GENERIC SECURITY FRAMEWORK FOR CLOUD COMPUTING USING "CRYPTONET"

Manisha Dawra[1], Ramdev Singh[2]

[1] *Al-Falah School of Engg. & Tech.,*
Vill-Dhauj, Ballabgarh-Sohna Road, Faridabad, Haryana (INDIA)-121004
[2]*Delhi Engineering College*
Vill-Ladiyapur, Ballabgarh-Sohna Road, Faridabad, Haryana (INDIA)-121004

IJARSE

ABSTRACT

In the last two decades there were many activities and contributions to protect data, messages other resources in computer networks, to provide privacy of users, reliability, availability and integrity of resources, and to provide other security properties for network environments and applications. Governments, international organizations, private companies and individuals are investing a great deal of time, efforts and budgets to install and use various security products and solutions. However, in spite of all these needs, activities, on-going efforts, and all current solutions, it is general belief that the security in today networks and applications is not adequate(1).

At the moment there are two general approaches to network application's security. One approach is to enforce isolation of users, network resources, (3) and applications. In this category we have Solutions like firewalls, intrusion–detection systems, port scanners, spam filters, virus detection and elimination tools, etc. The goal is to protect resources and applications by isolation after their installation in the operational environment. The second approach is to apply methodology, tools and security solutions already in the process of creating network applications. This approach includes methodologies for secure software design, ready–made security modules and libraries, rules for software development process, and formal and strict testing procedures(9).

The goal is to create secure applications even before their operational deployment. Current experience clearly shows that both approaches failed to provide an adequate level of security, where users would be guaranteed to deploy and use secure, reliable and trusted network applications.

Security Provider, Security Protocols, Generic Security Server, Security SDKs, and Secure Execution Environment. They are all mainly engine components of our security system and they provide the same set of cryptographic and network security services to all other security enhanced applications. Furthermore, for our individual security objects and also for larger security systems, in order to prove their structural and functional correctness, we applied deductive scheme for verification and validation of security systems.

Keywords: Security, Encryption, Generic, Abstraction, Cryptography, Environment, Mechanism, E-Mail.

I INTRODUCTION

CryptoNET is an integrated secure collaborative environment comprising the most popular standalone and distributed applications and associated security protocols. We have created several client components, such as Secure Station Manager (equivalent to Windows Explorer),

Secure E-Mail Client, Secure Documents Manager (security extensions of Open Office), and Secure Web Browser. In addition to those workstation components, we also designed and implemented corresponding servers: Secure E-Mail (SEM) Server, Secure Library Server, Secure Web Server and Secure Software Distribution Server. Security protocols between clients and servers are Strong Authentication, SAML–based Single-Sign-On, Secure Sessions, and some application– specific security protocols. All our applications and security protocols use functions and credentials of our single Generic Security Provider, which also transparently uses FIPS 201 Personal Identity Verification (PIV) [5] smart cards, if they are configured and attached. The components of our CryptoNET environment may also be connected to our cloud security infrastructure, so standard network security protocols, such as certification protocol [6], SAML authorization protocol, secure sessions, etc., are also supported in a Large - scale network environment.

II RESEARCH CONTRIBUTIONS

2.1 Generic Security Provider

Generic Security Provider is an engine component in our system. It provides a comprehensive set of security services, mechanisms, encapsulation methods, and security protocols for other components of our security system and for secure applications. The Provider is structured in four layers; each layer provides services to the upper layer and the top layer provides services to applications. Security services reflect requirements derived from a wide range of applications: from small desktop applications to large distributed enterprise environments. Starting from an abstract model, we describe design and implementation of an instance of the provider comprising various generic security modules: symmetric key cryptography, asymmetric key cryptography, hashing, encapsulation, certificates management, creation and verification of signatures, and various network security protocols. We describe the properties, extensibility, flexibility, abstraction, and compatibility of the Security Provider which is implemented using Java.

