Preety al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 1, Issue 1, June 2014, pp. 88-91

Mechanisms and security architecture of agent based mobile system and its security services

Preety[1], Deepika Sharma[2]

[1]PHD Scholar of DR.K.N Modi University ,Guest faculty in GNIMIT College
[2]Computer teacher in Modern School, New Delhi

Abstract: In this paper we specify a set of security services and architecture of agent which is based upon mobile system. The security services are mainly divide into three main classes which is named as security services for protection of execution platform, protecting agent using security services ,communication for security services. We also describe functionality of security and where functionality fits into architecture. For architecture we used different level of abstraction starting with highest level to lowest level. First level of abstraction involves with involved parties and their roles. Second level of abstraction involves device which is hardware component that uses wireless networking. It also include mobile terminals and non-mobile computers which are part of agent system. Third level of abstraction include agent execution environment. Fourth level of abstraction includes agent and how agent interact with its environment.

Keywords: Mobile agent, security

1. INTRODUCTION

Firstly we describe architecture of security based mobile system.

Involved parties

Involved parties are the highest level of architecture of security model. Parties behave as different individuals or organizations and one organization take role more than one entity but every party are not involved in particular scenario. We can divide party into eight part which are describe as

Device user

Device is control physically by the user but not necessarily same entity as device owner.

Device provider

In order to upgrades mobiles and add some new feature the devices provider will share a security context with device.
The security context involves provision of 'root' public keys.

Device owner

For example There are trust of devices and possible they related with other device of trust with cryptographic relationship. Device which uses to protect from various threats like malicious code.

Service Provider (SP)

It provides services like transport services, information services, payment services , directory services also including remote agent execution environment .Contract with client may or may not pre-established to service provider.

Home Service Provider (HSP)

A device owner can have many possible Home Service Provider. It will give bill for all services used by device owner and also extract payments. HSP is an entity which have contractual relationship with device user.

Trust Service Provider (TSP)

TSP behave like third party in trust which provide services e.g (Certification Authority, Registration Authority) a timestamping services , an electronic notary etc.

Agent Provider

Agent provider develop new agents for need of others agent. Agent provider and agent owner both are different entities. Agent are provided by agent provider having some specific objectives.

Agent owner

All agents are execute on a device under the control of Agent owner. Agent owner act on its behalf deploy agent to all parties.

© 2014 IJRRA All Rights Reserved

II. DEVICE STRUCTURE

We describe those parts of a device, including importantly agents and the agent execution environment. The device behaves as mobile one, but a similar structure may be behave as to exist within other devices in the infrastructure in which agents are executed. The given structure include element of device which are agent, agent execution environment, subscription module, remote resources, non – agent s/w and device resources. We consider as Security functionality here, and a complete execution environment would be more complex. . It should be noted that, depending on the device on which the agent execution environment is residing, not all the elements of this model may exist. A device might, for example, not support the downloading of agents – in which case the agent mobility service would not exist. The complete agent execution environment will include the following elements.

Agent management and control:

It governs the security platform. This element is responsible for managing all agents executing on the platform including monitoring and controlling access to resources as well as communication between agents executing on the local platform.

Agent communications service:

It provides communications facilities to agents executing within the environment. This includes secure communication services.

Agent security service:

It includes security services provided by the environment to executing
agents. For example, the environment may add a digital signature to data (signed with the private device signature key) at the request of an agent.

Agent mobility service:

It enables agents to send themselves (and associated stored state) to other devices. The service also includes functionality to assess received agents and any associated security information to decide if an agent shall begranted permission to execute on the platform. Agents requesting transfer to another platform will also be assessed for appropriate privileges here.

Event logging service:

It logs security relevant events for storage in an audit trail. It may also provide security intrusion detection based on processing of recorded events.
Agent execution environment architecture includes following services agent management & control ,agent mobility service ,event logging service ,agent security services ,agent communication services , agent execution area , access control database ,security policy ,Storage and post processing , device resources and subscription module ,other agent execution environments , TSP , remote resources
CASA (Collaborative Agent System Architecture) A mobile agent system runs, sends, and receives mobile agents, and attempts to protect the executing environment (host) against mobile agents which attempt any type of misuse or malicious behavior. The mobile agent system must be deployed on all the hosts to which the mobile agent may travel .CASA serves as an infrastructure for agent conversations. Agent conversations have several goals based on the services that each agent provides. As Kremer and Norrie describe, agents can carry out point-to-point, multi-cast, or broadcast conversations within a cooperation domain (CD) . Local area coordinators (LAC), as one of their responsibilities, activate a dormant agent when requested by a remote one or run an agent that has just arrived (in the case of being a mobile agent). In addition, CASA allows for the deployment and interaction of mobile agents with other existing agents and the system itself.

