A Solution based on Cryptographic HW for Agent Protection

Antonio Muñoz & Antonio Maña

E.T.S.I Informatica, Universidad de Málaga, GISUM

October 1, 2009

Outline

Outline

1. Introduction to Agent paradigm.
2. Description of the problem.

Outline

1. Introduction to Agent paradigm.
2. Description of the problem.
3. The Trusted Computing technology.

Outline

1. Introduction to Agent paradigm.
2. Description of the problem.
3. The Trusted Computing technology.
4. Conclussions.

Introduction to Agent Paradigm

1. A mobile agent is defined as:

Introduction to Agent Paradigm

1. A mobile agent is defined as:
    1. an autonomous,

Introduction to Agent Paradigm

1. A mobile agent is defined as:
   1. an autonomous,
   2. reactive,

Introduction to Agent Paradigm

1. A mobile agent is defined as:
   1. an autonomous,
   2. reactive,
   3. goal oriented,

Introduction to Agent Paradigm

1. A mobile agent is defined as:
    1. an autonomous,
    2. reactive,
    3. goal oriented,
    4. adaptive,

Introduction to Agent Paradigm

1. A mobile agent is defined as:
    1. an autonomous,
    2. reactive,
    3. goal oriented,
    4. adaptive,
    5. persistent,

Introduction to Agent Paradigm

1. A mobile agent is defined as:
    1. an autonomous,
    2. reactive,
    3. goal oriented,
    4. adaptive,
    5. persistent,
    6. socially aware software entity.

Introduction to Agent Paradigm

1. A mobile agent is defined as:
   1. an autonomous,
   2. reactive,
   3. goal oriented,
   4. adaptive,
   5. persistent,
   6. socially aware software entity.
2. Mobile agents can actively migrate from host to host and continue its execution on the destination host.

Introduction to Agent Paradigm

1. A mobile agent is defined as:
   1. an autonomous,
   2. reactive,
   3. goal oriented,
   4. adaptive,
   5. persistent,
   6. socially aware software entity.
2. Mobile agents can actively migrate from host to host and continue its execution on the destination host.
3. Mobile agents include code, data and execution state.

Introduction to Agent Paradigm

1. A mobile agent is defined as:
   1. an autonomous,
   2. reactive,
   3. goal oriented,
   4. adaptive,
   5. persistent,
   6. socially aware software entity.
2. Mobile agents can actively migrate from host to host and continue its execution on the destination host.
3. Mobile agents include code, data and execution state.
4. Theya are not bound to the system on which they begin execution.

Introduction to Agent Paradigm

1. A mobile agent is defined as:
   1. an autonomous,
   2. reactive,
   3. goal oriented,
   4. adaptive,
   5. persistent,
   6. socially aware software entity.
2. Mobile agents can actively migrate from host to host and continue its execution on the destination host.
3. Mobile agents include code, data and execution state.
4. Theya are not bound to the system on which they begin execution.
5. They are free to travel among the hosts in the network.

Benefits of Mobile Agents(I)

1. Improving locality of reference is achieved by moving the action towards the source of data or other end point of communication.
2. Survivability: similar to nomadic tribes or migratory birds, agents can survive if moved closer to resources.
3. Analogy to the real world helps some programmers to better understand programming paradigms expressed in terms of mobile agents. Examples are travelling salesmen, shoppers and workflow management systems.
4. Customization, for example, by adjusting the search according to a user specific criteria, or by performing an action specific to a remote site.
5. Autonomicity represents agent's independence from its owner. A user can start an agent to act on his behalf and disconnect. When the user reconnects, the agent returns or otherwise provides results.

Agent Areas of Development(I)

1. Slow and unreliable links – such as radio communication, where locality of reference improves performance, and avoids potential loss while transferring large amounts of data.
2. Software distribution– is easier by associating actions and state with each distributed version and copy of a particular software.
3. Network management– useful for automating control and configuration in large scale environments, such as network.
4. Electronic commerce– by modeling travelling salesmen or shoppers visiting stores in an electronic mall.
5. Data mining– locality of reference: agents optimize a search by wandering from site to site with large volumes of information.

