Army Research Laboratory

Secure Link Middleware

by Brian B. Luu

ARL-TR-4535 August 2008

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer’s or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-4535 **August 2008**

Secure Link Middleware

Brian B. Luu
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
August 2008

2. REPORT TYPE
Final

3. DATES COVERED (From - To)
FY 2007

4. TITLE AND SUBTITLE
Brian B. Luu

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Secure Link Middleware

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRD-ARL-CI-NT
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-4535

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. National Archives and Records Administration
8601 Adelphi Rd
College Park MD 20704-6001

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report documents the development of the ARL Secure Link middleware which was developed for the U.S. National Archives and Records Administration (NARA) under the research task entitled “Secure Communication Network Middleware”. The task was performed by the U.S. Army Research Laboratory (ARL) in FY07 to support NARA in its effort to build a secure distributed computing environment for processing sensitive electronic records archives (ERA).

15. SUBJECT TERMS
Middleware, security, encryption, authentication, routing, firewall

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Binh Nguyen
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	SAR	14	**19b. TELEPHONE NUMBER** (*Include area code*) 301-394-1781
U	U	U			

Contents

List of Figures **iv**

1. Background **1**

2. ARL Secure Link Middleware **2**

2.1 Design Architecture....2

2.2 Implementation Requirements 3

2.3 Operation Outline....4

3. Benefits **5**

4. Conclusion **6**

5. References **7**

Distribution List **8**

List of Figures

Figure 1. Functional and operational diagram of Secure Link in two networked computers.........3

Figure 2. A demonstration setup of using the ARL Secure Link middleware in an inter-network environment. 5

1. Background

One of the challenges for the U.S. National Archives and Records Administration (NARA) is to provide essential information assurance (IA) services for sensitive electronic records archives (ERA) in transit between networked computer systems. Current software technologies for securing data in transit rely on cryptographic algorithms and protocols provided in IP Security (IPSec), Virtual Private Network (VPN), or Secure Shell (*ssh*).

The general difficulties of using IPSec and VPN are the complexity and compatibility. IPSec has been evolved and updated with new standards since 1995 (with RFC 1825-1829) to 2005 (with RFC 4301-4309). VPN are generally designed and built based on proprietary algorithms. Usually, they should be acquired, installed, and operated from the same manufacturer. Therefore, typically, IPSec and VPN are implemented and operated at network routers by network administrator to provide security for network traffic between local area networks (LAN) rather than being used by end users at system level. For example, IPSec or VPN are used to connect internal LANs of different sites of an organization through a public network such as the Internet. But with this type of operation, there are no end-to-end encryptions between any two networked computers in the same LAN or in different LANs. Hence, communication traffic of two computers in a same LAN or communication traffic from a local node to its router has no protection.

The Secure Shell technologies and its derivatives such as Secure Copy (*scp*), Secure Shell File System (*sshfs*) are designed to operate at the application level and to provide network security for specific applications. For example, *ssh* is for securely logging in or accessing remote computers; *scp* is for securely copying files from remote computers; *sshfs* is for securely accessing remote file systems. The network security offered by these software applications does provide end-to-end encryptions for computers, but they are designed specifically for each particular application. They are not designed to provide network security for general purpose network applications. However, there are some successes in adjusting and tuning some application software and *ssh* to make application software to operate securely through *ssh* technologies. But it is cumbersome and difficult to tune application software to obtain the desired security, and sometimes the tuning limits the capability of the application software. Moreover, it is required network security skills (such as network administrator capability) to properly configure, tune, and operate *ssh* for other application software.

In brief, security technologies are available and developed to provide network traffic security, but they require network security administrator skills to use them properly, and they are designed and implemented for specific application or operated mainly at network gateway devices. Therefore, for NARA to achieve essential IA services for sensitive ERA in transit, the end-to-end encryption and authentication requirements should be implemented at the computer system level.

To meet NARA's technical requirements for having end-to-end encryption and authentication at the computer system level, Army Research Laboratory (ARL) developed a secure communication network middleware called "Secure Link" capable of providing essential IA services for accessing or transferring sensitive ERA between any two networked computers. This report documents the development of ARL Secure Link.

2. ARL Secure Link Middleware

ARL developed for NARA the Secure Link middleware based on the specified functional behavior and technical requirements documented in the ARL memorandum report entitled "Functional Requirements Assessment of Secure Link" (*8*).

