HAL
open science

A novel trust-based authentication scheme for low-resource devices in smart environments

Anas El Husseini, Abdallah M'Hamed, Bachar El Hassan, Mounir Mokhtari

To cite this version:

Anas El Husseini, Abdallah M'Hamed, Bachar El Hassan, Mounir Mokhtari. A novel trust-based authentication scheme for low-resource devices in smart environments. ANT-2011: 2nd International Conference on Ambient Systems, Networks and Technologies, Sep 2011, Niagara Falls, Canada. pp.362-369, 10.1016/j.procs.2011.07.047 . hal-00695951

HAL Id: hal-00695951

https://hal.science/hal-00695951v1

Submitted on 10 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT)

A Novel Trust-Based Authentication Scheme for Low-Resource Devices in Smart Environments

Anas EL HUSSEINI[a,b], Abdallah M'HAMED[a], Bachar EL HASSAN[b], Mounir MOKHTARI[a]

[a] *Telecom SudParis, Handicom Lab, Evry, France*
[b] *Lebanese University, LaSTRe Laboratory, Azm Center for Scientific Research, Tripoli, Lebanon*

Abstract

In smart environments, pervasive computing contributes in improving daily life activities for dependent people by providing personalized services. Nevertheless, those environments do not guarantee a satisfactory level for protecting the user privacy and ensuring the trust between communicating entities.
In this paper, we propose a trust evaluation model based on user past and present behavior. This model is associated to a lightweight authentication key agreement protocol (EC-SAKA). The aim is to enable the communicating entities to establish a level of trust and then succeed in a mutual authentication using a scheme suitable for low-resource devices in smart environments. Finally, we tested and implemented our scheme on Android mobile phones in a smart environment dedicated for handicapped people.

Keywords: Smart environments, privacy preservation, trust evaluation, authentication, low-resource devices

1. Introduction

Trust, Anonymity and Privacy preservation are known as major factors to the acceptance and the success of pervasive computing systems. Service Platforms tend to collect and manage a large amount of personal information about individuals in order to authenticate users and/or provide personalized services, therefore threating privacy and causing a conflict between service providers and personal information owners. People dislike automatic spread of personal and identifiable data, especially when it is transferred to other parties beyond control [1-2]. The trade-off between privacy and collecting private data for authentication poses nowadays a great challenge to security designers in Smart Environments.

For disabled and aging users, smart environments are deployed to facilitate every day life activities and adapt to their needs. However, within those environments, the security and confidentiality of sensitive data has to be guaranteed. Beside that, the anonymity and privacy of the users should be protected, therefore personal information concerning users like names, addresses, financial data, and medical profile should not be allowed to flow freely without protection. Therefore, one of the foundations of the security of users in smart environments is to protect individuals privacy.

Several researchers [1-2] have admitted that smart environments are vulnerable to many security and privacy threats, and that securing pervasive computing present critical challenges at many levels [3-5]. Below, some of the challenges addressed in [6] are outlined:

1. *Privacy Issues:* Sensors and actuators distributed in space expose a great danger to user privacy since the information collected can be disclosed to intruders, malicious insiders and tracking systems.

2. *User Interaction Issues:* The access control mechanisms in pervasive environments should allow users to interact easily with devices while assuring an appropriate authentication.

3. *Security Policies:* Smart environments should have a convenient method to define and manage security policies with dynamicity and flexibility, with respect to the behavior of entities in their systems.

4. Two new security challenges, introduced by [7], are to be added: Quality of Privacy (QoP) and Trustworthy Authentication.

In this paper, we design a trust model and implement it associated to an authentication protocol. The aim is to introduce an intermediate phase that evaluates the trustworthiness of communicating entities before the phase of service provision. Our scheme will preserve and protect user privacy since it will use non-sensitive information in the evaluation process of the trust. What makes trust evaluation models really needed in smart environments is that they create a secure yet more flexible environments that what security policies alone can do. That is because the strictness of static policies may limit the freedom of normal users while it is potentially vulnerable to malicious users; on the other hand, trust systems are very adaptive to the user needs, actions and behaviors.

The remainder of this paper is organized as follows. Section 2 briefly mentions the features of the Elliptic Curve-based Simple Authentication Key Agreement (EC-SAKA) protocol, designed for low-resource mobile devices. In section 3, we describe our proposed trust model featuring its advantages. Section 4 shows the implementation of our trust based authentication scheme. Finally, section 5 concludes our work.