III. THE SECURITY POLICY AND ACCESS CONTROL DATABASE

It regulate the behaviour of the security mechanisms. Information making up the security policy could include a rule base describing how, and when agents can be given access to the execution environment, and can interact with each other and their environment. Other examples include the specification of security related events for which log entries should be generated, and what controls should be implemented in order for an agent to start execution. The access control database contains information governing how various resources can be accessed by the various parties (this information could, for example, be in the form of an Access Control List (ACL) or a set of capabilities, or some combination of the two).

Remote systems

It can dispatch agents to the platform for execution. In the same manner, the agent execution environment can dispatch agents to execute in other environments.

Log storage and post processing

It manages and processes log data once generated.

Device resources and subscription module

It includes all kinds of resources (hardware and software) residing on the device.

Trust Service Provider (TSP)

It provides various trust services.

Remote resources

© 2014 IJRRA All Rights Reserved

These are resources residing on other platforms with which agents can communicate, including other agents.

An agent

The various agent parts are likely to have different properties that need to be addressed via appropriate security mechanisms. The following distinctions between component parts of an agent can be made. Note that this agent model is designed for the purposes of security analysis only. As a result, important agent functionality may not be covered within this model.

Core executable part:

This information is distinguished from other information to allow a user to obtain an agent from an independent party (agent provider).

Payloads:

An agent is likely to have various kinds of payloads. Payloads can consist of non executable. Data as well as executable information required by the agent to fulfill its task. Execution state, information supplied by the agent owner, and information collected at various hosts (for mobile agents), are all examples of payloads of an agent. In addition to this, an agent can obtain executable payloads to add agent functionality that is not part of the core executable part. By separating agent parts in this way integrity verification values can be created where appropriate. The use of the above distinctions becomes particular apparent for mobile agents, but is also relevant for agents that are transferred
to be executed on a platform not belonging to the agent provider. (We are here defining a mobile agent to be an agent that can move 'on its own initiative' and continue execution in the environment where it arrives.)

IV. SECURITY SERVICES

Various classes of security services can be identified in the context of the security model. We focus on one such class, namely services to protect the execution platform. However, we also briefly review services to protect the agents.

Platform protection

We now describe security functionality addressing the protection of the execution environment. Note that agent execution environments will exist in various kinds of devices and the precise functionality, including security functionality, provided by the environment will also vary.

Hence, the functionality described here may not be implemented in every device.

Logical Access Control.

The platform needs to protect itself and its hosted agents against unauthorised access. Such functionality is often implemented in existing operating systems and execution environments. It can be implemented by using the sandbox concept, where executable code (e.g. an agent) would be able to do anything within the sandbox while any actions involving resources outside the sandbox are closely regulated and monitored. With this approach, the effort necessary to ensure the correctness of code received from outside the platform can be limited. However, in order to make full use of agents they need to be able to access resources outside the sandbox. Resources outside the sandbox include resources located on the same physical device as well as the ability to communicate with other devices/hosts/agents. The execution environment has a security policy that regulates the requirements under which an
access request will be granted. At this stage of the system design process it appears possible that an access control list (ACL) in combination with a capability-based scheme may be required for the provision of access control information. While an ACL is rather static in its nature, although dynamic changes to the list can be made, a capability scheme allows a subject to provide the required information at the point of an access request. A capability scheme based on public key cryptography and a PKI will allow for the required delegation and transfer of rights between parties. The agent management and control element is the main entity within the agent environment architecture enforcing access control. However access control is also part of the functionality of mobility, event logging, agent security, and agent communication services.

Authentication of foreign code.

