Trust-based authentication scheme with user rating for low-resource devices in smart environments

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

Noname manuscript No.
(will be inserted by the editor)

Trust-Based Authentication Scheme with User Rating for Low-Resource Devices in Smart Environments

Anas EL HUSSEINI · Abdallah M'HAMED · Bachar EL HASSAN · Mounir MOKHTARI

the date of receipt and acceptance should be inserted later

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. An innovation in our trust model is that it uses an accurate approach to calculate trust in different situations, and includes a human-based feature for trust feedback, which is user rating. 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. A ubiquitous computing environment entails an extensive and complex computer architecture deployment, sophisticated data control, and a judicious external interface facilitating user interaction with the system [3]. The characteristics of ubiquitous systems amplify the concern of security problems, by promoting spontaneous interactions between diverse heterogeneous entities [4]. 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

Anas EL HUSSEINI · Abdallah M'HAMED · Mounir MOKHTARI
Telecom SudParis, Handicom Lab, Evry, France

Anas EL HUSSEINI · Bachar EL HASSAN
Lebanese University, LaSTRe Lab, Azm Center for Scientific Research, Tripoli, Lebanon

present critical challenges at many levels [5][6][7]. Below, some of the challenges addressed in [8] 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 ensuring 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 [9], are to be added: Quality of Privacy (QoP) and Trustworthy Authentication. The second is combination of trust and authentication in one scheme.

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 cannot 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. In our enhanced trust model, we also integrated two new ideas. The first is Questions of Trust that aims to enhance the model immunity against malicious manipulation of trust values, as well as dealing more efficiently with situations where there are only new neighbors. The other innovation is User Rating that allows users to give feedback about the quality of service they received, which will accordingly affect how much they trust the service provider in the future.

The remainder of this paper is organized as follows. In section 2, we present several researches in literature concerning trust and repuation models. In section 3, we describe our proposed trust model featuring its advantages, pointing out the two new features: Questions of Trust and User Rating. Section 4 describes the features of the Elliptic Curve-based Simple Authentication Key Agreement (EC-SAKA) protocol, designed for low-resource mobile devices. In section 5, we show some simulations concerning the trust model metrics and discuss the results. Section 6 shows the implementation of our trust based authentication scheme. Finally, section 7 concludes our work.

2 Related Work

Several researches and studies were done regarding trust models for different purposes and in different environments. Among those, we choose several prominent ones to discuss their strong and weak points, and build our model accordingly.

In their work, Mihaela et al [10] propose a new trust model for DEs which has several innovative features. The model is based on the concept of social networks and addresses trust at different levels: user, data, service and node. The model allows fast bootstrapping of trust by importing existing trust relationships from outside DE systems and by relying on certificates issued by trusted authorities external to the DE. Furthermore, trust can be measured in a variety of contexts by using user-defined tags â folksonomy . The model abstracts from specific reputation algorithms by providing necessary interfaces for plugging-in those on one's own choice.

Yan Lindsay et al present in their paper [11] an information theoretic framework to quantitatively measure trust and model trust propagation in ad hoc networks. In the proposed framework, trust is a measure of uncertainty with its value represented by entropy. They develop four Axioms that address the basic understanding of trust and the rules for trust propagation. Based on these Axioms, they present two trust models: entropy-based model and probability-based model, which satisfy all the Axioms. Techniques of trust establishment and trust update are presented to obtain trust values from observation. The proposed trust evaluation method and trust models are employed in ad hoc networks for secure ad hoc routing and malicious node detection. A distributed scheme is designed to acquire, maintain, and update trust records associated with the behaviors of nodes forwarding packets and the behaviors of making recommendations about other nodes. Simulations show that the proposed trust evaluation system can significantly improve the network throughput as well as effectively detect malicious behaviors in ad hoc networks.