2. EC-SAKA Authentication Protocol

Elliptic Curve-based Secure Authenticated Key Agreement protocol (EC-SAKA) was proposed by [8] in order to suit the needs for low-resource mobile devices that face difficulties when dealing with large-sized cryptographic keys. This protocol is based on an asymmetric Diffie-Hellman scheme to generate a common secret key without exposing it to eavesdroppers. Indeed, the asymmetry in this approach is to prevent malicious attacks that try to impersonate both of the entities communicating and forward one's data to the other aiming to expose their generated key in the process. EC-SAKA scheme also prevents impersonation attacks. In addition to key agreement, this protocol provides identity verification through El Gamal Signature scheme (ECEGS). This way, it provides identity verification and common key generation at the same time.

The EC-SAKA protocol will be used in our scheme in order to authenticate entities and make them confirm each other identities. The authentication phase comes after a level of trust is already established between the entities. If the result of the trust phase is higher than a specific threshold, which is either dependent of the service or globally set by the administrator, the application will proceed to the authentication phase. Otherwise, the application won't pass to the authentication phase and the service won't be provided, because one of the communicating party appears untrustworthy to the other. The process of trust evaluation is performed using the trust model detailed in the following section.

3. Our Proposed Trust Model

Before an authentication takes place, each of the communicating parties needs to trust the others, hence the need of a trust evaluation model. Many trust models for smart environments have been proposed in the last decade, some of them presented context awareness-based security [9], while others concentrated on the quick convergence of updated trust values [10-12], addressing the different situations with similar approaches. Our proposed model has an enhanced compatibility with low-resource devices, since it uses the memory-efficient elliptic curve cryptography and a bandwidth-efficient trust evaluation scheme. Our

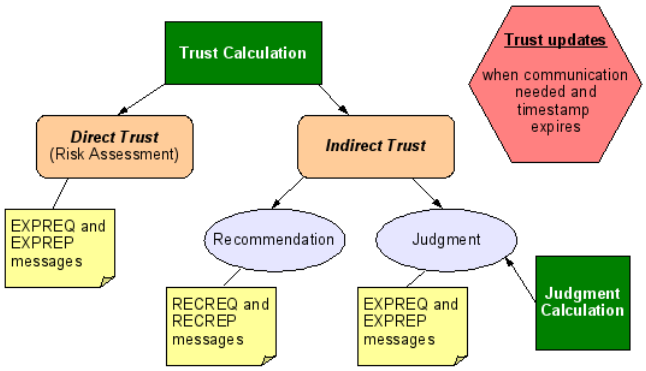

Figure 1: Our proposed Trust model with its different parts

model adopts some of their features like service-dependent trust, mentioned in [13], and moreover includes a lightweight security monitoring to prevent malicious attackers that attempt to forge trust data or alter it in any way. We made our model imitate human rational thinking in evaluating the judgment ability of recommending entities. That means whenever an entity is asked to recommend another, we take a look on how much that entity know the others, which will give a metric representing the judgment ability of this entity, before taking its recommendations as granted.

Figure 1 illustrates the architecture of our proposed trust model with its different modules, showing the steps of the trust evaluation process. Judgment is calculated based on the reports of experience messages. Indirect trust is obtained by a multiplicative relation between judgments and recommendation values given by recommendation messages. Direct trust is calculated through a risk assessment based on number of positive and negative actions of the node in question (also provided by experience messages). Finally, the net trust is a linear combination of the direct and indirect trusts. The trust updates occur only on demand or when the trust values have expired. We will explain those modules in more details in section 3. In the following, we will present the properties of our trust model and discuss the novel attributes introduced in it.

3.1. Minimization of Resource Usage

Our proposed trust model aims to serve smart environments where the hardware equipment have small processors and limited memories. Some trust models such as in [13] used a mesh-like approach in calculating trust values. That is each node of the network has always an updated trust table of all nodes in the network. That means that resources are wasted on calculating trust values that may expire before being used. Our proposed model evades this issue by evaluating the trust values only on demand and expiry.

Our proposed model tries also to use as minimal memory resources as possible. Essentially, each node in the network, independently of its nature, will need two matrices. The node will store trust values of nodes communicating with it only. Because of the memory limitation, the node will get rid of trust data related to nodes that had left the neighborhood or went dead in the network.

