Solving Information Privacy by Agent Architecture

Michal Laclavik*
laclavik.ui@savba.sk

Roman Pavlik*
pavlik.ui@savba.sk

Ladislav Hluchy*
hluchy.ui@savba.sk

Abstract: In this paper we propose how to solve information privacy using agent architecture. We focus mainly on security protection from the system itself as well as a different security view on distributed services and protection of data passed to those services. Our proposed architecture brings other difficulties with securing data but we try to overcome them. In addition, we describe possible applications of the architecture and securing private data in those applications.

Key Words: privacy, security, agents, architecture, wireless.

1 Introduction

Human right of privacy is 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 different forms or applications.

Also, today's technologies are giving ability to get information from a user, even he/she did not approve it, even he/she does not know about it. 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. When passing data through the Internet to some Internet application, no one can steal our data over the network but we cannot be sure what application on other site will do with the data. In other words, security of communication is all right but we have to trust software makers on each side of communication.

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 other side of the communication pipe?

As we mentioned, agents are the best solution for solving other side software security problem. Our solution is simple. Software on 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 discuss it more in

* Institute of Informatics, Slovak Academy of Sciences, Dubravska cesta 9, Bratislava, Slovakia

the proposal. Proposed architecture can be used also for non-wireless systems such as PCs, Clusters or any Internet and intranet based systems. We choose wireless platform because, according to our vision, it is easier to bring new technology into wireless world. Wireless technology for mobile phones, for example, depends on a few cell phones producers (Nokia, Ericsson, Motorola, Siemens, etc.). Moreover, bringing all changing technology into PCs is not easy and almost impossible to use. However, we will work on device independent architecture in the future. [12]

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 is not solved yet in any Internet Based System. Security holes and successful hackers 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 to 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 by using SSL what is a standard encryption method used for example for Internet Banking. Securing of communication in MAS is described in our Security proposal. 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 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. Each Agent Place or each agent who has a public key has to have this key stored in this DPK with its information. When new agent gets created, the public key is generated and sent to DPK by its creator with its information. DPK Security Agent can represent DPK. Each agent has standard method to access DPK agent by secure connection to get confirmation about public keys of other agents.

Security Against Inside Software and Other Side Software

is the only unsecured spot in today's systems. This article tries to answer this problem.

3 Proposal for Agent Architecture for Wireless Devices

3.1 Overview of the problem

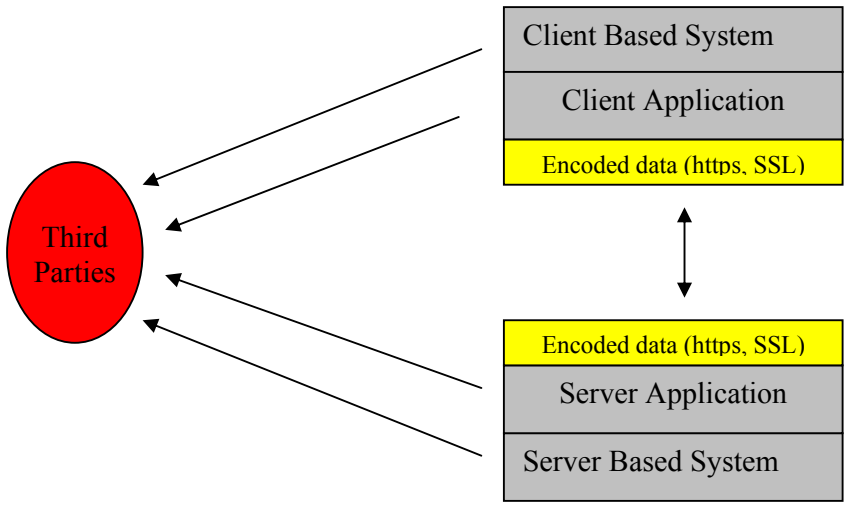
There are various applications of Intelligent Agents. MAS already supports communication, security managers, or secure migrating. What is not supported is protecting agents and information against system itself.

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 below, 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

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 has 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. A user can check this way that Environment is all right, but except of this he/she can see the source code and look for possible security holes and Trojan horse itself.

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.

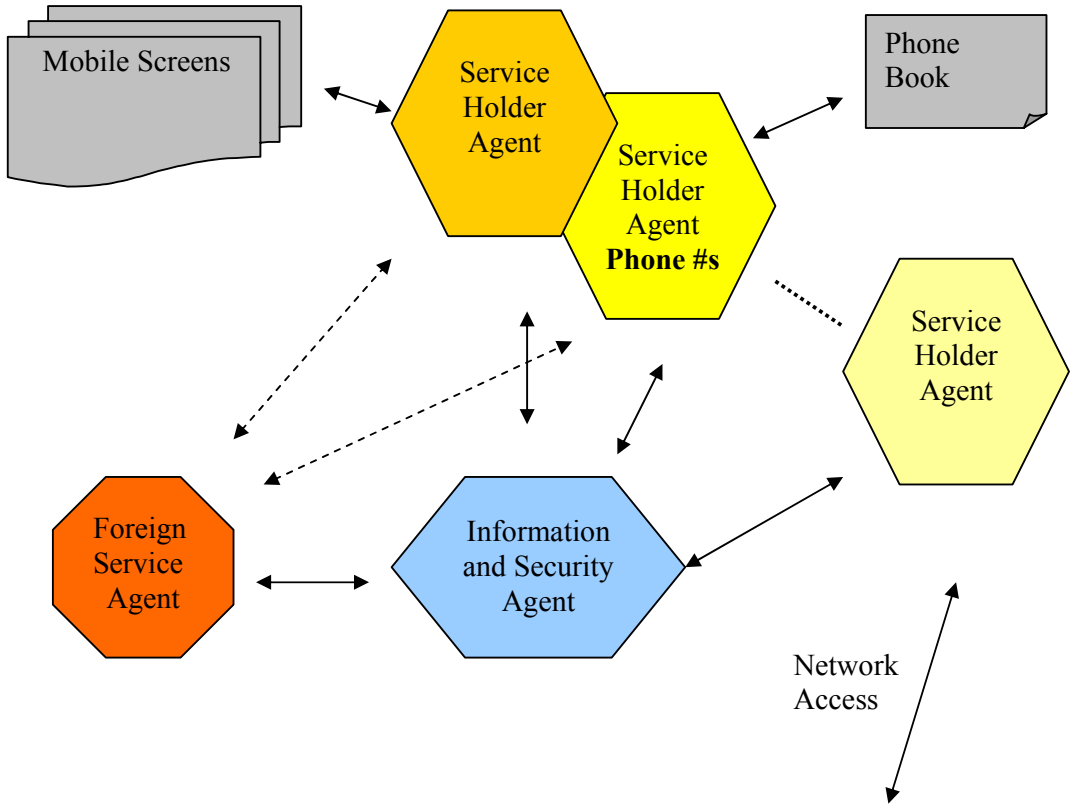
Foreign Service Agent (FSA)

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.



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

4 Open Problems and Feature work

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 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 the one to be blamed.

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

5 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 does only what we want. 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].

6 Conclusion

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 for Wireless Devices [12] because agents seem to be promising technology to solve these problems. We implemented main parts of our proposal in the past and we described some possible applications for this platform. As we already mentioned, we chose wireless platform because, according to our vision, it is easier to bring new technology into wireless world. However, we will work on device independent architecture in the future.

This work was supported by the Slovak Scientific Grant Agency within Research Project No. 2/7186/20

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