2.1 Design Architecture

The ARL Secure Link was designed as daemon program (a program executed in the background) and performed three main functions: authentication, encryption, and encapsulation. Its functions are to secure network traffic of selected network applications between two computers that run the Secure Link middleware. The authentication and encryption were achieved by using OpenSSL library (*1*), open source software that has been certified by the National Institute of Standards and Technology (NIST). ARL developed the encapsulation function ARL to achieve one tunnel mechanism with one endpoint at one end but many endpoints at the end. The special tunneling allows encapsulation of network packets from different destinations coming from one endpoint but distributing many endpoints based on destination IP addresses.

The network security provided by ARL Secure Link was achieved first relying on the Netfilter (*2*) (Linux security firewall software) to channel selected network traffic of application software to ARL Secure Link. After receiving network packets from selected application software, ARL Secure Link would perform a peer-to-peer (P2P) authentication with the specified destination host, based on the destination IP address of network packets, to mutually authenticate each other and generate a symmetric session key to be used for encryption function between two host computers running ARL Secure Link. Then, the application network traffic would be encrypted using the generated session key and passed to encapsulation process to tunnel the encrypted traffic to the destination. Upon receiving encapsulated traffic, the destination host would de-tunnel encapsulated traffic and decrypt ciphered traffic and then finally forward deciphered network traffic to application software.

Figure 1 depicts the functional and operational diagram of Secure Link relative to the application software, firewall, IP routing engine, and local area network of two networked computer systems.

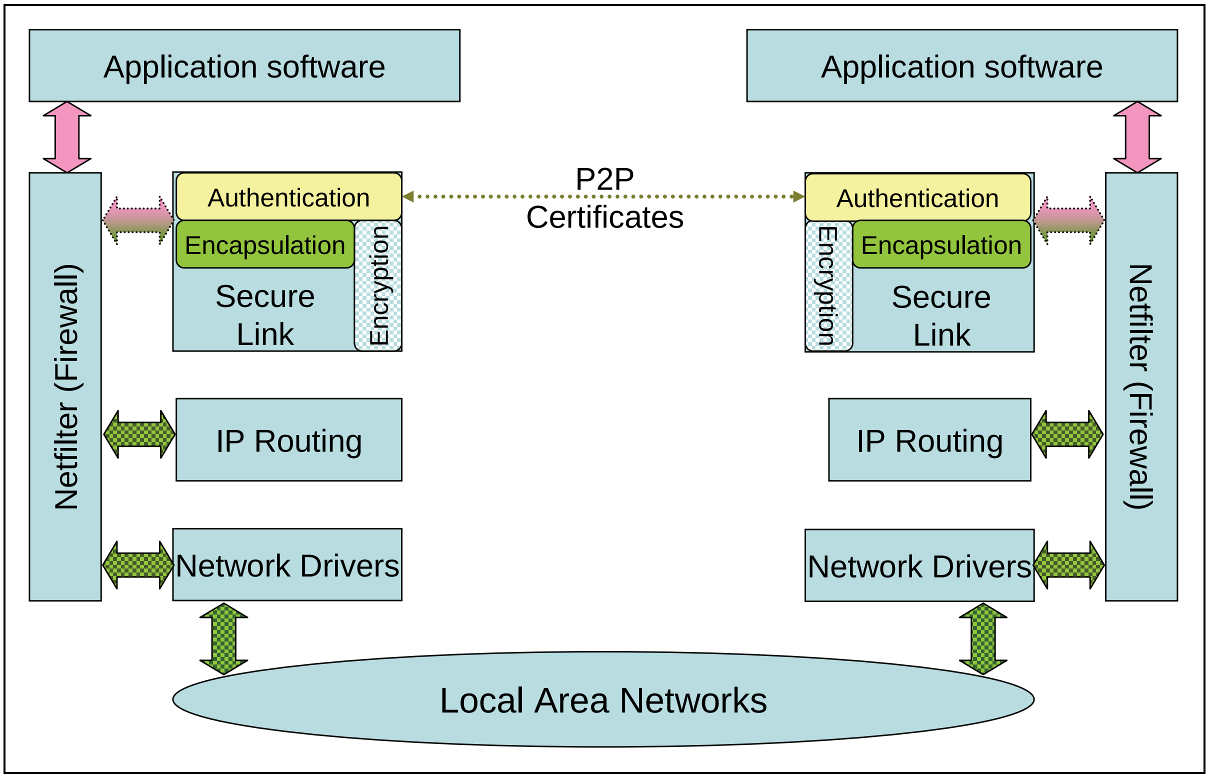

Figure 1. Functional and operational diagram of Secure Link in two networked computers.