3.2. Trustworthiness

Trustworthiness is used in this paper to refer to the level of trust of an entity B in respect to a separate entity A. The net trustworthiness is obtained by calculation of two values: direct trust and indirect trust. Direct trust is what is commonly called “Risk Assessment”. It is used for dealing with newcomers which the entity has not yet any records of trust evaluation. In case where trust is service-dependent, we added a multiplicative factor, called the Security Action Coefficient (SAC), to the number of negative actions. This coefficient refers to the security level of a service. If the application possesses a high security level, the associated SAC should be high (e.g. 10). On the other hand, if the service didn’t need any security, such

as a weather broadcasting service, SAC can be as low as 1. Direct trust is obtained using the following equation:

$$DT = \frac{\Sigma PA_i}{\Sigma PA_i + SAC \times \Sigma NA_i} \quad . \tag{1}$$

where PA_i represents the number of positive actions done by the node in question and noticed by node i. NA_i refers to the number of negative actions, and SAC is the Security Action Coefficient related to the security level of the service.

The indirect trust, representing the recommendations of other nodes, is:

$$IT = \frac{\Sigma Tw_i \times J_i}{n} \quad . \tag{2}$$

where Tw_i and J_i are the trustworthiness and judgment values corresponding to the node i.

The value of the net trustworthiness is a combination of direct and indirect trust:

$$Tw = \alpha_{DT} \times DT + \alpha_{IT} \times IT \quad . \tag{3}$$

where α_{IT} the indirect trust coefficient is:

$$\alpha_{IT} = \frac{TS_{self}}{TS_{self} + \Sigma \frac{TS_i}{n_{recomm}}} \times \frac{\Sigma J_i}{n_{tot}} \quad . \tag{4}$$

and α_{DT} the direct trust coefficient is:

$$\alpha_{DT} = 1 - \alpha_{IT} \quad . \tag{5}$$

where TS_{self} refers to the timestamp of the trust value of the node itself, while TS_i denotes the timestamp of the trust value of the node i. n_{tot} is the total number of nodes in the subnetwork, whereas n_{recomm} is the number of nodes that responded with recommendations.

3.3. Judgment

Judgment is one of the new features introduced that aims to imitate the human behavior in a technical approach. The judgment ability is represented by the overall experience of dealing with the node in question. That experience includes both the total number of control messages exchanged and the total number of actions whether positive or negative.
The judgment related to the number of actions is equal to the total number of actions ΣA_i over the maximum number of actions $Maximum\ \ A$, as follows:

$$J_A = \frac{\Sigma A_i}{Maximum\ \ A}, \text{ if } J_A > 1 \text{ then } J_A = 1 \quad . \tag{6}$$

Similarly, the judgment related to the number of messages exchanged is:

$$J_M = \frac{\Sigma\ messages_i}{Maximum\ \ messages}, \text{ if } J_A > 1 \text{ then } J_A = 1 \quad . \tag{7}$$

At last, the overall judgment value is:

$$J = J_A \times J_M \quad . \tag{8}$$

3.4. Control Messages

In order to control the aspect of trust evaluation and share the trust data, short control messages are used for that purpose. Most of these messages are trigger-based type, except for consistency and hello messages. Those messages are:

1. *Recommendation messages:* used as request for trust recommendations. The addressed nodes will reply, if possible, with a recommendation reply containing the trustworthiness value requested.

2. *Experience messages:* used to retrieve information about statistical behavior. The reply contains information about messages exchanged and positive and negative actions, used later in the calculation of Judgment and Direct Trust.

3. *Hello messages:* periodic messages that are issued to inform the neighbors about self existence.

4. *Consistency messages:* periodic messages that aim to test the consistency behavior of a certain node in order to prevent any suspicious behavior trying to affect the trust evaluation.

5. *Knowledge Migration messages:* issued only when a node is about to pass out or leave the network. The message is a notification of the availability of trust data that is going to be lost. Interested nodes will respond by asking for recommendations and experience data.

4. Simulation and Discussion

The object of the judgment value is to increase the accuracy of the trust calculation. Not only that, the judgment value helps making the net trust evaluation converge quickly. Unlike many trust evaluation techniques where the trust metrics oscillate before reaching a stable value, we made a simulation that demonstrates how the net trust in our model instantaneously reflects the variation in the trust metrics.

