6. Security Mechanisms in the Comcute System

Piotr Szpryngier

Gdańsk University of Technology

Faculty of Electronics, Telecommunications and Informatics

Computer Systems Architecture Department

e-mail: piotrs@eti.pg.gda.pl

Abstract

The aim of this paper is pointing out the basic security problems and mechanisms in the Comcute system – maintenance system of large computing power in the face of critical crisis. Moreover security mechanism and tools useful to apply in laboratory model as well as target version of the Comcute system are presented.

Keywords: *grid computing, parallel programming, volunteer computing, security policy, confidentiality, authenticity, Public Key Infrastructure, PK certificate.*

6.1. Assumptions and security requirements

Laboratory Comcute system is used to investigate the distributed system properties and ability to disperse compute-intensive calculation over computers in local area network (laboratory) or over computers connected to Internet. Laboratory system model should evolve with time and intensity of research (according to the intentions of designers) probably into prototype of the target system. For this reason security mechanisms related to the prototype are still valid and important as well as for ready to operation system.

However we need to express some additional remarks. When the laboratory system usually works in the closed environment (university laboratories) and simulates behavior of the target system, then fully fledged system will be available and accessed on many layers from Internet. Moreover, the Comcute system temporarily will store some client data and information about results of computations. For this reason designing, planning and deploying of security policy is necessary. Security policy document should be approved by directorate of organization (owner of the target Comcute system) and available to all clients as some kind of calculations quality guarantee and expected good practices with client data and other client resources. Such security policy document should contain following items [1,4,6, 9,10]:

1. General provisions, basic legal constraints, glossary of items;
2. Authentication and identification rules;
3. List of available services and terms of usability;
4. Description of resources available during computations;
5. Risk analysis against security costs and the expected benefits;
6. Operation requirements for security (selected communication activities, protocols, provision rules of the services, etc.);
7. Security remedies, tools and remedies to achieve security targets;
8. Security and risk management (audit, control frequency, use of external auditor companies), staff and client policies.

Let assume following types of the Comcute users [2]:

- Configuration (architecture) administrators;
- Security administrators – privileges and roles (authorization) managers, etc.;
- System operators and managers (task configuration, computation flow management and control);
- Clients – principals of calculations.

Depending on the number of users different models of access control may be considered. For smaller number of users (up to few hundred) one can choose more restricted model (like Bella-LaPadula or other Mandatory Access Control model) [6] or some milder (Discretionary Access Control with object separation) [5,6]. For bigger number of users (>1000) very expensive on introduction stage but easy to manage is Role Based Access Control (RBAC) model [5].

6.2. Information Security Management

General model of security management layers are shown in Tab. 6.1:

Tab. 6.1. The layers of general security management model

Security Management (applications, databases, EDE, e-mail, etc.)
Security Agents, Security Protocols (authentication, key management, etc.)
Security Services (confidentiality, integrity, non-repudation, etc.)
Security mechanisms (digital signature, authentication)
Basic Modules (algorithms, modes of operation)

In this chapter we are going to discuss some issues from layer 3 (security services) together with security mechanisms (layer 2 – commonly known and described) and partially security protocols (layer 4). All security management elements must be linked with security policy (which is out of scope in this chapter). However basic security modules like algorithms, modes of operation,

key generators, random and pseudo random number generators, etc. are perfectly described [3,4,5,6] and there is no need here to cover this area. Below there is some explanation how the basic cryptographic protocols used for building security mechanisms. The aim of first one – secure message transfer with encrypted session key [3,4,5,6] – is ensuring the confidentiality content of transmitted message or file. Additional side effect is ensuring receiver's authenticity of the message. It runs as follows:

A) Sender prepares message M, generates random session key K and creates cryptogram $C_K(M)$ using symmetric cryptographic algorithm of good quality like AES, Twofish, etc.