Problem Description

1. Agents represent an appropiate paradigm for a wide set of possible applications.
2. Security problems: Current agent platforms have low level of security (Aglets, Cougaar, JACK, JADE, JAVACT, AgentSpeak)
   - One way protection
     - Protection of Agents.
     - Protection of Agencies.

1. Sanctuaries.
2. Obfuscations techniques.
3. Watermarking.

Protection of Agencies/Hosts

1. Sandboxes.
2. Proof Carrying Code.
3. Path Histories.
4. State Appraisal.
5. Signed Code techniques.

Problems of the Current approaches

1. Do not provide a complete solution (only partial ones).

Problems of the Current approaches

1. Do not provide a complete solution (only partial ones).
2. Their integration in current agent tools is not easy (f.i JADE, JavaAct,...).

Problems of the Current approaches

1. Do not provide a complete solution (only partial ones).
2. Their integration in current agent tools is not easy (f.i JADE, JavaAct,...).
3. Do not use state of the art security.

Problems of the Current approaches

1. Do not provide a complete solution (only partial ones).
2. Their integration in current agent tools is not easy (f.i JADE, JavaAct,...).
3. Do not use state of the art security.
4. Definitely hard to apply for non security experts.

Problem Description (II)

1. We focus on solving the malicious host problem.

Problem Description (II)

1. We focus on solving the malicious host problem.
2. Our target is to develop a trusted migration proccess.

Problem Description (II)

1. We focus on solving the malicious host problem.
2. Our target is to develop a trusted migration proccess.
3. Our solution provides a 2-way protection (agent-host).

Problem Description (II)

1. We focus on solving the malicious host problem.
2. Our target is to develop a trusted migration proccess.
3. Our solution provides a 2-way protection (agent-host).
4. We base our solution on the TPM functionalities.

Problem Description (II)

1. We focus on solving the malicious host problem.
2. Our target is to develop a trusted migration proccess.
3. Our solution provides a 2-way protection (agent-host).
4. We base our solution on the TPM functionalities.
5. Shows a possible independent application of the TPM.

The Trusted Computing Technology

1. Origin:Bill Arbaugh, Dave Farber and Jonathan Smith, "A Secure and Reliable Bootstrap Architecture" IEEE Symposium on Security and Privacy (1997)

The Trusted Computing Technology

1. Origin:Bill Arbaugh, Dave Farber and Jonathan Smith, "A Secure and Reliable Bootstrap Architecture" IEEE Symposium on Security and Privacy (1997)
2. Current Status:Trusted Computing Group Specifications, Available from www.trustedcomputinggroup.org

1. A tamperproof hardware device is user to build a fully secured system bottom-up.

1. A tamperproof hardware device is user to build a fully secured system bottom-up.
2. The basic idea is to create a chain of trust between all elements in the computing system.

1. A tamperproof hardware device is user to build a fully secured system bottom-up.
2. The basic idea is to create a chain of trust between all elements in the computing system.
3. In a Trusted Computing scenario a trusted application runs exclusively on top of trusted supporting software.