2.2 Generic Security Protocols

Generic Security Protocols play an important role for implementing security services in distributed applications and cloud computing environments. In this contribution, we designed several security protocols. They are based on the concepts of generic security objects and on a modular approach. The objects of security protocols are complete in terms of their functionality, so each object provides features to client and server applications. The protocols are designed using on well-established secure technologies and standards in order to make their host components interoperable with other components. Some of our authentication protocols are specifically designed for a specific

IJARSE

operating system, while other protocols are platform independent and generic. Therefore, they can be integrated with any application for secure communication, authorization, key distribution, Single-Sign-On and strong authentication. These protocols are based on our Generic Security Provider in order to perform cryptographic functions and communications with smart cards. In addition, these protocols are generic what makes them easy to use by developers for building secure cloud computing applications(10).

2.3 Generic Security Server

Generic Security Server is also engine component which provides basic structure to implement secure application servers. It is designed as a template which provides complete set of standard security and administration functions along with a number of extended security functions and features. These functions are based on well-established security standards and services.

It provides basic structure for developers in order to develop customized Secure Application Servers. We already implemented several initialization and management functions and several administrative actions. We also included APIs and libraries for cryptographic functions and security protocols in order to provide the same set of security services for all instances of Secure Application Servers. The structure of our Generic Security Server is flexible and it is available in the form of Eclipse-plug-ins, which is easy to extend according to customization requirements of each application.

2.4 Secure SDK

Secure SDK is a set of various security components which are protected using strong encryption techniques. For protection of software modules, we designed a solution using strong encryption techniques. This solution comprises Secure Software Distribution Server and Web Server in order to generate and distribute protected software modules only to authorized users (8).

Our solution encapsulates these modules in the form of specially designed Extensible Markup Language (XML) file which represents general syntax of protected software modules. Secure SDK and encapsulation of software modules are based on well-established secure technologies and standards, like FIPS 201 (PIV) smart cards, FIPS 196 strong authentication, and authorization policies based on eXtensible Access Control Markup Language (XACML).

2.5 Secure Execution Environment

Secure Execution Environment is also key component of our system. It executes protected software modules in controlled environment. In our design, all software modules and all other components of the CryptoNET are encrypted in order to protect them against reverse engineering, illegal tempering, program-based attacks, BORE (Break-Once-Run-Everywhere) attack, and unauthorized use of software. We extended standard execution environment with special security features and functions (7). Our extended Secure Execution Environment supports standard security services and network security protocols. These are: transparent handling of certificates, use of FIPS-201 compliant smart cards, Single-Sign-On protocol, strong authentication protocol, and secure sessions.

IJARSE

III CRYPTONET SYSTEMS

3.1 Crypto NET: Secure E-mail System

This section describes the design and implementation of a secure, high assurance and very reliable E-mail system(11). The system handles standard E-mail security services – signing and encryption of E-mail letters and, in addition, provides a number of extended and innovative security features. These new features are: transparent handling of certificates, strong authentication between Secure E-Mail Client and Secure E-Mail Server, archiving and recovery of encrypted address books, simple and secure handling of cryptographic keys, security sessions management, tracking of E-mail letters using confirmation messages, elimination of spam messages, prevention of fraudulent and infected attachments, and usage of smart cards. The system is based on the concepts of proxy architecture that makes it compatible with existing E-mail infrastructure. We also used XACML–based authorization policies at the sending and receiving Secure E-Mail (SEM) Servers in order to provide complete protection against spam. In our system, these policies are enforced by the Policy Enforcement Point (PEP), a component of the SEM server. In order to interconnect Secure E-mail systems deployed in individual domains, we introduced new infrastructure-level servers in order to develop trust between domains, exchange SEM registration information, and certify and verify domain names.

3.2 Crypto NET: Secure Web System

Our Secure Web System represents the design and implementation of a comprehensive system for protection of Web content stored at Web Servers, for execution of protected Web pages, and for their distribution to authorized users. We introduced additional components and added extended security features to a standard Web Server in order to provide confidentiality and integrity of Web content. We also designed and implemented an extended secure execution environment for Java Web Server, which is capable to process and execute different types of encrypted and digitally signed Web pages encapsulated in PKCS7Signed And Enveloped Data format. This system follows component-based architecture what makes it compatible with the exiting Web infrastructure (4).