B) Sender retrieves Receiver's public key O_{pu} from trusted database or retrieves it from public key certificate signed by trusted Certification Authority (CA) and then creates cryptogram of session key encrypted with receiver's public key. Sender uses the same symmetric cryptographic algorithm in ECB mode as in the step A [3]. Additional cryptogram of session key $C_{Opu}(K)$ is concatenated with cryptogram $C_K(M)$ and all these enciphered messages are sent to Receiver. Please note that in one message can be placed many encrypted messages for many receivers, each containing session key encrypted with each receiver's public key.

C) Each receiver of the message deciphers session key K from cryptogram using corresponding private key O_{pr} (forming a pair with public key O_{pu}) and obtains $K=D_{Opr}(C_{Opu}(K))$.

D) In following step Receiver deciphers message M using session key K: $M=D_K(C_K(M))$

Digital signature protocol is second basic mechanism used to achieve authenticity, integrity and non-repudiation of communication. Digital signature (a cryptogram created with use of Sender's signing key) is *de facto* an appendix to a message. It runs as follows [4,11]:

A) Sender prepares message M, calculates hash of the message $H(M)$ using one-way function of good quality like SHA-1, SHA-256 and then, using some digital signature schema (like RSA, DSA) and signer's private key K_S, generates cryptogram (signature) $S_{Ks}(H(M))$.

B) Sender concatenates message M with digital signature $S(H(M))$ and sends forward whole packet to Receiver.

C) Receiver retrieves Sender's public key K_{Spu} from trusted database or retrieves it from public key certificate signed by trusted Certification Authority (CA), then calculates a value of one-way hash function $H(M')$ from received message and uses both these values ($H(M')$ and K_{Spu}) to validate authenticity of digital signature $S_{Ks}(H(M))$.

D) If signature is valid (for example in RSA schema this means that hash value of received message $H(M')$ is equal to hash value $H(M)$ deciphered with Sender's public key K_{Spu}), then accept message as original and credible. In other case a message is rejected as originated from unknown source or probably message was destroyed during transit.

Public Key Infrastructure (PKI) [4,11] is composed as a network of CA (Certification Authority), RA (Registration Authority) - servers used for user verification and registration altogether with security policy defining all public key management procedures. Public key certificate is a data structure, signed by issuer of certificate. This signed structure contain user identity data applying for certificate (and hence private key owner's identity consisting a pair with this public key placed in the PK certificate), public key (the mandatory part of certificate), certificate identification number, issuer description and signature. Certificate can also comprise many additional data: issue date, time and the period of validity, certificate (keys) destination (application), PKI standard version, application constraints, delegation of privileges, etc. It is very noteworthy fact that public key certificate strictly connect user with his private key (of course it must be hidden and kept safely). With this property, using public key contained in a certificate, we can verify the authenticity of all user operations with his/her private key (consisting a pair with this public key published in a certificate)[11].

6.3. Interface Security: Client –Comcute System

In this section we look at security issues of the Comcute client interface or communication protocols between W servers layer and outer world [2]. In the section of the Comcute system design and implementation documents entitled as "System Architecture – requirements from client point of view", this interface should allow for the implementation of following features:

- Problem defining altogether with necessary data and parameters,
- Source program defining and adaptation for distributed computing in Comcute system,
- Results and their status (validity code) reading,
- Additionally obtaining auxiliary information and data.

There are typical menaces in this area of system activity [3,5,6]:

- Eavesdropping, wiretapping – breach of confidentiality (passive attack),
- Destroying or modification of communication, spoofing, etc. (active attacks),
- Traffic analysis.

We can distinguish at least three alternate methods of client communication with the Comcute system:

a. Classical tiny interface (internet browser) – form to fill in taken from the Comcute webpage,
b. As a webservice, published and available for authorized clients only,
c. Other communication – any secure channel (direct personal communication, classical paper postage, delivery service, etc.).