With the growing popularity of wireless mobile ad hoc networks (MANETs), many security concerns have arisen from MANETs especially in that misbehaving nodes pose a major threat during the construction of a trusted network. A reputation-based trust system can track the behavior of nodes and thereby proceed by rewarding well-behaving nodes and punishing misbehaving ones. However, existing techniques are usually either energy-consuming or complicated since the relevant reputation information is propagated throughout the network. In their paper, Yonglin et al propose [12] a novel trust computation and management system, called TOMS, which not only establishes the new concepts of trust and community but also includes both the trust computation model and trust management mechanism.

In order to obtain more suitable trust evaluation in MANETs, Junhai et al [13] suggest that the measurement and computation of trust to secure interactions between mobile nodes is crucial for the development of trust mechanisms. The calculation and measurement of trust in unsupervised ad-hoc environment involves complex aspects such as credibility rating for opinions delivered by a node, the honesty of recommendations provided by a mobile node, or the assessment of past experiences with the node one wishes to interact with. The deployment of suitable algorithms and models imitating fuzzy logic can help to solve these problems. In this paper, RFSTrust, a trust model based on fuzzy recommendation similarity, is proposed to quantify and to evaluate the trustworthiness of nodes, which includes five types of fuzzy trust recommendation relation ships based on the fuzzy relation theory and a mathematical description for MANETs. Fuzzy logic provides a natural framework to deal with uncertainty and the tolerance of imprecise data inputs for the subjective tasks of trust evaluation, packet forwarding review and credibility adjustment. Theoretical analysis and experimental results show that RFSTrust is still robust under more general conditions where selfish nodes cooperate in an attempt to deliberately subvert the system, end-to-end packet delivery ratio more quickly, and decreases the average energy consumes more effectively. The effect of node rating data's sparsity can be greatly reduced and show the excellent performance on typical data set.

According to Masthoff [14], Trust is a popular and much disputed topic in various research communities. In his paper, he attempts to integrate existing knowledge on trust into a simple computational model. The model incorporates the impact of direct experiences, reputation, stereotypes, empathy and user characteristics on trust. He also presents the results of two exploratory experiments testing and improving aspects of the model.

Felix Gomez Marmol et al [15] describe in their review the different scenarios where trust evaluation is threatened by different attacks of malicious users. Examples of those attacks are attacks of individual malicious users, attacks of collective malicious users, attacks of malicious collective with camouflage, malicious spies, sybil attacks, man in the middle attacks, malicious pretrusted peers, partially malicious users, etc. The paper also explains how affecting with peer's reputation can also become a security threat, such as driving down the reputation of a good user. Finally, the paper discusses how some of the known trust models deal with those trust evaluation threats. The trust models mentioned in that section were: EigenTrust, PeerTrust, BTRM-WSN , and PowerTrust. Finally, it concludes with a table summarizing the vulnerabilities and resiliencies of those trust and reputation models against the mentioned attacks.

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 [17], while others concentrated on the quick convergence of updated trust values [18][19][20], addressing the different situations with similar approaches. In our model, we tried to work on the points that are missed or lacking in the previously mentioned trust models, such as the guarantee of the accuracy of trust values, the rapid convergence of trust values during estimation, the compatibility with low-resource devices, and the adaptivity to the needs and abilities of dependent people.

Our proposed model has the following properties:

1. an enhanced compatibility with low-resource devices, since it uses the memory-efficient elliptic curve cryptography and a bandwidth-efficient trust evaluation model;

2. useful features like service-dependent trust, mentioned in [21];

3. a lightweight security monitoring to prevent malicious attackers that attempt to forge trust data or alter it in any way;

4. imitation of 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;

5. ability to deal with situations which it has no prior experience dealing with, using Questions of Trust;

6. a user rating system that allows to take feedback from users and integrate it in the trust evaluation process.

Figure 1 illustrates the architecture of our proposed trust model with its different modules, showing the steps of the trust evaluation process. The trust updates take into consideration the feedback obtained by user ratings. 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 the following paragraphs. Next, we are going to present the properties of our trust model and discuss the novel attributes introduced in it.