Figure 2 shows the graphs of two simulations: the left one represents the variations of net trust and judgment when the positive actions are increasing, while the right showings the trust variations when the negative actions are increasing. The judgment value increase linearly with the number of actions, positive or negative. That's because the judgment represents the experience which is directly related to the number of actions. On the other hand, the net trust increases/decreases faster when the number of actions increases. The reason is that the grown experience increases the weight of indirect trust (recommendations) and accelerates the variations of net trust. The trustworthiness in our model directly reflects the behavior of the nodes, while in other trust models it takes some time oscillating before converging to an accurate value [13].

In comparison with other trust models, our proposed model is using metrics that directly reflect the present and the past line of actions committed by an entity in the network. In addition to that, our trust model tends to decrease the overall power consumption since it triggers trust updates only on demand or expiration. Finally, the security monitoring part of the model protects the entities from potential malicious attacks.

5. Implementation

In order to validate the proposed scheme composed of an authentication module and a trust module, we have implemented it using Java language and Eclipse IDE platform. A server in our architecture is the device that provides the service to other nodes. It can be a computer, an RFID reader, or even a sensor. The server part of the implementation contains no graphical interface, since it only calculates different data and

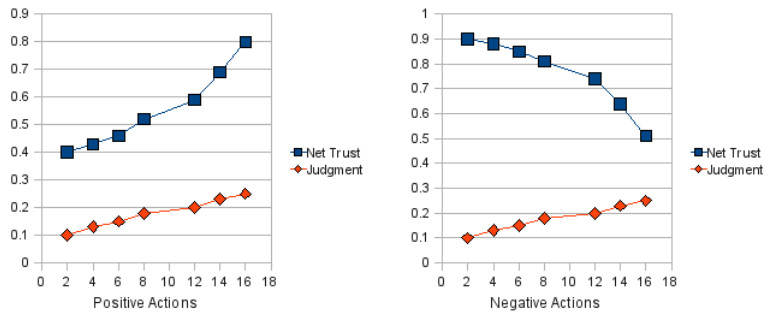

Figure 2: The variation of net trust and judgment with respect to positive and negative actions

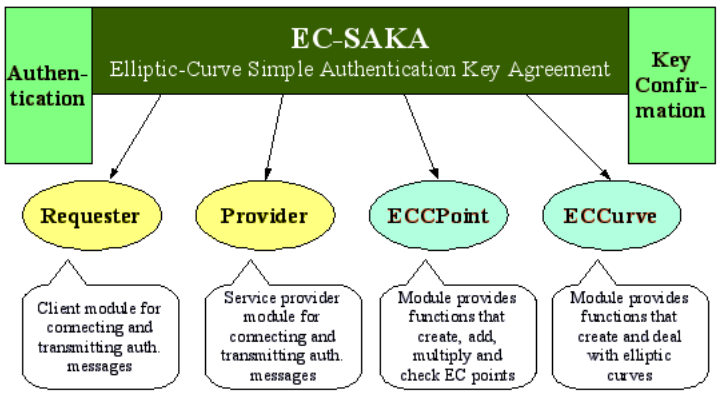

Figure 3: The different classes used in the implementation of the authentication module

communicates them to other nodes. On the other hand, the client part includes a graphical interface and can run on almost any mobile device that supports Java. We chose to test our implementation on Android mobile phones using Android SDK tools in Java. The code itself is implementable on simpler platforms than Android phones, such as active RFID tags and sensors. Android systems remain easier to work on since the implementation will only take place software-wisely.

For the authentication module, it has the responsibility to establish secret key generation and identity confirmation using the EC-SAKA protocol. The implementation of this module was done through 4 Java classes. The first two ECCPoint and ECCurve take care of all mathematical definitions and calculations related to Elliptic Curves, which are the base of the EC-SAKA protocol. The other two classes, Requester and Provider are used by the authenticating parties to exchange the messages needed for the establishment of the authentication. Since EC-SAKA uses a 3-way asymmetric scheme, the messages sent received by one authenticating party are not alike to those sent and received by the other party, thus the need of two different classes for the two parties. Figure 3 shows the class diagram of the authentication module.

The other module implemented, the trust module, uses two Java classes. The first class is for exchanging control messages. The other class uses the information provided by recommendation and experience messages to calculate direct trust, indirect trust and judgment. This class also calculates the trust weights, described in the previous section, in order to evaluate the net trust.

Figure 4: Snapshots of Android Trust client application

Our implementation of this scheme takes part in a project called "Cohabit", a smart environment project for dependent people. The project takes place in a particular residence for disabled people called ADEP. The services provided in that smart environment are daily services needed by dependent people, such as opening the door, turning on/off the light, closing the curtains, etc. The security modules we have developed evaluate the trust between the users and their environment before giving them access to use those services. Figure 3 shows several snapshots the client application on the Android phone listing the existing services spotted in the environment.