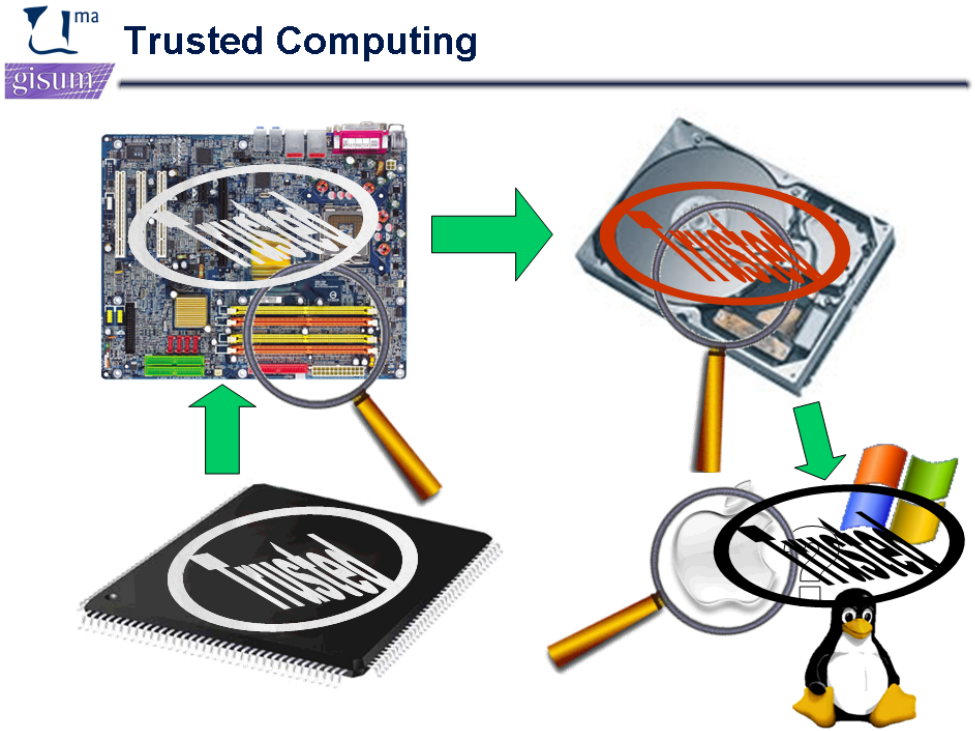

Figure: Chain of trust

Description of the Protocol

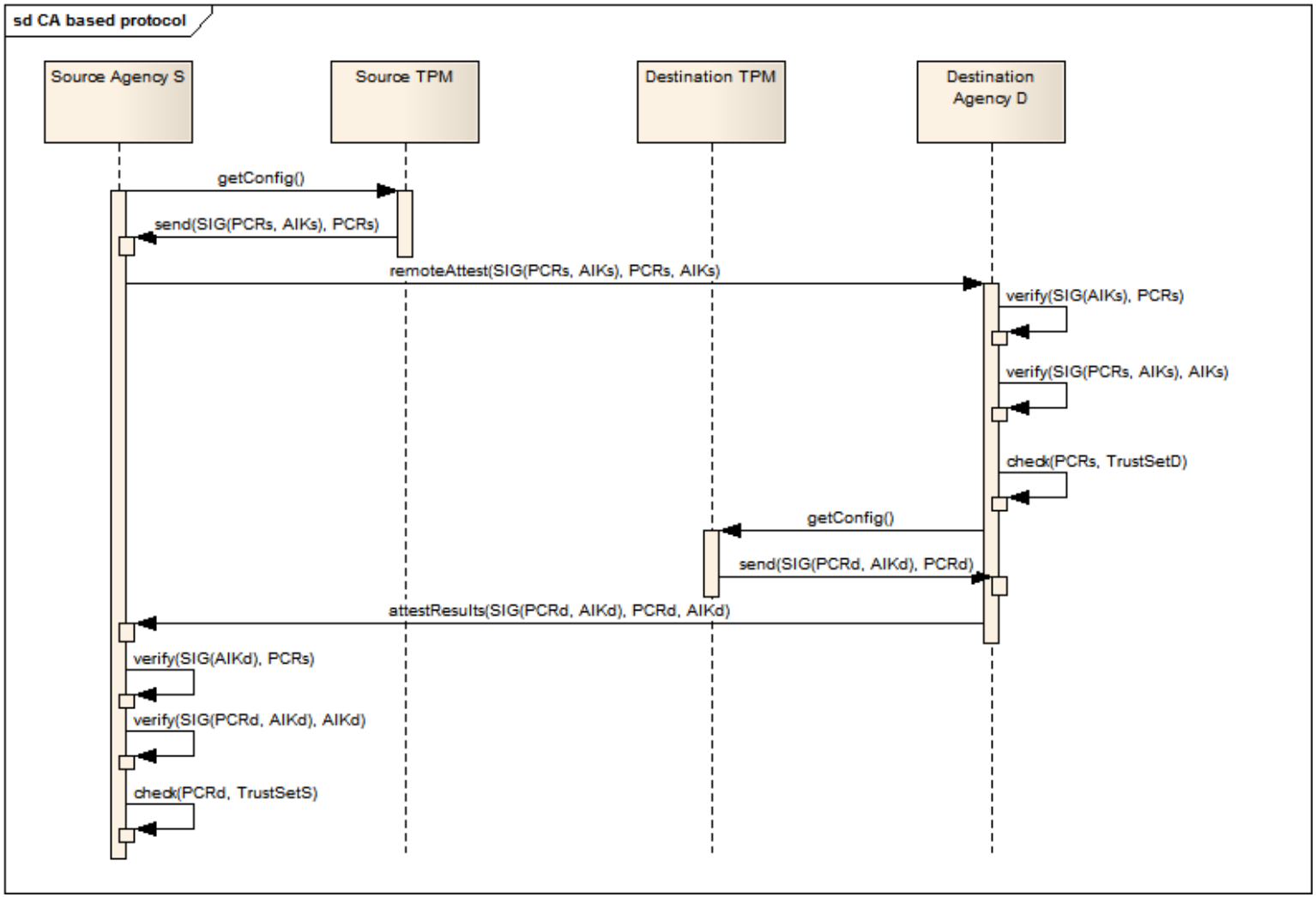

Figure: Complete description of the Secure Migration Protocol

Time-of-use time-of-check problem.

- The time-of-check-to-time-of-use problem is named after a type of software bug caused by changes in a system between the checking of a condition (such as a security credential) and the use of the results of that check.

Time-of-use time-of-check problem.

- The time-of-check-to-time-of-use problem is named after a type of software bug caused by changes in a system between the checking of a condition (such as a security credential) and the use of the results of that check.
- In our case, it refers to an attack based on modifying the destination platform once the agent has checked its trusworthiness.