To provide flexibility a host needs to be able to receive, retrieve and execute agents. In fact, this applies to any downloadable code, and not only agents. In a mobile environment, with constant changes taking place, the ability to receive and execute software is likely to be very important. As mentioned above, limited access can be given to an untrusted program in such a way that its
behaviour can be regulated to prevent any potentially harmful behaviour. However, this is not enough to provide more powerful functionality. Applications will need to be given access to resources that, if misused, can result in unauthorised and potentially harmful actions. Research on 'provably secure code' has been undertaken for several years. This research aims to verify that a piece of code is secure before it begins execution. However useful this would be, this is still very much an emerging area, and it is not clear how feasible it would be to restrict agents to those which have formal proofs of security. A more pragmatic approach is to trust a particular piece of software because one decides to trust the developer/supplier of the software. This technique is used in Java as well as in MExE (2). Using this technique we need ways of verifying that a particular

© 2014 IJRRA All Rights Reserved

piece of software does originate from a particular party. This can be done through cryptographic means. When an agent arrives at the execution environment, various security checks are made by the mobility service. The following information associated with the agent can be verified and used by the mobility service in order to decide whether an agent should be granted execution rights:
Agent owner, Agent provider, Required resources, Submitting host, Agent trail.

Platform communication.

The platform will communicate with other entities in the infrastructure. For example, agents will be transferred between platforms and various trusted service providers will be contacted. Depending on the nature and sensitivity of the communication, various levels of protection are required.

Event logging.

Unlike most security features which prevent security breaches, auditing enables follow-up when something goes wrong. The main purpose of an audit trail is to store information for later examination. Examples of applications for
audit data include fraud detection, intrusion detection, and follow-up in case of failure or security breach. Audit information can also be used for real-time monitoring in order to take immediate actions in case of security violation.
The event logging service within the agent execution environment is responsible for generating audit trails. The security policy governs what is regarded as a security event to be logged. (Audit events can also be generated through the
initiative of an agent.)Once audit data is generated it needs to be stored and properly protected. Storage can be at the local platform but can also be at a trusted party or other remote site. If security of the platform is compromised it can be valuable to have transferred the audit data prior to the point of attack. This does, of course, involve network traffic, and hence is not always the best option. Once stored, audit data can be analysed. The analysis can be automatic, e.g. by looking for known patterns or anomalies, or manual. The latter would apply particularly in the case of a security breach.

Agent protection

Analogously, security functionality is needed to protect agents executing in the agent execution environment. Issues to be addressed include: physical security, agent/platform authentication, agent mobility, agent communication, nonrepudiation and event logging.

V. CONCLUSIONS

In future work within Mobile VCE Core 2 we will develop specifications for security mechanisms and protocols to provide the security services specified in this security architecture.

VI. REFERENCES

[1]. FIPA, *FIPA Abstract Architecture Specification.*Document number: XC00001J, 10/08/2001, Available online from www.fipa.org.

[2]. ETSI, *Mobile station application execution environment (MExE), Functional description*, Stage 2, 3GPP TS23.057 version 4.3. Release 4,October 2001.

[3]. Joris Claessens, Bart Preneel, and Joos Vandewalle. Secure communication for secure agent-based electronic commerce applications.

[4]. In J. Liu and Y. Ye, editors, E-Commerce Agents: Marketplace Solutions, Security

[5]. issues, and Supply and Demand, number 2033 in LNAI, pages 180{190.Springer-Verlag, Berlin, 2001.

[6]. Cloakware Corporation. Protecting Digital Content using Cloakware Code Transformation Technology, white paper, 1.2 edition, 2002.

[7]. [30] Bruno Crispo. Delegation of responsibility (position paper). In B. Christianson,

[8]. B. Crispo, W.S. Harbison, and M. Roe, editors, Security proto-cols: 6th International Workshop, Cambridge, UK, number 1550 in LNCS, pages 118{124. Springer-Verlag, Berlin, 1998.

[9]. David H. Crocker. Standard for the format of ARPA Internet text messages, RFC 822. IETF, August 1982.

[10]. Ivan Damg_ard and Maciej Koprowski. Practical threshold RSA signatures without a trusted dealer. In Birgit P_tzmann, editor, Advances in Cryptology { Eurocrypt 2001 proceedings, number 2045 in LNCS, pages 152{165. Springer-Verlag, Berlin, 2001.

[11]. Y. Desmedt. Society and group oriented cryptography. In C. Pomerance, editor, Advances in Cryptology { Crypto '87 proceedings, number 293 in LNCS, pages 120{127. Springer-Verlag, Berlin, 1988.

[12]. Y. Desmedt and A.M. Odlyzko. A chosen text attack on the rsa cryptosystem and some discrete logarithm schemes. In H.C. Williams, editor, Advances in Cryptology { Crypto '85 proceedings, number 218 in LNCS, pages 516{522. Springer-Verlag, Berlin, 1986-203

© 2014 IJRRA All Rights Reserved