and can protect terminal and servers from data interception and main-in-middle attack. Proposed mechanism is shortcoming that system resources should be got to apply in present mobile communication.

5. ACKNOWLEDGMENT

We are grateful to our university for giving us full support in all fields.

6. REFERENCES

[1] J. Mitola, "The Software Radio Architecture", *IEEE Comm. Magazine*., Vol. 33, No. 5, pp. 26–38, May 1995.

[2] L. B. Michael, M. J. Mihaljevic, and S. Haruyama, "A Framework for Secure Download for Software Refined Radio", *IEEE Comm. Magazine*, Vol. 40, No. 7, pp 88-96, July 2002.

[3] M. Cummings and S. Heath, "Mode Switching and Software Download for Software Defined Radio: The SDR Forum Approach", *IEEE Comm. Magazine*, Vol. 37, pp. 104–106, Aug. 1999.

[4] Software Defined Radio (SDR) Forum: *http://www.sdrforum.org*.

[5] Massachusetts Institute of Technology, "Kerberos: The Network Authentication Protocol", *http://web.mit.edu/Kerberos.*

[6] H. Shiba, T. Shono , Y. Shirato, I. Toyoda, K. Uehara, and M. Umehira, "Software Defined Radio prototype for PHS and IEEE 802.11 wireless LAN", *IEICE Ttrans. Comm.*, Vol. E85–B, No.12, pp. 2694-2702, Dec. 2002.

[7] E. Gallery, "A Policy-Based Framework for The Authorization of Software Downloads in a Mobile Environment", *In 2nd Software Defined Radio Technical Conf. (SDR03), Orlando, Florida, USA*, pp. 17-19, Nov. 2003.

[8] A. O. Freier, P. Karlton and P.C. Kocher, "The SSL Protocol – Version 3.0", *http//home.netscape.com/eng/ssl3.*

[9] H. Krawczyck," The Order of Encryption and Authentication for Protecting Communications (or: How Secure is SSL)," *Advances in Cryptology- CRYPTO, LNCS*, Vol. 2139, pp. 310-31, 2001.

[10] H. Shiba et al.,"Software Defined Radio Prototype for PHS AND Wireless LAN System (l)-System Design and over-the-air-download",*Tech. rep. IEICE, SR01-12*, pp. 33-38, sept. 2001.

[11] S. Fluhrer, I. Mantin, and A.Shamir, "Weaknesses in the Key Scheduling algorithm of RC4", *Selected Areas in Cryptography-SAC200, LNCS*, Vol. 2259, pp. 1-24, 2001.