International Journal of Trend in Research and Development, Volume 3(1), ISSN: 2394-9333
www.ijtrd.com

A Survey on Security Services and Mechanisms in Distributed Systems

[1]Istam Shadmanov, [2]Kamola Shadmanova and [3]Rakhimjon Rakhimov,
[1,2] Department of Information Technology, Bukhara State University, Uzbekistan
[3] Department of Information Technology, Urgench State University, Uzbekistan

Abstract *— Security plays an extremely important role in distributed systems. A distributed system must provide security services by which a wide range of security policies can be implemented, and the mechanisms that allow a variety of different security policies to be enforced. This paper about number of important design issues, that is need to be taken into account when implementing of security services and mechanisms.*

Keywords *— Security Issues, Security Mechanisms, Security Services, Security Policies, Layer Of Mechanisms.*

I. INTRODUCTION

In recent years, creating successful distributed systems requires how the system is designed, developed, maintained, supported, governed and protected. One could argue that security is one of the most difficult and important principles of distributed systems, as security needs to be pervasive throughout a system. A single design flaw with respect to security may render all security measures useless. Protecting of distributed systems from various types of attacks which given in [2, 4, 6] is generally considered a very difficult task. But, a distributed system, or any computer system for that matter, must provide security services and mechanisms, by which a wide range of security policy can be implemented.
Security service is a processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service. Security services include authentication, access control, data confidentiality, data integrity, non repudiation, and availability, which discussed in section 2.

Security mechanisms typically involve more than a particular algorithm or protocol. They also require that participants be in possession of some secret information (e.g., an encryption key), which raises questions about the creation, distribution, and protection of that secret information. There also may be a reliance on communications protocols whose behavior may complicate the task of developing the security mechanism. For example, if the proper functioning of the security mechanism requires setting time limits on the transit time of a message from sender to receiver, then any protocol or network that introduces variable, unpredictable delays may render such time limits meaningless.

In the section 3, we concentrate on the various mechanisms that are generally incorporated in distributed systems to support security. Building all kinds of security mechanisms into a system does not really make sense unless it is known how those mechanisms are to be used, and against what. This requires that we know about the security policy that is to be enforced.

II. TYPES OF SECURITY SERVICES

X.800 defines a security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers. Perhaps a clearer definition is found in RFC 2828, which provides the following definition: a processing or communication service that is provided by a system to give a specific kind of protection to system resources; security services implement security policies and are implemented by security mechanisms. X.800 divides these services into five categories and fourteen specific services are shown in Fig. 1. [7]

Security services that are placed in the middleware layer of a distributed system can be trusted only if the services they rely on to be secure are indeed secure. For example, if a secure RPC service is partly implemented by means of SSL, then trust in the RPC service depends on how much trust one has in SSL. If SSL is not trusted, then there can be no trust in the security of the RPC service. [1]

A. Authentication – is the assurance that the communicating entity is the one that it claims to be. The authentication service is concerned with assuring that a communication is authentic. In the case of a single message, such as a warning or alarm signal, the function of the authentication service is to assure the recipient that the message is from the source that it claims to be from. In the case of an ongoing interaction, such as the connection of a terminal to a host, two aspects are involved. First, at the time of connection initiation, the service assures that the two entities are authentic, that is, that each is the entity that it claims to be. Second, the service must assure that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties for the purposes of unauthorized transmission or reception.

1. ***Peer Entity Authentication***- provides for the corroboration of the identity of a peer entity in an association. Two entities are considered peers if they implement to same protocol in different systems; e.g., two TCP modules in two communicating systems. Peer entity authentication is provided for use at the establishment of, or at times during the data transfer phase of, a connection. It attempts to provide confidence that an entity is not performing either a masquerade or an unauthorized replay of a previous connection.
2. ***Data-Origin Authentication***- provides for the corroboration of the source of a data unit. It does not provide protection against the duplication or modification of data units. This type of service supports applications like electronic mail, where there are no prior interactions between the communicating entities.

In a connectionless transfer, provides assurance that the source of received data is as claimed.