2.2 Implementation Requirements

As described above, the Secure Link middleware operates like a firewall, but its function is to secure certain desired network traffic by diverting them to a secure tunnel. The secure tunnel will be established on demand between two computer systems running Secure Link to channel selected network traffic between them. Two computers will mutually authenticate each other before establishing the secure tunnel, and cryptographic protocols will be used to encrypt network traffic to form the secure tunnel.

To minimize costs and efforts, the implementation of the Secure Link relied on open source software, such as Linux Netfilter, OpenSSL library, VTun (*3*), and Universal TUN/TAP (*4*). As a result, the current implementation of the Secure Link supports only Linux-based systems, and secures only unicast network traffic using the Internet protocols (IP). The implementation of Secure Link requires a rebuild of a Linux kernel (especially Linux kernel 2.6.20 at the current development) to take advantage of additional features provided by Netfilter, such as ROUTE TARGET. The OpenSSL library is used for authentication using peer-to-peer method and encryption using AES (*5*) (Advanced Encryption Standard) and Blowfish (*6*) algorithm (Blowfish is a default algorithm). VTun and Universal TUN/TAP provide a framework to develop the Secure Link middleware especially creating secure tunnels.

During the process of implementing the Secure Link, ARL successfully developed a method of generating public key infrastructure (PKI) for use with peer-to-peer method. The successful development of PKI certificates for peer-to-peer authentication (instead of the typical client-and-server authentication) provides more versatility in using the development of the special secure tunnel with one endpoint to many endpoints. The method for generating the peer-to-peer cryptographic certificates was based on the method and procedure for creating client-server certificates and developed by the same author (*9*).

2.3 Operation Outline

In order to properly use the current development version of the ARL Secure Link middleware for end-to-end encryption and authentication between Linux computer systems, the system administrators must first obtain peer-to-peer PKI certificates for each participating computer system using ARL developed software. Then, the system administrators of the systems, which are built with Linux kernel 2.6.20 and capable of redirecting selected network traffic by using *iptables*, should start the ARL Secure Link middleware as a Linux daemon called SLD. From this point, the system administrators can select any desired network traffic applications to be protected by using *iptables* commands to redirect selected network traffic (specified by transport port number of IP traffic, e.g., TCP port 80, UDP 43, or, …) to SLD. Upon receiving any redirected network traffic, the SLD will authenticate destination hosts using P2P PKI certificates if the destination hosts are not verified before. If destination hosts are validated, the encryption/decryption and tunneling (encapsulation)/detunneling mechanism are used by SLD to securely send/receive selected network traffic.

To facilitate the operation of the ARL Secure Link middleware on Linux systems, ARL developed a graphical method (*7*) for executing SLD and selecting or deselecting desired network traffic. The system administrator can choose to secure desired network traffic in one direction only, e.g., sending or receiving direction, by having one communicating system not route desired network traffic through SLD. Communicating systems can still communicate with any hosts without SLD, as long as network traffic destining to those hosts not routing through SLD.

Figure 2 shows a demonstration setup using the ARL Secure Link middleware in a network environment (which simulates the Internet) of six LANs, five routers, and four communicating systems (*cavalier*, *colonial*, *hokie*, and *patriot*). Three communicating systems (*cavalier*, *colonial*, and *patriot*) have P2P certificates from the same PKI and run ARL Secure Link middleware as specified by the circled 'sld', sld. Using a network traffic analyzer (e.g., tcpdump) at router *bulldog* and *tiger*, ARL verified and confirmed that desired network traffic, such as ping, nfs, and *ssh* application, among systems *cavalier*, *colonial*, and *patriot* were encrypted and encapsulated inside UDP packets. But, network traffic to/from *hokie*, which did not run SLD, was clearly detected and eavesdropped by the network traffic analyzer.

Secure Link can cause network applications to experience additional network latency due to the central processing unit (CPU) processing for encryption and routing, but the appreciably increased network latency depends on the processing power of the system CPU. The additional network latency is very typical for the operation of network security whether its operation is executed at the computer system level or at the router level.