Protocol using the sealed bind key functionality of TPM

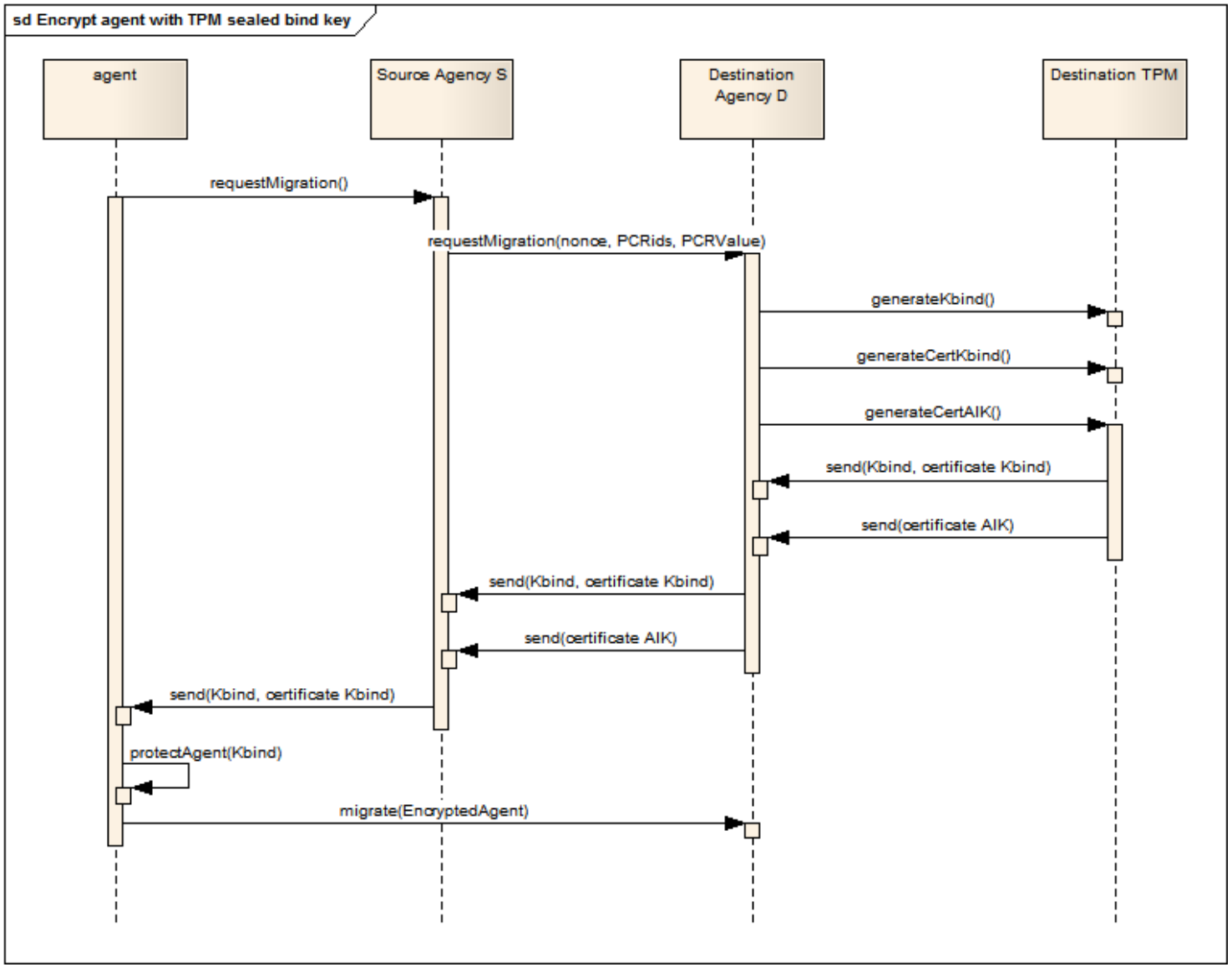

Figure: Solution with the use of a Sealed Bind key.

Main advantages

- The necessary trusted hardware is integrated in the heart of the computing system.

Main advantages

- The necessary trusted hardware is integrated in the heart of the computing system.
- Fully secure systems are possible...

Main advantages

- The necessary trusted hardware is integrated in the heart of the computing system.
- Fully secure systems are possible...
- well, ... provided everything is perfect !

Main advantages

- The necessary trusted hardware is integrated in the heart of the computing system.
- Fully secure systems are possible...
- well, ... provided everything is perfect !
- This approach provides a secure environment for agent execution through a friendly interface.

JADE