B. Access Control: - is the prevention of unauthorized use of a resource (i.e., this service controls, which can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do). Access control is the ability to limit and control the access to host systems and applications via communications links. To achieve this, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

C. Data Confidentiality- is the protection of data from unauthorized disclosure. Confidentiality is the ability to conceal messages from a passive attacker so that any message communicated via the sensor network remains confidential. This is the most important issue in distributed systems security. With respect to the content of a data transmission, several levels of protection can be identified. The broadest service protects all user data transmitted between two users over a period of time. For example, when a TCP connection is set up between two systems, this broad protection prevents the release of any user data transmitted over the TCP connection. Narrower forms of this service can also be defined, including the protection of a single message or even specific fields within a message. These refinements are less useful than the broad approach and may even be more complex and expensive to implement. The other aspect of confidentiality is the protection of traffic flow from analysis. This requires that an attacker not be able to observe the source and destination, frequency, length, or other characteristics of the traffic on a communications facility.

1. ***Connection Confidentiality***- is the protection of all user data on a connection.
2. ***Connectionless Confidentiality*** - is the protection of all user data in a single data block.
3. ***Selective-Field Confidentiality*** – is the confiden-tiality of selected fields within the user data on a connection or in a single data block.
4. ***Traffic-Flow Confidentiality***- the protection of the information that might be derived from observation of traffic flows.

D. Data Integrity- is the assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay). As with confidentiality, integrity can apply to a stream of messages, a single message, or selected fields within a message. Again, the most useful and straightforward approach is total stream protection. A connection-oriented integrity service, one that deals with a stream of messages, assures that messages are received as sent with no duplication, insertion, modification, reordering, or replays. The destruction of data is also covered under this service. Thus, the connection-oriented integrity service addresses both message stream modification and denial of service. On the other hand, a connectionless integrity service, one that deals with individual messages without regard to any larger context, generally provides protection against message modification only. We can make a distinction between service with and without recovery. Because the integrity service relates to active attacks, we are concerned with detection rather than prevention. If a violation of integrity is detected, then the service may simply report this violation, and some other portion of software or human intervention is required to recover from the violation. Alternatively, there are mechanisms available to recover from the loss of integrity of data, as we will review subsequently. The incorporation of automated recovery mechanisms is, in general, the more attractive alternative.

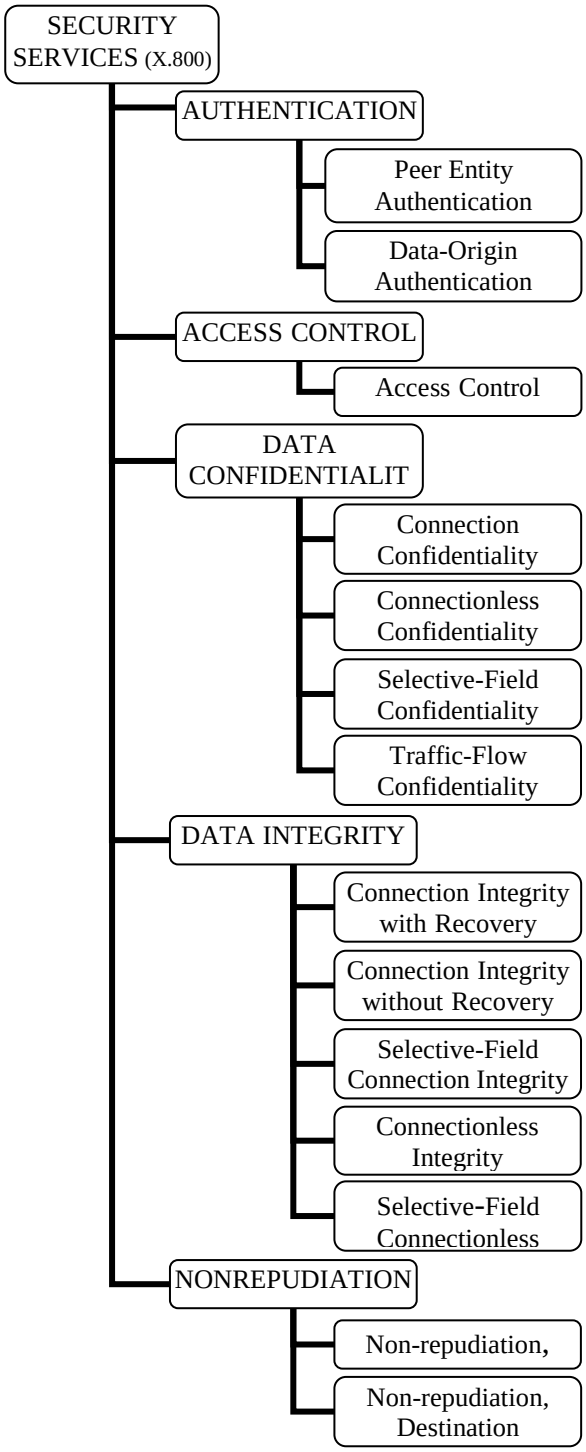

Fig. 1 Security Services (X.800)