3.3 Crypto NET: Secure Documents System

We designed a Secure Documents System in order to protect documents in local and collaborative environment (8). Our Secure Documents System comprises a set of security functions, features and components functioning as security extensions of the Open Office. The extended security features are: protection of documents in local environments, distribution of secure documents to group members, group key management, and enforcement of section level XACML policies for access control, smart card-based cryptographic functions, and transparent handling of security credentials. The design of the system is based on our generic security objects and plug-in architecture, what makes it easy to extend and integrate with existing document systems. In addition, Secure Documents System is linked to our cloud security infrastructure which provides security services in global environments by using certificates and SAML technologies (12).

IJARSE

IV CLOUD SECURITY INFRASTRUCTURE

Cloud security infrastructure is an environment in which several standard security components are deployed as services. These components are: Local Certification Authority, Policy PKI Server, Top PKI Server, Identity Management System (IDMS), XACML Policy Server, and Strong Authentication Server.

V CRYPTOOL

CrypTool is open source crypto library [14]. The aim of this project is to provide a platform for e-learning of cryptography and cryptanalysis in a modular and easy-to-use way. Currently, the team of CrypTool is working on JCrypTool and CrypTool 2.0. CrypTool 2.0 is based on C++ programming language, while JCrypTool is based on Java and Eclipse. JCrypTool provides security features using modular plug-in approach. It is structured in the form of plug-ins, which are structured based on their functionality e.g. logging, core engine, data objects, etc. The objective of this separation is to provide flexibility and extensibility to end-users.

VI CONCLUSION

The Integrated Secure Workstation provides comprehensive set of security services for PC environments and selected applications. The main principle was to cryptographically protect local IT resources, potations and messages. It transparently handles security functions and services. The design of ISW is based on the concept of generic security objects, which can be used by any application included in the Secure Workstation. Our future research topics are security issues of application servers which will enable CryptoNET to provide a omplete secure framework for network applications.

REFERENCES

[1]. http://www.openssl.org/docs/OpenSSL, http://www.openssl.org/docs/, [visited: January 2009].

[2]. RSA Security, Inc. "*BSAFE: A Cryptographic Toolkit*", Library Reference Manual Version 4.0 http://www.rsa.com/products/bsafe/documentation /cryptoc_411_ reference.pdf.

[3]. SUN Corporation, "*Java Cryptographic Extensions (JCE)*", www.sun.com, [visited: February 2009].

[4]. Microsoft Corporation, "*Cryptographic Services Provider (CSP)*", www.microsoft.com, [visited: February 2009].

[5]. FIPS PUB 201-1, *"Personal Identity Verification (PIV) of Federal Employees and Contractors*", Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, March 2006

[6]. Adams, S. Farrell, T. Kause, T. Mononen, "*RFC 4210, Internet X.509 Public Key Infrastructure Certificate Management Protocol*", September 2005.

[7]. J. Linn, "*Generic Security Service Application Program Interface*", RFC-2743, RSA Laboratories,

IJARSE

January 2000.

[8]. K. Brown, "*Explore the security support provider interface using the SSPI workbench Utility*", MSDN Magazine, Aug. Available at http://msdn.microsoft.com/msdnmag/issues/0800/Security/Security0800.asp, 2000.

[9]. Denis Piliptchouk, "*Java vs. .NET Security, Part 2 Cryptography and Communication*"http://www.Onjava.com/pub/a/onjava/2003/12/10/javavsdotnet.html?page=1, [visited: October 2008].

[10]. Microsoft Inc., "*Microsoft CryptoAPI and Cryptographic Service Providers*", http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/distrib/dscj_mcs _xxgl.mspx?m fr =true build date on11/19/2009.

[11]. Scott Oaks, "*Java Security (Security Providers)",* Chapter 8, Java Security", First edition, ISBN 1-56592-403-7E, publisher O'Reilly Media and published in May 1998.

[12]. Y. Zhang, A. Timkovich, and J. Peck, "*IBM Security Providers: An Overview*", Oct. 2004.

[13]. Institute for Applied Information Processing and Communication (IAIK), http://jce.iaik.tugraz.at/sic /products, [visited: December 2008].

[14]. J. Lee , J. Kim, J. Na, and S. Sohn, "ESES/j-Crypto and its application" published in Proceedings of IEEE International Symposium on Industrial Electronics (ISIE 2001), Pusan South Korea, pp. 1373-1377, vol.2, 2001.

IJARSE