- JADE (Java Agent DEvelopment Framework) is a software Framework fully implemented in Java language.

JADE

- JADE (Java Agent DEvelopment Framework) is a software Framework fully implemented in Java language.
- It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases.

JADE

- JADE (Java Agent DEvelopment Framework) is a software Framework fully implemented in Java language.
- It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases.
- FIPA is the standards organization for agents and multi-agent systems officially accepted by the IEEE as its eleventh standards committee on 8 June 2005.

Blocks Diagram

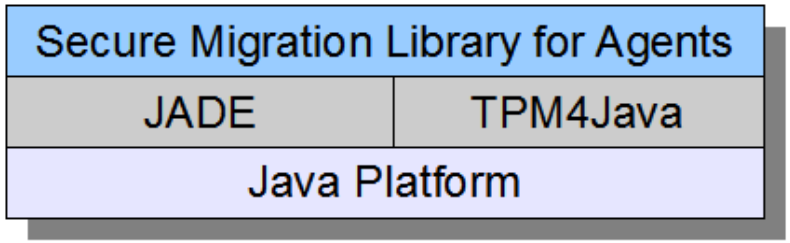

Figure: Blocks of Software of our library

Friendly use of our library

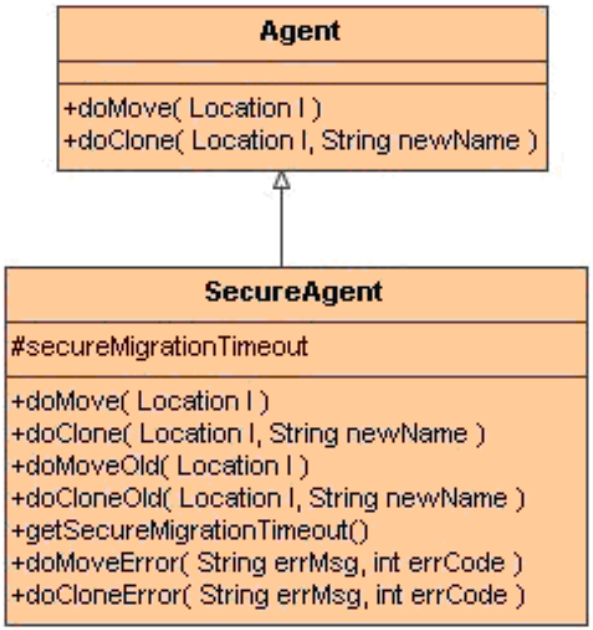

Figure: Instead of creating an Agent object, we create a SecureAgent object

Requirements of the Secure Migration Library for Agents

- Each hosting platform contains a TPM.