1. ***Connection Integrity with Recovery*** - provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.
2. ***Connection Integrity without Recovery*** - as above, but provides only detection without recovery.
3. ***Selective-Field Connection Integrity*** - provides for the integrity of selected fields within the user data of a data

block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.

4. ***Connectionless Integrity*** - provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.
5. ***Selective-Field Connectionless Integrity*** - provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.

E. Non-Repudiation – is an ability to prove to a third party that an entity performed an action. Non-Repudiation provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication. Non-repudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message.

1. ***Non-repudiation, Origin*** – is a proof that the message was sent by the specified party.
2. ***Non-repudiation, Destination*** – is a proof that the message was received by the specified party.

Approach is to qualify of functionality through a service and to provide the needed functionality by that service. Therefore, we discussed respectively the following services: distributed authentication, distributed access control, distributed monitoring /auditing, and secure communications between different nodes. This services-oriented approach provides more flexibility (as services evolve, a service can be replaced or enhanced with new capabilities), scalability (since several instances of the same service could run on the same node), and fault tolerance (as high availability techniques can be used to provide service availability). [5]

III. SECURITY MECHANISMS

Security mechanism is- a process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack. [7]
A wide variety of security schemes can be invented to counter malicious attacks. The mechanisms are divided into those that are implemented in a specific protocol layer, such as TCP or an application-layer protocol, and those that are not specific to any particular protocol layer or security service. X.800 distinguishes between reversible Encryption mechanisms and irreversible Encryption mechanisms. A reversible Encryption mechanism is simply an encryption algorithm that allows data to be encrypted and subsequently decrypted. Irreversible Encryption mechanisms include hash algorithms and message authentication codes, which are used in digital signature and message authentication applications.
In Fig. 2 lists the security mechanisms defined in X.800. [7]

A. Specific Security Mechanisms

Specific security mechanisms may be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

Encryption – is the process of encoding a message in such a way as to hide its contents. Modern cryptography includes several secure algorithms for encrypting and decrypting messages. They are all based on the use of secrets called keys. A cryptographic key is a parameter used in an encryption algorithm in such a way that the encryption cannot be reversed without knowledge of the key. [3]

There are two main classes of encryption algorithm in general use. The first uses shared secret keys – the sender and the recipient must share a knowledge of the key and it must not be revealed to anyone else. The second class of encryption algorithms uses public/private key pairs. Here the sender of a message uses a public key – one that has already been published by the recipient – to encrypt the message. The recipient uses corresponding a private key to decrypt the message. Although many principals may examine the public key, only the recipient can decrypt the message, because they have the private key. Both classes of encryption algorithm are extremely useful and are used widely in the construction of secure distributed systems.