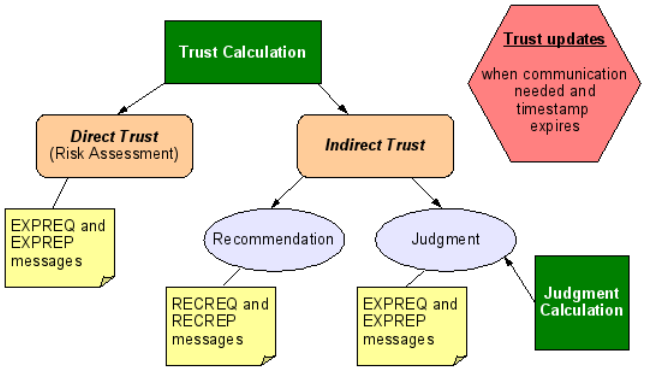

Fig. 1 Our proposed Trust model with its basic parts

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 [21] 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 a factor that recommenders have to consider in the selection of reliable peers for collaboration. Most approaches in this regard estimates trust base on global user profile similarity or history of exchanged opinions [22]. 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:

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

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

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

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.

3.5 Questions of Trust

Our basic trust model described in [23] is suitable to deal with calculating indirect trust based on recommendations of neighbors. However, it may not be efficient enough in case there are several malicious nodes giving false recommendations in order to earn the user's trust. To prevent that, the user can carry out a procedure called "Questions of Trust" before asking for recommendations. The procedure is about sending questions to the targeted nodes and comparing the answers upon receiving them, either with each other or with a set of previously known answers. The questions can be service-related. For example, if the service provided is weather-casting, the questions can be regarding the weather table at a certain geographic location at a certain period of time. Alternatively, the questions can be about the trust tables themselves; and by comparison

with trust tables of other neighbors, a contradictory or non-consistent trust values can be detected. With this, the user can be wary of nodes trying to deceive him and thus he will not ask for their services nor their recommendations.

This procedure can be repeated several times to increase the odds of detecting suspicious behaviors. The questions can be also sent directly to targets, or indirectly through other nodes as proxies for masking the identity of the original sender. Questions of Trust are also very handy if the user is entering the network for the first time or is dealing with neighbors he has no prior experience with.

This metric, like all other metrics mentioned in this paper, can generally target a large audience of users, independently of their individual capabilities. When we are targeting dependent people in particular for our system, the same mathematical model still applies, but what differs is the type of feedback taken from the users which will become an input for our model. Depending on the type of disability of the user, this feedback can be either vocal or physical (movement of fingers on touchpad or keypad).

3.6 User Rating

So far, all the indicators in our trust model are machine-based, but since trust is a concept initially borrowed from humans we decided to add a human-based indicator in our model. User ratings were used before, as in [24][25], for evaluating trust in social networks, e-shopping websites, etc. They indicate how much customers are satisfied by the quality of service provided by sellers, or how much a social network user trust another based on real-life activities. However, user rating were not used for trust models in smart environments before, although the user satisfaction/dissatisfaction after using a service from a certain provider may be quite a good indicator of the accuracy of trustworthiness calculated by machine-based indicators. In fact, the quality of service value estimated by the user rating is closely related to the trustworthiness of the service provider, since untrustworthy service providers are not expected to provide a good level of service all the time. Many advantages can be gained from adding user rating to the trust model. First, the machine-based trust models do not take into consideration the human satisfaction/dissatisfaction of a certain service, which is generally a good indicator whether the service is real or fake, and whether there is an encouraging reason for a new user to try this service with this particular service provider. Second, user ratings, unlike trust recommendations, last much more longer and therefore presents a good alternative in case of the absence of the latter. Therefore, we can consider the user rating as a human recommendation and integrates it in our formula to calculate trust. Similarly to how we defined the weights of direct and indirect trusts, the amount of participation of human recommendation in the final value of trust (also referred as Human-Machine Trust) will depend on the numbers of users that have rated the target, and how many times they used the service in question provided by the target. This can be elaborated by the following formulas:

$$\alpha_H = \frac{1}{N_R} \times \Sigma Q_i \quad \text{if } \Sigma i \leq N_R \tag{9}$$

$$\text{or } \alpha_H = \Sigma Q_i \quad \text{if } \Sigma i > N_R \tag{10}$$

$$Q_i = \frac{U_i}{U_{max}} \quad \text{but if } Q_i > 1 \text{ then } Q_i = 1 \tag{11}$$

$$\alpha_M = 1 - \alpha H \tag{12}$$

$$T_{HM} = \alpha_H \times T_H + \alpha_M \times T_M \tag{13}$$

where T_{HM} is Human-Machine Trust. T_H and T_M are respectively Human Trust and Machine Trust, and α_H and α_M are their respective weights. Σi is the total of nodes that rated the service. N_R is a constant for normalizing the number of nodes that have rated the target (typically $N_R = 10$). U_i is the number representing how many times the node i used the service from that provider, and U_{max} is the normalizing value (typically $U_{max} = 100$). Q_i is the normalized ratio of the number of utilizations of the service done by node i.

Our user rating mechanism is designed in a way that ensures the fair usage of this feature. Since we cannot expect all users to be rational in their ratings, we tried at least to integrate our own approach to fight against abuse of rating, coming from either users or service providers. In our scheme, the interacting parties are the user, the service provider, the Rating Authority and the Abuse Control Authority. The role of the Rating Authority is only to update and sign rating tables and verify the signatures. So it only stores public keys used for the verification of digital signatures,

and it does not store any rating data, as to preserve the decentralized nature of the network. Each service provider will store his own rating table(s) (one rating table per service provided), and since the table(s) is digitally signed by the Rating Authority, it will prevent any party from modifying the ratings. Another positive point about the User Rating feature is that they are permanent even if the user who did the rating leaves the network, since the service provider always carry the rating tables along; whereas trust tables depend on the presence of the node, who is going to recommend, in the neighborhood. The Abuse Control Authority is an optional entity that aims to prevent abuse coming from either users or service providers. If the service provider abstain from giving the user his rating table after service provision (i.e. preventing the user from rating him), the user can report him to the Abuse Control Authority using the service receipt he obtains at the beginning of service provision. On the other hand, if the user abuses his right of rating repetitively, he'll be detected by the rating investigation carried out by the Abuse Control Authority. The abuse control mechanism, like spam filtering, can be an automated task based on pattern recognition in abuse behaviors.

To begin with, we assume that we already have a key management scheme enabled and working in the network. The process of the user rating goes as it is shown in Figure 2. First, the user get a copy of the service provider's public key from the Rating Authority. Next, the user will request the service from the service provider. The service provider will send him a service receipt digitally signed and the service will be executed. After the service provision is complete, the service provider will take the Rating Authority's digitally signed copy of his rating table, sign it with his private key, and send the double-signed copy to the user. The user will verify the signatures, add his signature to the rating table, rate and sign his own rating, and send his rating and the rating table to the Rating Authority. The authority will verify all signatures, and if they are valid it will update the rating table based on the new user rating, and send the updated rating table back to the service provider after signing it. To defend against replay attacks, all digitally-signed exchanged data implicitly include a timestamp and/or a nonce.

Figure 3 shows two possible cases of rating abuse. In the first case the abuse is coming from the service provider, where the server abstains from giving the rating table to the user, thus blocking away the user's right of rating. As a deterrent for such situations, we allow the user to file a complaint by reporting to the Abuse

