Agents as Key Elements for Information Security and Privacy*

Michal Laclavík, Ladislav Hluchý
Institute of Informatics,
Slovak Academy of Sciences, Dubravska cesta 9, Bratislava, Slovakia
E-mail: laclavik.ui@savba.sk, hluchy.ui@savba.sk

Abstract

In this paper we propose how to solve information security and privacy using agents. We discus unsecured spots in recent systems. We focus mainly on protecting data against software makers. And describe our agent solution for this protection. We also describe real life insecure situations solved by agents. Part of this paper is overview of our already proposed Agent Architecture too.

1. Introduction

Human right of privacy is a part of Universal Declaration of Human Rights, but today we can see how a lot of personal information is misused for different purposes, when registering on different websites, filling in various forms or applications.
Agents can solve some of these privacy problems but none of commercial PC or wireless platform is an agent-based platform. We will try to explain recent security problems in non-agent platforms.
Security is on a very good level in today's computer or wireless platforms. However, security against software makers' impacts is not solved. Any software installed on our device can possibly take control over our data and misuse them.
Open Source or any software where code is available is a great solution for trusting software on your device side. You can check source code for any Trojan horses or security holes. Now what about the other side of the communication pipe?
As we mentioned above, agents are the best solution for solving other side software security problem. Our solution is simple. Software on the other side will come to our device as an agent, device will lock it inside and service of the agent will be provided. Naturally, not all applications can be solved using this method but most of them can. We will devote more attention to it in the proposal. Proposed architecture can be used also for non-wireless systems such as PCs, Clusters or any Internet and intranet based systems.

2. Security in Recent Systems and in Multi Agent Systems (MAS)

Security is very important issue in all the systems. Many people can see and think that security has not been solved yet in any Internet Based System. Security holes and successful hacker impacts occur very often in the Internet world because of programming mistakes or human failure. Theoretically we can say "Security has been already solved" and that is true. The problem is that it has not been proposed and implemented yet in many systems. We can divide security into several levels [7]:

- Security of communication
- Security of system against outside impacts
- Access rights
- Approving users, agents and others
- Security against inside software and other side software

Security of Communication. We can provide this using asymmetric cryptosystems. KQML [4][5] is used as communication language in our experiments and our proposal.
Security of System against Outside Impacts. Choosing a right and secure platform with installed security patches can solve most of

* This work was supported by European Project Pellucid IST-2001-34519 and by the Slovak Scientific Grant Agency within Research Project No. 2/7186/20

those problems. In addition, some access restrictions must be set up. In MAS based on Java, implementing a good security manager can solve this. [2][5]

Access Rights. A very important thing is securing against inside impacts. We need to define permissions for a certain level for people, agents etc. Also its communication with system has to be encrypted by public private key method.

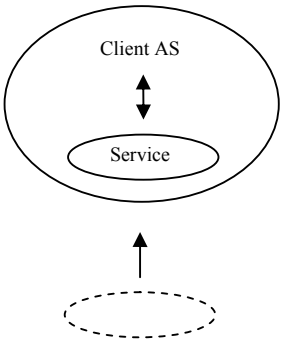
Approving Users, Agents and Others. Approving someone, who communicate is important. Even if we are securing communication using the public-private key method, we want to be sure that agent on the other side is the one, which we expect to be. Certification authorities take care of this. In our secure communication proposal, [11] central or distributed database of agent public keys (DPK) is taking this place.

Security Against Inside Software and Other Side Software

is the only unsecured spot in today's systems. This article tries to answer this problem. The illustrations and pictures can be placed optionally inside the columns or out of the columns.

3. Proposal for Agent Architecture for Wireless Devices

Our proposal is quite simple – bring service to Client Agent System as an Agent and lock it there as it is on the picture 1. This way Client can control data flow.



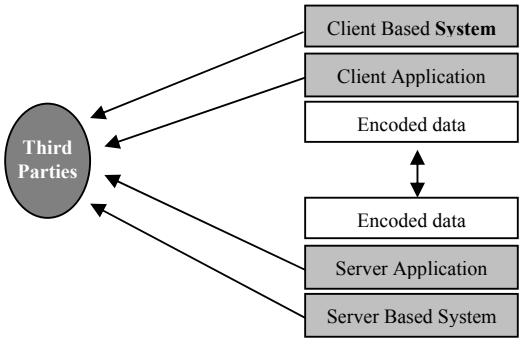
Picture 1

3.1 Overview of the problem

There are various applications of Intelligent Agents. MAS already support communication, security managers, or secure migrating. What is

not supported is protecting agents and information against system itself.

Picture 2



Let us describe those problems:[12]

Here is an example: somewhere on the Internet is a service (data + application)

We will access this application by Internet browser. Application will work on https protocol in a way all communication between user and application is secured.

As you can see on the picture 2, Client System or Applications as well as Server System and applications can send any data to the third parties over the network.

In our proposal we are trying to secure those unsecured spots.