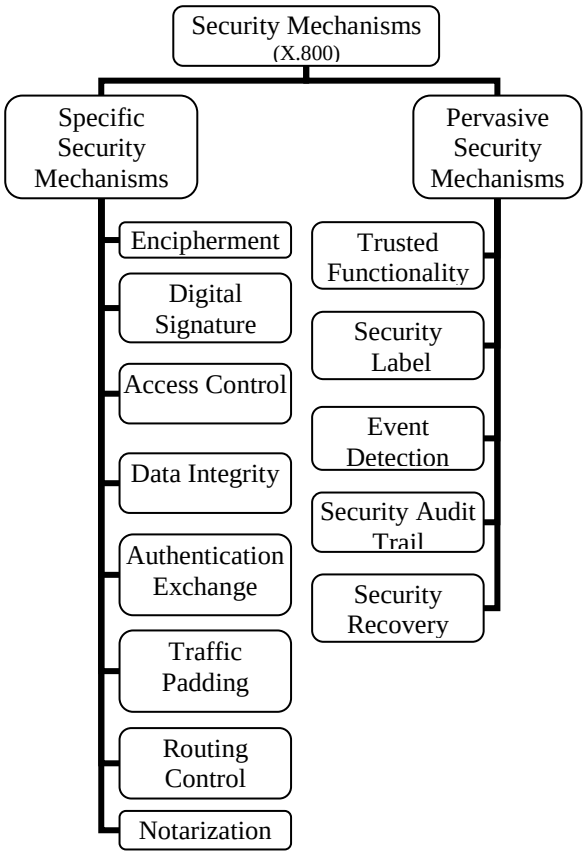

Figure 2: Security Mechanisms

Digital Signature – is the data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient). Digital signature techniques are based upon an irreversible binding to the message or document of a secret known only to the signer. This can be achieved by encrypting the message – or better, a compressed form of the message called a digest – using a key that is known only to the signer.

Access Control – is a variety of mechanisms that enforce access rights to resources. In the context of network security, access control is the ability to limit and control the access to host systems and applications via communications links and the prevention of unauthorized use of a resource. To achieve this, each entity trying to gain access must first be identified,

or authenticated, so that access rights can be tailored to the individual.

Data Integrity – is a variety of mechanisms used to assure the integrity of a data unit or stream of data units. Integrity of information means that information should be complete and accurate it should not be modified without permission if it's legitimate user. [6]

Authentication Exchange – is a mechanism intended to ensure the identity of an entity by means of information exchange.

Cryptography is used in support of mechanisms for authenticating communication between pairs of principals. A principal who decrypts a message successfully using a particular key can assume that the message is authentic if it contains a correct checksum or (if the block-chaining mode of encryption, is used) some other expected value. They can infer that the sender of the message possessed the corresponding encryption key and hence deduce the identity of the sender if the key is known only to two parties. Thus if keys are held in private, a successful decryption authenticates the decrypted message as coming from a particular sender. [3]

Traffic Padding – is the insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.

Routing Control – enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.

Notarization – is the use of a trusted third party to assure certain properties of a data exchange.

B. Pervasive Security Mechanisms

Pervasive security is provided by a combination of physical interfaces and network interfaces to a service to a user includes establishing by the user's client device network connectivity to the service, transmitting by the service an identifier to the user's client device, determining by the service whether the user enters the identifier into the service in physical proximity to the service, and invoking the service once the user has entered the identifier into the service while in physical proximity to the service. The service can provide indication that the service has been granted by sending a control page to the user's client device.
Mechanisms that, are not specific to any particular OSI security service or protocol layer.

Trusted Functionality – that which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).

Security Label – is the marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.

Event Detection – *is a* detection of security-relevant events.

Security Audit Trail – is a data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.

Security Recovery – deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.[7]

Table I, based on one in X.800, indicates the relationship between security services and security mechanisms.

Table I: Relationship Between Security Services And Mechanisms

Security Mechanisms // Security Services	Encryption	Digital Signature	Access Control	Data Integrity	Authentication Exchange	Traffic Padding	Routing Control	Notarization
Peer Entity Authentication	Y	Y			Y			
Data Origin Authentication	Y	Y						
Access Control			Y					
Confidentiality	Y						Y	
Traffic Flow Confidentiality	Y					Y	Y	
Data Integrity	Y	Y		Y				
Non-repudiation		Y		Y				Y
Availability				Y	Y			

C. Security Mechanisms in Distributed Systems

The main difficulty in designing secure distributed systems is not so much caused by security mechanisms, but by deciding on how those mechanisms are to be used to enforce a security policy. An important issue in designing secure systems is to decide at which level security mechanisms should be placed. A level in this context is related to the logical organization of a system into a number of layers. For example, computer networks are often organized into layers following some reference model. The organization of distributed systems consists of separate layers for applications, middleware, operating system services, and the operating system kernel. Combining the layered organization of computer networks and distributed systems, leads roughly to what is shown in Fig. 3. [1]

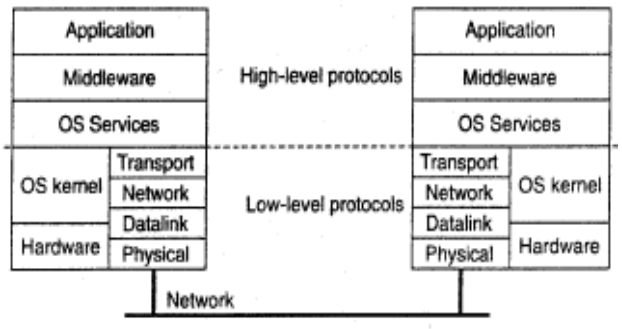

Figure 3: The logical organization of a distributed system into several layers.

