CryptoNET

Generic Security Framework for Cloud Computing Environments

ABDUL GHAFOOR ABBASI

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the degree of Doctor in Philosophy on Monday the 16th May 2011, at 1:00 p.m. in Sal-D, Forum/KTH, Isafjordsgatan 39, Kista.
Abstract

The area of this research is security in distributed environment such as cloud computing and network applications. Specific focus was design and implementation of high assurance network environment, comprising various secure and security-enhanced applications. “High Assurance” means that
- our system is \textit{guaranteed} to be secure,
- it is \textit{verifiable} to provide the complete set of security services,
- we prove that it always \textit{functions correctly}, and
- we justify our claim that it \textit{can not be compromised} without user neglect and/or consent.

We do not know of any equivalent research results or even commercial security systems with such properties. Based on that, we claim several significant research and also development contributions to the state–of–art of computer networks security.

In the last two decades there were many activities and contributions to protect data, messages and 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.

At the moment there are two general approaches to network application’s security. One approach is to enforce \textit{isolation} of users, network resources, 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. 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.

Therefore, in the current situation, it is obvious that a new approach and a new thinking towards creating strongly protected and guaranteed secure network environments and applications are needed. Therefore, in our research we have taken an approach completely different from the two mentioned
above. Our first principle is to use *cryptographic protection of all application resources*. Based on this principle, in our system data in local files and database tables are encrypted, messages and control parameters are encrypted, and even software modules are encrypted. The principle is that if all resources of an application are always encrypted, i.e. “enveloped in a cryptographic shield”, then
- its software modules are not vulnerable to malware and viruses,
- its data are not vulnerable to illegal reading and theft,
- all messages exchanged in a networking environment are strongly protected, and
- all other resources of an application are also strongly protected.

Thus, we strongly protect applications and their resources before they are installed, after they are deployed, and also all the time during their use.

Furthermore, our methodology to create such systems and to apply total cryptographic protection was based on the *design of security components in the form of generic security objects*. First, each of those objects – data object or functional object, is itself encrypted. If an object is a data object, representing a file, database table, communication message, etc., its encryption means that its data are protected all the time. If an object is a functional object, like cryptographic mechanisms, encapsulation module, etc., this principle means that its code cannot be damaged by malware. Protected functional objects are decrypted only on the fly, before being loaded into main memory for execution. Each of our objects is complete in terms of its content (data objects) and its functionality (functional objects), each supports multiple functional alternatives, they all provide transparent handling of security credentials and management of security attributes, and they are easy to integrate with individual applications. In addition, each object is designed and implemented using well-established security standards and technologies, so the complete system, created as a combination of those objects, is itself compliant with security standards and, therefore, interoperable with exiting security systems.

By applying our methodology, we first designed enabling components for our security system. They are collections of simple and composite objects that also mutually interact in order to provide various security services. The enabling components of our system are: 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. We used the following principle: “if individual objects are verified and proven to be secure, if their instantiation, combination and operations are secure,
and if protocols between them are secure, then the complete system, created from such objects, is also verifiably secure”. Data and attributes of each object are protected and secure, and they can only be accessed by authenticated and authorized users in a secure way. This means that structural security properties of objects, upon their installation, can be verified. In addition, each object is maintained and manipulated within our secure environment so each object is protected and secure in all its states, even after its closing state, because the original objects are encrypted and their data and states stored in a database or in files are also protected.

Formal validation of our approach and our methodology is performed using Threat Model. We analyzed our generic security objects individually and identified various potential threats for their data, attributes, actions, and various states. We also evaluated behavior of each object against potential threats and established that our approach provides better protection than some alternative solutions against various threats mentioned. In addition, we applied threat model to our composite generic security objects and secure network applications and we proved that deductive approach provides better methodology for designing and developing secure network applications. We also quantitatively evaluated the performance of our generic security objects and found that the system developed using our methodology performs cryptographic functions efficiently.

We have also solved some additional important aspects required for the full scope of security services for network applications and cloud environment: manipulation and management of cryptographic keys, execution of encrypted software, and even secure and controlled collaboration of our encrypted applications in cloud computing environments. During our research we have created the set of development tools and also a development methodology which can be used to create cryptographically protected applications. The same resources and tools are also used as a run–time supporting environment for execution of our secure applications. Such total cryptographic protection system for design, development and run–time of secure network applications we call CryptoNET system. CryptoNET security system is structured in the form of components categorized in three groups: Integrated Secure Workstation, Secure Application Servers, and Security Management Infrastructure Servers. Furthermore, our enabling components provide the same set of security services to all components of the CryptoNET system.

Integrated Secure Workstation is designed and implemented in the form of a collaborative secure environment for users. It protects local IT resources, messages and operations for multiple applications. It comprises four most commonly used PC applications as client components: Secure Station Manager (equivalent to Windows Explorer), Secure E-Mail Client, Secure Web Browser, and Secure Documents Manager. These four client components for their security extensions use functions and credentials of the enabling components in order to provide standard security services (authentication, confidentiality,
integrity and access control) and also additional, extended security services, such as transparent handling of certificates, use of smart cards, Strong Authentication protocol, Security Assertion Markup Language (SAML) based Single-Sign-On protocol, secure sessions, and other security functions.

Secure Application Servers are components of our secure network applications: Secure E-Mail Server, Secure Web Server, Secure Library Server, and Secure Software Distribution Server. These servers provide application-specific services to client components. Some of the common security services provided by Secure Application Servers to client components are Single-Sign-On protocol, secure communication, and user authorization. In our system application servers are installed in a domain but it can be installed in a cloud environment as services. Secure Application Servers are designed and implemented using the concept and implementation of the Generic Security Server. It provides extended security functions using our engine components. So by adopting this approach, the same sets of security services are available to each application server.

Security Management Infrastructure Servers provide domain level and infrastructure level services to the components of the CryptoNET architecture. They are standard security servers, known as cloud security infrastructure, deployed as services in our domain level could environment.

CryptoNET system is complete in terms of functions and security services that it provides. It is internally integrated, so that the same cryptographic engines are used by all applications. And finally, it is completely transparent to users – it applies its security services without expecting any special interventions by users. In this thesis, we developed and evaluated secure network applications of our CryptoNET system and applied Threat Model to their validation and analysis. We found that deductive scheme of using our generic security objects is effective for verification and testing of secure, protected and verifiable secure network applications.

Based on all these theoretical research and practical development results, we believe that our CryptoNET system is completely and verifiably secure and, therefore, represents a significant contribution to the current state-of-the-art of computer network security.

**Key Words**

Generic Security Object, CryptoNET, Security, Framework