Requirements of the Secure Migration Library for Agents

- Each hosting platform contains a TPM.
- The state of the Trusted Agent platform is measured and the measurements stored to the TPM PCRs.

Requirements of the Secure Migration Library for Agents

- Each hosting platform contains a TPM.
- The state of the Trusted Agent platform is measured and the measurements stored to the TPM PCRs.
- The initial host platform from which the mobile agent originates is considered trusted.

Requirements of the Secure Migration Library for Agents

- Each hosting platform contains a TPM.
- The state of the Trusted Agent platform is measured and the measurements stored to the TPM PCRs.
- The initial host platform from which the mobile agent originates is considered trusted.
- Any static agent information is digitally signed by the originator.

Requirements of the Secure Migration Library for Agents

- Each hosting platform contains a TPM.
- The state of the Trusted Agent platform is measured and the measurements stored to the TPM PCRs.
- The initial host platform from which the mobile agent originates is considered trusted.
- Any static agent information is digitally signed by the originator.
- The use of PCR registers to store measurements representative of a trusted agent platform´s software state is consistent amongst all the trusted platforms.

Requirements of the Secure Migration Library for Agents

- Each hosting platform contains a TPM.
- The state of the Trusted Agent platform is measured and the measurements stored to the TPM PCRs.
- The initial host platform from which the mobile agent originates is considered trusted.
- Any static agent information is digitally signed by the originator.
- The use of PCR registers to store measurements representative of a trusted agent platform´s software state is consistent amongst all the trusted platforms.
- Every Trusted platform has enrolled at least one of their AIKs with a Privacy-CA which is know to every other trusted agent platform.

Conclusions

1. The agents have characteristics such as autonomy, reasoning, reactivity, social abilities, pro-activity, etc. which make them appropriate for developing dynamic and distributed systems based on Ambient Intelligence

Conclussions

1. The agents have characteristics such as autonomy, reasoning, reactivity, social abilities, pro-activity, etc. which make them appropriate for developing dynamic and distributed systems based on Ambient Intelligence
2. the Agents and MAS represent an interesting alternative that is well worth exploring to try to meet the challenges posed by Ambient Intelligence.

Conclussions

1. The agents have characteristics such as autonomy, reasoning, reactivity, social abilities, pro-activity, etc. which make them appropriate for developing dynamic and distributed systems based on Ambient Intelligence
2. the Agents and MAS represent an interesting alternative that is well worth exploring to try to meet the challenges posed by Ambient Intelligence.
3. Security is essential for a practical agent based system.

Conclussions

1. The agents have characteristics such as autonomy, reasoning, reactivity, social abilities, pro-activity, etc. which make them appropriate for developing dynamic and distributed systems based on Ambient Intelligence
2. the Agents and MAS represent an interesting alternative that is well worth exploring to try to meet the challenges posed by Ambient Intelligence.
3. Security is essential for a practical agent based system.
4. Security must be easy to integrate for Software developers.

Conclussions

1. The agents have characteristics such as autonomy, reasoning, reactivity, social abilities, pro-activity, etc. which make them appropriate for developing dynamic and distributed systems based on Ambient Intelligence
2. the Agents and MAS represent an interesting alternative that is well worth exploring to try to meet the challenges posed by Ambient Intelligence.
3. Security is essential for a practical agent based system.
4. Security must be easy to integrate for Software developers.
5. We plan to integrate the direct anonymous attestation in our model.

Conclusions

1. The agents have characteristics such as autonomy, reasoning, reactivity, social abilities, pro-activity, etc. which make them appropriate for developing dynamic and distributed systems based on Ambient Intelligence
2. the Agents and MAS represent an interesting alternative that is well worth exploring to try to meet the challenges posed by Ambient Intelligence.
3. Security is essential for a practical agent based system.
4. Security must be easy to integrate for Software developers.
5. We plan to integrate the direct anonymous attestation in our model.
6. We are studying the ways to overcome the rigidity of the current model by using external attestation servers.