In essence, Fig. 3 separates general-purpose services from communication services. This separation is important for understanding the layering of security in distributed systems

and, in particular, the notion of trust. The difference between trust and security is important. In which layer security mechanisms are placed depends on the trust a client has in how secure the services are in a particular layer. In distributed systems, security mechanisms are often placed in the middleware layer. Middleware-based distributed systems thus require trust in the existing local operating systems they depend on. If such trust does not exist, then part of the functionality of the local operating systems may need to be incorporated into the distributed system itself. Consider a microkernel operating system, in which most operating-system services run as normal user processes. In this case, the file system, for instance, can be entirely replaced by one tailored to the specific needs of a distributed system, including its various security measures.

In contrast with traditional centralized systems, modern distributed systems are well suited to the provision of physical separation. They necessarily comprise a number of physically separated components, each with the potential for dedication to a single security level or a single function. To achieve security, then, we must provide trustworthy reference monitors to control communications between the distributed components and to perform other security critical operations. The real challenge is to find ways of structuring the system so that the separation provided by physical distribution is fully exploited to simplify the mechanisms of security enforcement without destroying the coherence of the overall system. [8]

CONCLUSION

Security is not as simple as it might first appear. The requirements seem to be straightforward; indeed, most of the major requirements for security services can be given self-explanatory, as authentication or integrity. But the mechanisms used to meet those requirements can be quite complex, and understanding them may involve rather subtle reasoning. In developing a particular security mechanism or algorithm, one must always consider potential attacks on those security features. In many cases, successful attacks are designed by looking at the problem in a completely different way, therefore exploiting an unexpected weakness in the mechanism is necessary.

References

[1] Andrew S. Tanenbaum, Maarten Van Steen, *"Distributed systems Principles and Paradigms"*, 2nd ed., Upper Saddle River, NJ, USA: Pearson Higher Education, 2007, pp. 377-389.

[2] Charles P. Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, *"Security in computing"*, 5th ed., USA, Pearson Education, Inc., Massachusetts, January 2015, pp. 300-304.

[3] George Coulouris, Jean Dollimore and Tim Kindberg, *"Distributed System- Concepts and design"*, 5th ed., USA, Addison- Wesley, Massachusetts, 2012, pp. 465-473.

[4] Istam Shadmanov, Kamola Shadmanova, *"Summarization of Various Security Aspects and Attacks in Distributed Systems: A Review"*, ACSIJ Advances in Computer Science: an International Journal, Vol. 5, Issue 1, No.19 , January 2016.

[5] Makan Pourzandi, David Gordon, William Yurcik, Gregory A. Koenig, *"Clusters and Security: Distributed Security for Distributed Systems"*, Conference paper: "Cluster Computing and the Grid", IEEE International Symposium, Vol. 1, 2005.

[6] Manoj Kumar, Nikhil Agrawal, *"Analysis of Different Security Issues and Attacks in Distributed System A-Review"*, International Journal of Advanced Research in Computer Science and Software Engineering, Vol.,3, Issue 4, April 2013.

[7] William Stallings, *"Cryptography And Network Security Principles And Practice"*, 5th ed., NY, Pearson Education, Inc., 2011, pp. 809-812.

[8] Randell Brian, Rushby John , *"Distributed Secure Systems: Then and Now"*, Computer Security Applications Conference, Publisher:IEEE, Miami Beach, FL., 10-14 Dec. 2007.

AUTHORS PROFILE

First author - Istam Shadmanov Uktamovich

1. From 2013 y., - up to now - Teacher of the Department of Information Technology of Bukhara State University;
2. 2011-2013 y., Master`s Diploma on Applied Mathematics and Information Technology;
3. 2007-2011 y., Bachelor`s Diploma on Applied Mathematics and Informatics;
4. Number of papers- 11.

Second author- Kamola Shadmanova Umed qizi

1. From 2012 y., - up to now – Student of the Department of Information Technology of Bukhara State University;
2. Number of papers- 4.