Depending on the method of communication we can point out following mechanism necessary to achieve minimum security level:

a. For www (browser) interface – cryptographic tunnel SSL/TLS [5] and strong authentication methods - apart from login and password additional use, for example, of ZK (Zero Knowledge) identification protocol using smartcards [3]. Furthermore, it is worth considering additional limitations of communication traffic from only trusted nodes using properly configured firewalls to accept only chosen IP addresses. The SSL protocol and firewall are remedies for intruders and protect against eavesdropping and modification of communication, but do not secure against traffic analysis.
b. For webservices – publication of service and interface details in web UDDI registers (OASIS) is subject to restrictions defined by standard description [7]. Available services must be secured in the same way as in p. a (SSL/TLS and strong authentication). And again – these remedies do not secure against traffic analysis.
c. Other way of communication (direct, personal) can secure against traffic analysis. Additionally calculation service will be configured by operator. Such member of the Comcute system team can deploy many computations and for each one can select different latency of start point. Due to this property no one can establish for which client this calculations are deployed. In such case we can obtain protection against eavesdropping and modification of communication using the same remedies (SSL, operator authentication).

6.4. W Server Layer Security

In this section the *W* server layer security issues are described. The *W* servers are internal system nodes used for calculation task (issued by client or system operator) management. In the section of the Comcute system design and implementation documents entitled as "System Architecture – problems and solutions", this system layer deals with following features:

- Task receiving and verification (program code, data, additional computation requirements),
- Data partitioning,
- Task execution (finishing) criteria definition,
- Eventually additional arbiter (judge) set up to evaluate the progress of calculation and to make some result of computation assessment, resolve conflicts, etc.
- Conversion (eventually) of program code provided by client into execution code accepted by computers of Internet users,
- Storage od result data and available it to client.

As we can note from this list of functionalities, *W* servers store the important data (input and results of calculation) of external client. Let assume that there is more than one group of W servers, located in different local area networks connected via Internet. Each group of W servers can cooperate with chosen group of S servers. There are many possible menaces to materialize: attacks against integrity of stored data (unauthorized modification, destroying), wiretapping (reading of data and results of computation, client identity

discovery). For these reasons data protection is very important and system designer must rethink it carefully on many levels:

- Local area network containing group of *W* servers must be protected with properly configured firewall to accept external communication with chosen and strictly defined IP addresses only.
- All communication between *W* servers is performed within SSL/TLS cryptographic tunnel using authentication for both sides.
- Transaction mechanism should be implemented for distributed database used by *W* servers or full replication (database copies at each node of the *W* layer) should be considered, possibly synchronous, together with distributed transaction management [8].
- Access control mechanism should delimit access to distributed database for local application only and for *W* server layer users (eventually for laboratory users of *S* layer).
- Access to *W* servers layer should be limited to authenticated and authorized users only.

6.5. Security Issues for Communication between *W* Server layer and *S* servers

In this section the *W* server layer and *S* severs interoperation (distribution of calculation task and collecting of the partial results) security issues are discussed. In the section of the Comcute system design and implementation documents entitled as „System Architecture – problems and solutions", in this communication link between system layers *W* and *S* there are following basic functionalities:

- Transmission of computational task (program code and data subpacket) from *W* server to *S* servers.
- Collecting of partial results from *S* servers by *W* layer.

Possible mainly attacks at this communication link are: destroying or replacement of information parts of the packets (program code and data needed for calculation), spoofing, eavesdropping and data swapping (Man-in-the-Middle - MiM attack). Considering this menaces we can note that protection of *W*-*S* communication link should consist of:

- Providing consistency and authenticity of transmitted data using digital signatures [3].
- Securing communication via SSL/TSL cryptographic tunnel or with use secure message transfer with encrypted session key, providing confidentiality and authentication of message receiver [3].
- To protect against spoofing and MiM attacks there is need for use of public key infrastructure for proper use and management of PK certificates [11].

6.6. The *S* Server Layer Security and Communication Link Protection between *S* Layer and Internet Users

In this section the *S* server layer (service hosts of www providers) security issues are described together with communication link protection between S layer and computers of internet users. *S* server usually controls and manages the subtask of computation received from one of *W* servers. In the section of the Comcute system design and implementation documents entitled as "System Architecture – problems and solutions", in this communication link between system layers *S* and *I* (Internet users) there are following basic functionalities:

- Computation subtask transmission (or pointer to calculation task data located in different server, for example advertisement services) from *S* server to internet user computer *I*.
- Computation partial results receiving from internet user computer *I* directly or indirectly – via ad server and in such case server *W* is notified only that calculation subtask was finished and results are accessible on other server.
- Transmission of calculation partial results from *S* server to *W* server or a message transfer from *S* server to *W* server with information where the partial result data are located.