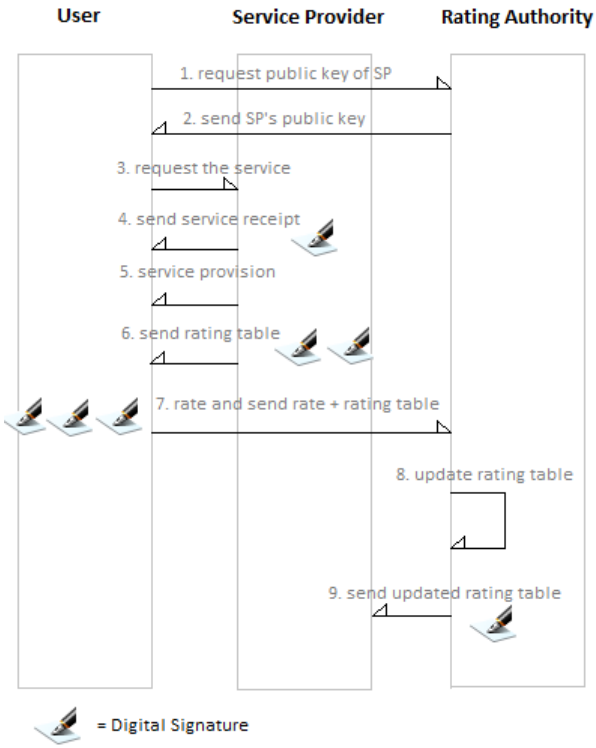

Fig. 2 Diagram showing actions performed between user, service provider and rating authority

Control Authority what happened, using his service receipt as proof of his right to rate. In the second case in the illustration, the abuse is coming from the user where he is giving untruthful ratings to increase/decrease the reputation of a certain service provider. In both abuses, the Abuse Control Authority registers the abuse either by received a justified report or by detecting using abuse pattern recognition. The algorithm describing the mechanism of the Abuse Control Authority is out of the scope of this article.

4 EC-SAKA Authentication Protocol

Elliptic Curve-based Secure Authenticated Key Agreement protocol (EC-SAKA) was proposed by [16] in order to suit the needs for low-resource mobile devices that face difficulties when dealing with large-sized cryptographic keys. This protocol benefits from the power of elliptic curve discrete logarithm problem that is even more powerful than traditional discrete logarithm problems. It 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

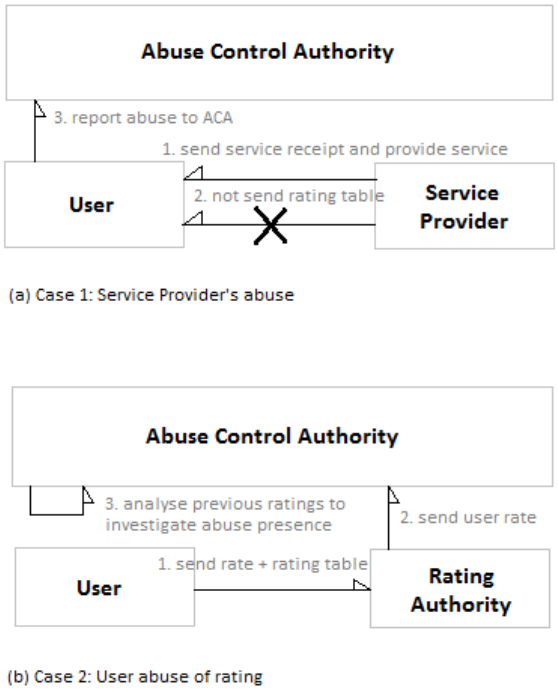

Fig. 3 Two cases involving rating abuse and Abuse Control Authority

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. Figure 4 illustrates the three passes of EC-SAKA protocol.

In Fig. 4, P and Q represent two public points on the elliptic curve chosen. n is large public number chosen by Bob and Alice, used to define the elliptic curve. a and b represent two secret values, each known only by its respective owner, Alice and Bob. A and B are the respective public keys for Alice and Bob generated from a and b. $h()$ is a hashing function, and n is sufficiently large number. K is the common secret key that Alice and Bob will be able to generate at the end of this 3-way authentication.