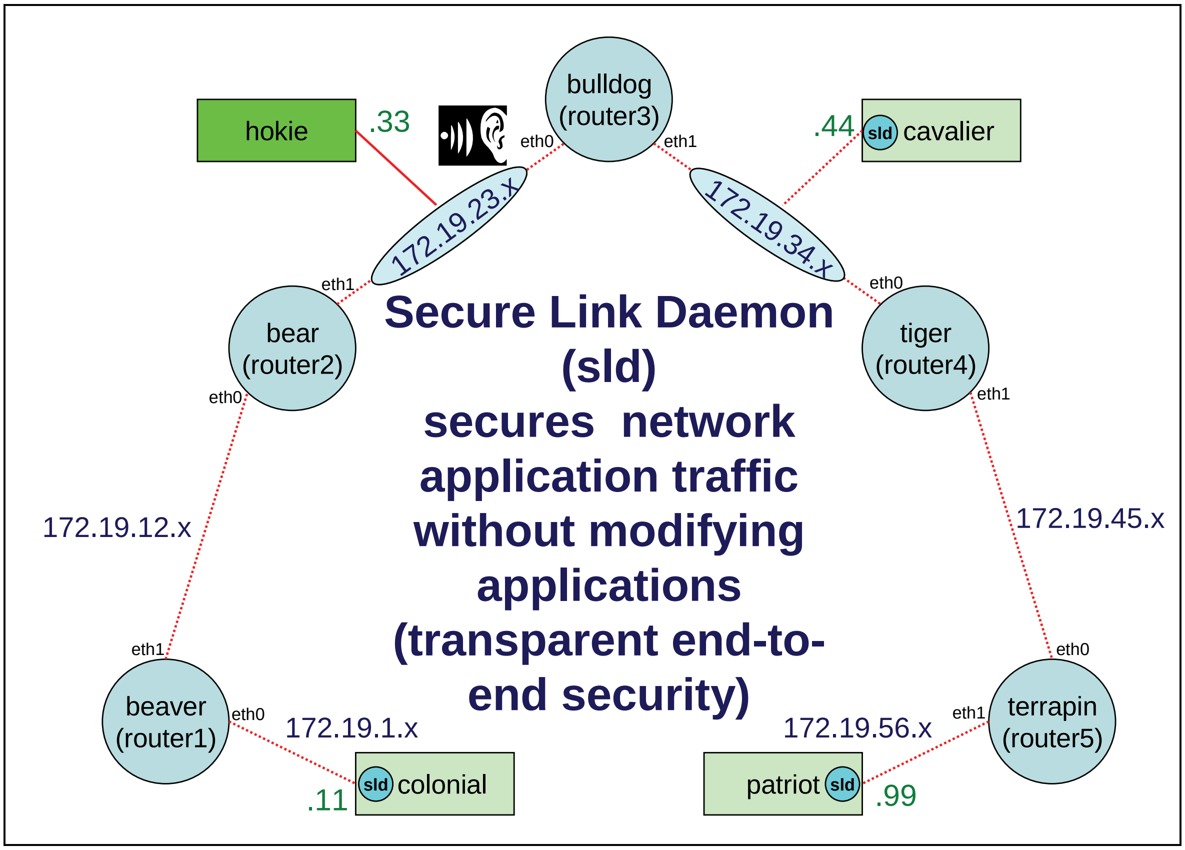

Figure 2. A demonstration setup of using the ARL Secure Link middleware in an inter-network environment.

3. Benefits

As shown in the demonstration, the use and operation of the ARL Secure Link middleware is simple and easy. It provides authentication and end-to-end encryption services for NARA application in securing ERA information in transit. Networked computer systems executing SLD (Secure Link daemon) can securely communicate with each other by first authenticating each other and then encrypting their exchanged network traffic through the secure tunnel. This secure communication can be expanded to other network applications running in these computers as selected or configured by the system administrator. This allows network applications such as telnet (remote logging in a computer), ftp (transferring files), or *nfs* (remote accessing file systems) to be securely operated and used among networked computer systems without any modification to the application software and their operations. The ARL Secure Link middleware can operate independently and concurrently with other security application software (e.g., *ssh*, *scp*, or *https*) without any adjustments.

By using the peer-to-peer authentication, the ARL Secure Link provides more flexibility in authenticating computer systems using the same P2P PKI certificates for various communicating scenarios such as client-to-server, client-to-client, server-to-server, client-to-many-clients, client-to-many-servers, or server-to-many-servers.

The execution of the ARL Secure Link middleware in a computer system can lessen the use of individual security application software (e.g., *ssh*, *scp*, *ssh* fs, or Oracle with security options) without sacrifice network communication security. This reduced usage of extraneous security application software will free up computer resources such as CPU cycles, random access memory (RAM), and disk storage and reduce computer costs such as software acquisition, installation, and maintenance (update and upgrade). Ultimately, the use of the ARL Secure Link will reduce the time and effort required to monitor security, thereby increasing system efficiency.