We have to note that communication between internet user computer *I* and *S* server is out of control and can't be managed from *W* servers point of view (and of course the whole Comcute system). The Comcute system cannot force mandatory communication enciphering (task program code and data packet) without additional agreement (contract) signed by both sides: the Comcute system owner and *S* server owner (manager) and/or advertisement server owner. When there is no cryptographic protection of communication links between *S* server and internet user computer *I* then we have to deal with typical attacks like spoofing, breaches, eavesdropping and traffic analysis. Digital signature protocol can provide some protection against integrity attacks on transmitted data.

There is needed to discuss one more issue – what key (located in the public key certificate) we have to use for communication validity verification. To avoid necessary acceptance of public key certificate every time Internet user checks digital signature validity, this public key certificate should be signed and published by CA (Certification Authority) automatically recognized by every typical browser. It is necessary condition if we want not to stress internet user or engage her/him without any need for it.

6.7. Necessary PKI Infrastructure

Taking under consideration all previously discussed security issues we can note that there is need for two types of PK certificates:

- Public key certificate issued and signed by CA commonly and automatically accepted (known) by browser. The private key associated with public key contained in the such public key certificate can be used for signing program

code and data of calculation task sent to internet user computer from *W* server with use *S* server and eventually for signing all communication from *W* servers to *S* servers and, if there is such need, to advertisement servers also. We have to note that strong protection of private key is necessary to avoid compromise of whole system.

- Public key certificates used for signing and protection of communication within the Comcute system, managed by internal PK infrastructure. Master key (*root*) can be signed, but it is not necessary, by qualified public key certification authority. Typically one can use self-signed PK root certificate for internal use.

6.8. Conclusion Remarks

In chapter the basic mechanisms necessary for the Comcute system protection and its elements were presented. Due to well-known threats from Internet the security policy for Comcute system should be developed first to define ways of achieving its objectives and procedures in terms of risk and danger. While some elements of laboratory system (which is partially isolated from external world) may be omitted, however in the case of target system it is advisable and necessary to design and implementation of following elements of security:

- Security policy – document approved by the Board or the Comcute system owner;
- Public Key Infrastructure – the hierarchical structure for whole system and Simple Public Key Infrastructure (SPKI) for Internet layer;
- Mechanisms and protocols providing confidentiality and authenticity of communication.

References

1. NIST Documents NCSC-TG, 1986-1991.
2. COMCUTE – design and implementation documents, Faculty of ETI, GUT 2010-2012.
3. J. Menezes, P. C. van Oorschot, S. A. Vanstone - „Handbook of Applied Cryptography" CRC Press, 1997.
4. Schneier B. - „Applied Cryptography", J.Wiley&Sons, 1996.
5. Gollmann D. -„Computer Security", 3rd. ed., J.Wiley&Sons, 2011.
6. J. Stokłosa, T. Bilski, T. Pankowski – Data Security in Computer Systems (in Polish), PWN, 2001.
7. Nadalin, C. Kaler - OASIS Web Services Security: WS-Security Core Specification 1.1, http://www.oasis-open.org/committees/ download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf, 2006
8. M. T. Özsu and P. Valduriez - Principles of Distributed Databases (3rd ed.), Springer, 2011.

9. Bosworth, S., Kabay, M.E. (edit.) - Computer Security Handbook, 4th ed., J. Wiley&Sons, 2002.
10. J. Pieprzyk, T. Hardjono, J. Seberry – Theory of Computer Systems Security (in Polish), Helion, 2005.
11. Szpryngier, P.: SPKI – Public Key Infrastructure (in Polish). in: KASKBOOK. SaaS Technologies. ed. H. Krawczyk, Gdańsk: GUT, 2004.