When the user of the Android phone chooses a service, he will be directed to another window that lists the security steps needed to activate that service. As it appears in Fig. 4, when clicking on 'Evaluate Trust' button, the trustworthiness value of the service provider will be calculated, as described in section 3, and the threshold value set by the administrator will be displayed. For more flexibility, the procedure was divided to several steps, to allow the user to use services on his own risk - if he wishes to - when the trustworthiness value is less than the threshold. The next button uses the EC-SAKA protocol to generate a secret shared key and enable the two nodes to verify each other's identities. If the user sees two checkmarks next to the two buttons, he can now execute the service he has chosen. Some services might not need trust evaluation and authentications, such as date & time and weather-forecast services. In this case, the corresponding buttons for trust and authentication will be disabled and only the service execution button will be enabled. Nevertheless, the actions of non-authenticated services will be recorded in the action matrices.

6. Conclusion

After emphasizing on the necessity and importance of security, privacy and trust in smart environments, we have demonstrated the effectiveness of Elliptic Curve Cryptography as new candidate in publickey cryptosystems. We have adopted a lightweight authentication protocol called Elliptic Curve Secure Authenticated Key Agreement (EC-SAKA) protocol in our scheme. That authentication protocol uses a 3-pass scheme to generate a common secret key, in addition to an elliptic curve-based digital signature.
Next, we have proposed a new trust model that respects the limitation of tiny mobile devices in terms of resources and bandwidth. Our trust model contain two new features that enhance the process of trust evaluation. The first one is a new trust metric introduced called the ability of judgment. This value tends to imitate the human rational thinking in trust and recommendation acceptance. The second feature is a lightweight security monitoring ability within the trust model to defend against security threats.

Finally, we have implemented our scheme into independent modules using Java language. Our trust based authentication scheme will be embedded on Android mobile phones to be used by dependent people in their residential areas.

References

[1] Langheinrich M.: Privacy by Design-Principles of Privacy-Aware Ubiquitous Systems. In Proceeding of the 3rd International Conference on Ubiquitous Computing (UbiComp 2001), Springer-Velag LNCS 2201. pp. 273-291.

[2] Stajano F.: Security for Ubiquitous Computing. Halsted Press, 2002.

[3] Yin Shuxin, Ray Indrakshi: A Trust Model for Pervasive Computing Environments. International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2006.

[4] Taherian Mohsen, Jalili Rasool, Amini Morteza: PTO: A Trust Ontology for Pervasive Environments. 22nd International Conference on Advanced Information Networking and Applications â Workshops. IEEE, 2008.

[5] Cheng Heng Seng, Zhang Daqing, Tan Joo Geok: Protection of Privacy in Pervasive Computing Environments. Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), 2005.

[6] Campbell R., Al-Muhtadi J.: Towards Security and Privacy for Pervasive Computing. In Proceedings: ISSS, Tokyo, Japan, 2002, pp 1-15.

[7] Tentori M., Favela J.: Supporting Quality of Privacy (QOP) in Pervasive Computing. In Proceeding of the Sixth Mexican International Conference on Computer Science. ACM, 2005, Press, pp. 58-67.

[8] Abi-Char Pierre, Mhamed A., El Hassan B.: A Fast and Secure Elliptic Curve Based Authenticated Key Agreement Protocol For Low Power Mobile Communications. The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007).

[9] Moloney Maria, Weber Stefan.: A Context-aware Trust-based Security System for Ad Hoc Networks. IEEE Conferences, 2005.

[10] Surie Ajay, Perrig Adrian, Farber David J.: Rapid Trust Establishment for Pervasive Personal Computing. Published by the IEEE Computer Society. IEEE, 2007.

[11] Lagesse Brent, Kumar Mohan, Paluska Justin Mazzola, Wright Matthew: DTT: A Distributed Trust Toolkit for Pervasive Systems. IEEE Conferences, 2009.

[12] Sheikh I. Ahamed, Moushumi Sharmin, Shameem Ahmed: A Risk-aware Trust Based Secure Resource Discovery (RTSRD) Model for Pervasive Computing. Sixth Annual IEEE International Conference on Pervasive Computing and Communications. IEEE, 2008.

[13] Ghorbel M., Mhamed A., Mokhtari M.: Secured and Trusted Service Provision in Pervasive Environment. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2009.