3.2 Proposal of Architecture

Agent technology is suitable for this because service provided by a service provider can be brought as an agent to our device and act there. Also, on our side, service holder agents can secure device because only they have access to network, screen, file system, etc.

Our architecture contains four key elements.[12]

- Environment where agents acts – Multi Agent System e.g. Java Based
- Service holder agents
- Information and security agent
- Foreign service agent

Environment must be some Agent based programming environment, where code of this environment is available. Environment has its security manager and certain devices such as screen, network, file system etc. are available only to service holder agents.

In the real life, environment code should be available, signed and proved by different

software producers for any security holes. Signatures can be done similar way as signatures of active X controls or other software.

Service Holder Agents (SHA)

holds its service. They communicate with Information and Security Agent (ISA) only and if ISA passes them foreign service agent (FSA) identifier, they can communicate directly with FSA. ISA passes also allowed communication data (protocol) to SHA. Thus FSA cannot misuse SHA. SHAs can communicate between themselves but always through ISA.

Information and Security Agent (ISA)

ISA is designed to communicate with the outside world, with a user and also Foreign Service Agents (FSA). FSAs can use SHAs only through ISA.

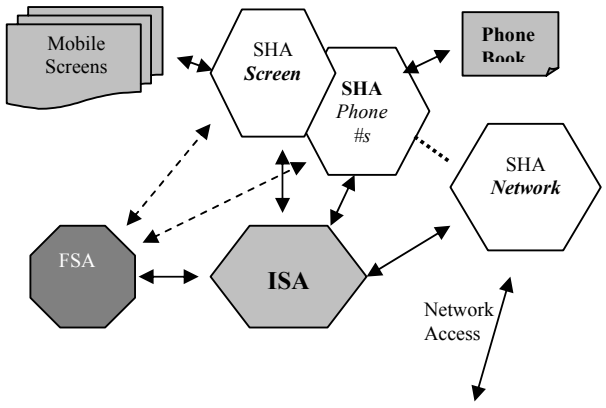
Foreign Service Agent (FSA)

FSA is service brought by network or whatever connection into environment.

FSA can be brought by different ways

- If a user wants to use some service, he/she makes request to ISA. ISA contacts FSA on some other device on the network. Then FSA comes into environment by Network Service Holder Agent and FSA can start communication with ISA.
- If FSA wants to come to environment, it connects to it and starts communicating with ISA. ISA puts it in a queue, refuses FSA or starts communication with FSA.

FSA is destroyed after all or its incoming instance can migrate into other device.



Picture 3

Partial implementation of this proposal was done in previous paper [12]

3.3 Open Problems

We can say that the proposed architecture solves privacy on a customer side but FSA can be stolen by a customer agent platform. This way service provider may worry that a customer can somehow get information from inside of FSA and misuse it for competition fight or other purposes. It is important to solve this part too.

The solution [13] is that service provider will send its FSA agent only on such agent platform, which is signed by electronic signature of different software makers and organizations. This way they can be sure that there is no Trojan horse on destination agent platform to crack their FSA agent. Environment, which controls executing and migrating of agent must be signed part of agent system only. Environment cannot migrate, clone or retrieve FSA without its permission. It only can dispose an agent.

Our privacy is often breaking when passing our personal data to different databases such as social security, police, banks, schools, medical institutions or different services. None of those organizations really need all our data such as address, social security number, birth number, phone, bank account etc. This extended information is often misused for advertising or other purposes. What they really need is to know that we really are who we are and that they can reach us somehow if they need. We have some visions how this problem can be solved using asymmetric cryptosystems and in the next chapter we are explaining it on our example applications. When our personal information is distributed, there is a less chance to misuse it and if it is misused we know right a way who is to blame. [13]

Our future work will focus on improvement and better description of such secure systems and databases.

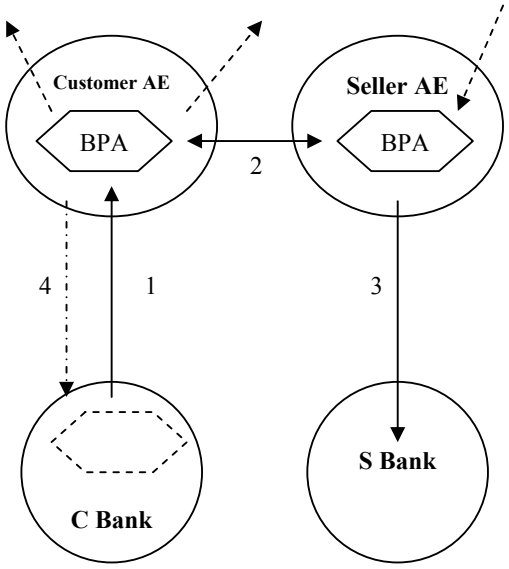
3.4 Off-line payments

We will try to explain how more complicated services such as off-line banking payments can be simply done by proposed agent architecture.
Motivation for off-line payments: Solving of electronic off-line payments is very important. Today's we can pay off-line by credit cards only. Payments by debit card can be provided only on-line what means extra cost for the bank because service have to be available 24 hours a day and also for service provider or goods seller due to on-line expenses.
Motivation for off-line payments from customer point of view as we see it can be better information privacy. When paying debit or credit card bank and seller knows who paid, how much and what for. This information can be misused for advertising then.