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

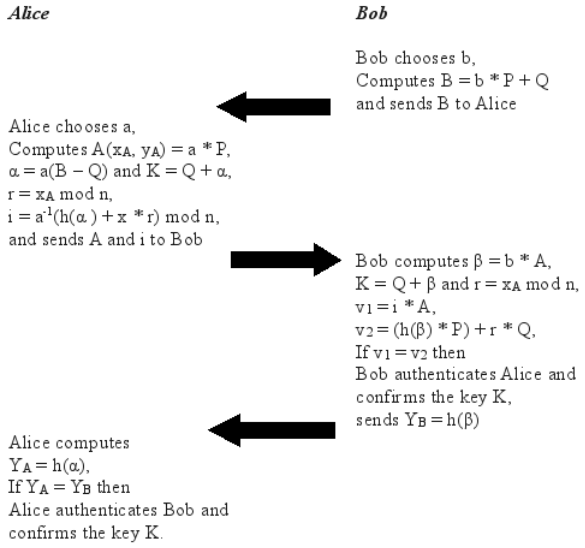

Fig. 4 The three passes of EC-SAKA protocol

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 that was already detailed in the previous section.

5 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 the trust evaluation techniques [13][17][18][19][20][21] 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.

The reason that the trust values in our trust model converges rapidly and do not oscillate like in the others is the fact that we are already depending on an always-ready judgement database. This database consists of two elements: the experience gathered by monitoring and reporting the negative actions performed by other nodes during any communication, and the user feedback through their rating which will enrich this database and makes the trust evaluation easier and faster.

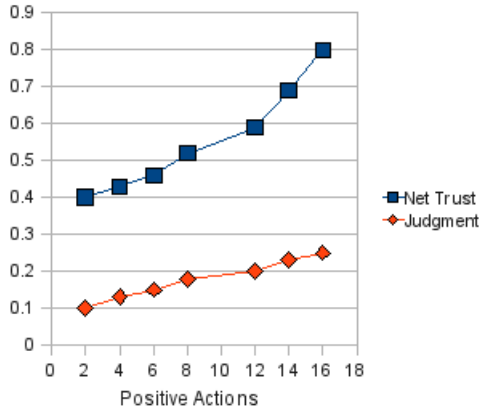

Fig. 5 The variation of net trust and judgment with respect to positive actions

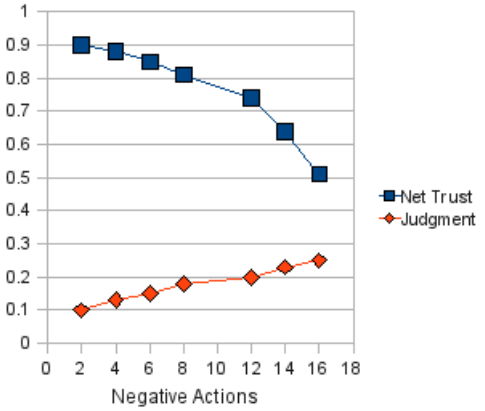

Fig. 6 The variation of net trust and judgment with respect to negative actions

Figures 5 and 6 show the graphs of two simulations: the first one represents the variations of net trust and judgment when the positive actions are increasing, while the second shows 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 [21].

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. Furthermore, the security monitoring part of the model protects the entities from potential malicious attacks. Questions of Trust also assist in preventing malicious manipulations that affect trust values. Finally, the user rating adds a human-based feature that allows human users to assist in trust decisions based on the satisfaction obtained from using services. Further simulations on real scenarios are taking place by testing the model on dependent users in a special residence in the aim of validation and confirmation of the proprieties of the model.

6 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 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 7 shows the class diagram of the authen-

tication module.