4. Conclusion

The U.S. Army Research Laboratory successfully developed the ARL Secure Link middleware for the U.S. National Archives and Records Administration in the effort to build a secure distributed computing environment for processing sensitive electronic records archives (ERA). A demonstration setup of computer nodes connected through inter-networks also successfully verified the security aspects of the ARL Secure Link, as defined in the Functional Requirements Assessment of Secure Link, ARL-MR-663. The development of Secure Link also yields a successful implementation of peer-to-peer authentication method and its PKI-certificate-generating method which have not been successfully implemented or publicly available based on Internet searches.

5. References

1. Openssl, “The Openssl Project” (http://www.openssl.org, accessed June 16, 2008).
2. Netfilter (http://www.netfilter.org, accessed June 16, 2008).
3. Vtun (http://vtun.sourceforge.net, accessed June 16, 2008).
4. Universal TUN/TAP (http://vtun.sourceforge.net/tun, accessed June 16, 2008).
5. Advanced Encryption Standard (http://en.wikipedia.org/wiki/Advanced_Encryption_Standard, accessed Jun16, 2008).
6. Blowfish (http://en.wikipedia.org/wiki/Blowfish_(cipher), accessed Jun16, 2008).
7. Nguyen, Binh. *A Graphical Method for Running the ARL Secure Link Middleware*; ARL-TR-4353; U.S. Army Research Laboratory: Adelphi, MD, January 2008.
8. Luu, Brian. *Functional Requirements Assessment of Secure Link*; ARL-MR-663; U.S. Army Research Laboratory: Adelphi, MD, March 2007
9. Nguyen, Binh. *Issuing Cryptographic Certificates Using OpenSSL*; ARL-MR-655; U.S. Army Research Laboratory: Adelphi, MD, December 2006.

No. of Copies	Organization
1 ELEC	ADMNSTR DEFNS TECHL INFO CTR ATTN DTIC OCP 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	DARPA ATTN IXO S WELBY 3701 N FAIRFAX DR ARLINGTON VA 22203-1714
1 CD	OFC OF THE SECY OF DEFNS ATTN ODDRE (R&AT) THE PENTAGON WASHINGTON DC 20301-3080
1	US ARMY RSRCH DEV AND ENGRG CMND ARMAMENT RSRCH DEV AND ENGRG CTR ARMAMENT ENGRG AND TECHNLGY CTR ATTN AMSRD AAR AEF T J MATTS BLDG 305 ABERDEEN PROVING GROUND MD 21005-5001
1	US ARMY TRADOC BATTLE LAB INTEGRATION & TECHL DIRCTRT ATTN ATCD B 10 WHISTLER LANE FT MONROE VA 23651-5850
1	PM TIMS, PROFILER (MMS-P) AN/TMQ-52 ATTN B GRIFFIES BUILDING 563 FT MONMOUTH NJ 07703
1	US ARMY INFO SYS ENGRG CMND ATTN AMSEL IE TD F JENIA FT HUACHUCA AZ 85613-5300
1	COMMANDER US ARMY RDECOM ATTN AMSRD AMR W C MCCORKLE 5400 FOWLER RD REDSTONE ARSENAL AL 35898-5000
5	NATL ARCHIVES & RECORDS ADMIN ELECT RECORDS ARCHIVES PROG MGMT OFC ATTN R CHADDUCK 8601 ADELPHI RD COLLEGE PARK MD 20740-6001
1	US GOVERNMENT PRINT OFF DEPOSITORY RECEIVING SECTION ATTN MAIL STOP IDAD J TATE 732 NORTH CAPITOL ST NW WASHINGTON DC 20402
1	US ARMY RSRCH LAB ATTN AMSRD ARL CI OK TP TECHL LIB T LANDFRIED BLDG 4600 ABERDEEN PROVING GROUND MD 21005-5066
1	DIRECTOR US ARMY RSRCH LAB ATTN AMSRD ARL RO EV W D BACH PO BOX 12211 RESEARCH TRIANGLE PARK NC 27709
6	US ARMY RSRCH LAB ATTN AMSRD ARL CI N G RACINE ATTN AMSRD ARL CI NT B NGUYEN ATTN AMSRD ARL CI NT B RIVERIA ATTN AMSRD ARL CI OK T TECHL PUB ATTN AMSRD ARL CI OK TL TECHL LIB ATTN IMNE ALC IMS MAIL & RECORDS MGMT ADELPHI MD 20783-1197
1 ELEC	US FEDERAL COMMUNICATIONS COMMISSION (FCC) ATTN B LUU

TOTAL: 23 (2 ELEC, 1 CD, 20 HCS)