Proposal of off-line payments
As seen on picture 4 we can describe four steps of off-line payments:

- Step 1 - If we want to get some money from the bank we will ask bank for FSA – Bank Payment Agent (BPA) with certain amount of money. BPA will go only on such agent environment (AE) which is signed by bank or other trusted party. This way bank can be sure that BPA cannot be stolen or cloned etc. BPA migrate to Customer AE. Now customer has got certain amount of money such as electronic wallet in agent form. Connection to the bank can be closed.
- Step 2 - If customer wants to pay for something he can use BPA to connect to seller BPA and pay for certain goods or services. Customer BPA sends a message which contains an amount of money and timestamp and signs it by its signature which BPA on other side can recognize. Customer AE shows message to customer so he can be sure that no other information is passed to BPA than amount of money. Customer BPA decrease amount automatically and Seller BPA increase it.
- Step 3, 4 – If a seller or a customer wants to put money back to the bank, they can do so by sending signed amount message to Bank. If the amount of money is all money what BPA have BPA is automatically destroyed.

This solution is economically good because of off-line payments and also privacy of bank customers is not broken because they can check and approve all communication of FSA – BPA agents.



Picture 4

4. Examples Close to Reality

Sending Postcards
When we want to send a postcard by some website, we have to write our data such as name, some wishes and also an email address of a receiver. Now let us bring this application into our architecture. ISA makes request to certain FSA, which provides service of the Internet postcard. FSA migrates into Environment of our device. It communicates over ISA and SHA of display with user. A user chooses a postcard and writes text and destination of postcard. FSA creates an FSA, which holds created postcard in html for example, and this new FSA can be signed by signature SHA with user signature. New FSA is sent to its destination and original FSA is destroyed. Now it depends on destination user

if he/she accepts FSA with postcard and views it. After viewing this FSA is destroyed.
This way neither original nor postcard FSA can pass any data to the third party. On the other hand original FSA can view some commercial on user screens. This way sending of postcard means benefit for FSA provider [12].

Buying a Book
This is a bit different example but we will try to explain it by our architecture.
Some agent classes can be stored or deactivated in our device. When we wanted to buy something in the past, we looked on Internet for such agent and made sure it did only what we wanted. Now we already have buying agent in our device A. FSA (buying agent) is now not foreign but "ours". By ISA and device screen it will ask for a name of a book, a price range and delivery address from the user. We do not want to pass delivery address to device B so we will encrypt it with our private key and public key of post office and also by public key of bookstore. FSA leaves device to certain bookstore or starts to search for some stores (it can visit stores from the past for example.). FSA will find the book on device B, negotiate about price, agree or refuse it. If it agrees on the price, FSA will pass encrypted delivery address to ISA on B. This way device B can ship a book to the post office with encrypted delivery address and the post office will know where to ship it but device B cannot misuse our address. ISA on hosted B device can allow agent to send price and payment code back to device A. ISA on A device will activate payment agent and payment agent will make transaction. ISA on B device will check if payment was made. If yes, it will ship the book. If device B did not pass any data except of price to our FSA agent, this can be return back to us with additional information about founded bookstores but it can be also disposed by device B [12].

3. Conclusion and Future work

When building commercial application, security and privacy are the most important issues. We can see how a lot of personal information is misused for different purposes. Solving of those privacy and security problems is extremely important. We proposed Secure Agent Architecture and also Off-line payments and it seems to be promising technology to solve these problems. In the past we implemented main parts of our proposal and we described some possible applications for this platform. In the future we will focus on better implementation and real life experiment of our platform and also on summarizing privacy and security issues and solutions based on agents.

References

[1] Certification Authority – http://www.verisign.com
[2] Lange, D.: Programming Java Mobile Agents with Aglets. Addison-Wesley, 1998. Canada
[3] Balogh, Laclavik, Hluchy, : Model of Negotiation and Decision Support for Goods and Services, ASIS 2000
[4] KQML Website - http://www.cs.umbc.edu/kqml/
[5] FIPA: Foundation for Intelligent Physical Agents Geneva, Switzerland 1997
[6] IBM Aglets - http://www.trl.ibm.co.jp/aglets/
[7] Laclavik, M.: Negotiation and Communication in Agent Systems, 2001
[8] JKQML IBM - http://www.alphaworks.ibm.com/tech/JKQML
[9] Grasshopper - http://www.grasshopper.de/
[10] TCL - http://agent.cs.dartmouth.edu/general/agenttcl.html
[11] Laclavik, M.: Secure Inter-agent Negotiation and Communication, ICETA 2001
[12] Laclavik, Balogh, Hluchy: Secure Agent Architecture for Wireless Devices, IASTED AIA 2002, Austria
[13] Laclavik, Pavlik, Hluchy: Solving Information Privacy by Agent Architecture, ICM 2002, Czech Republic