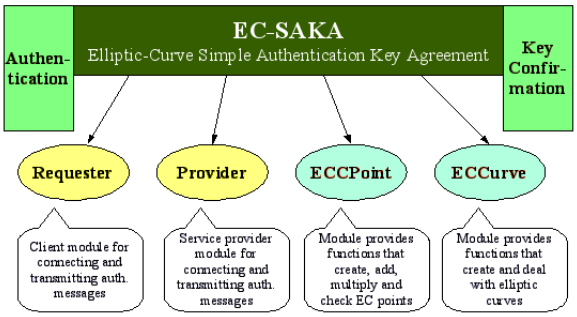

Fig. 7 The different classes used in the implementation of the authentication module

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.

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. Figures 8, 9 and 10 shows several snapshots the client application on the Android phone listing the existing services spotted in the environment.

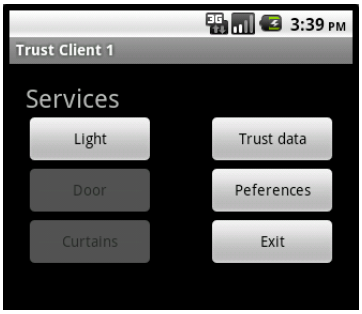

Fig. 8 Snapshot of the service provision in the Android app

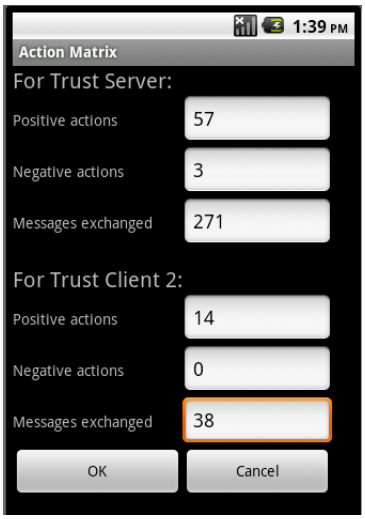

Fig. 9 Snapshot of the Trust action matrix in the Android app

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. 10, 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.

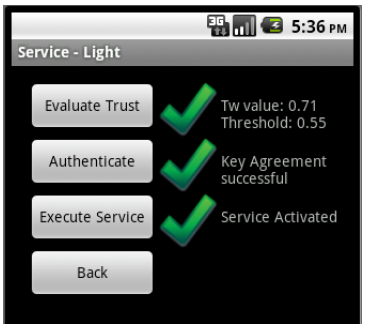

Fig. 10 Snapshot of the trust-authentication scheme in the Android app

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.

7 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.

Furthermore, we introduced two new features in our enhanced trust model. One is Questions of Trust that enable users to ask service-related questions to the service provider prior to service execution, in order to detect any fraud or fake services. The other is User Rating that gives the model a more humanistic side by integrating human feedback in the trust evaluation process.

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

References

1. Langheinrich M. (2001) 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. (2002) Security for Ubiquitous Computing. In: Halsted Press.
3. Martin Modahl, Bikash Agarwalla, T. Scott Saponas, Gregory Abowd and Umakishore Ramachandran (2006) UbiqStack: a taxonomy for a ubiquitous computing software stack. In: Personal and Ubiquitous Computing Journal, Volume 10, Number 1, pp 21-27.
4. Colin English, Sotirios Terzis and Paddy Nixon (2005) Towards self-protecting ubiquitous systems: monitoring trust-based interactions. In: Personal and Ubiquitous Computing Journal, Volume 10, Number 1, pp 50-54.
5. Yin Shuxin, Ray Indrakshi (2006) A Trust Model for Pervasive Computing Environments. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing.
6. Taherian Mohsen, Jalili Rasool, Amini Morteza (2008) PTO: A Trust Ontology for Pervasive Environments. In: 22nd International Conference on Advanced Information Networking and Applications - Workshops, IEEE.
7. Cheng Heng Seng, Zhang Daqing, Tan Joo Geok (2005) Protection of Privacy in Pervasive Computing Environments. In: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05).
8. Campbell R., Al-Muhtadi J. (2002) Towards Security and Privacy for Pervasive Computing. In: Proceedings of ISSS, Tokyo, Japan, 2002, pp 1-15.
9. Tentori M., Favela J. (2005) 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.
10. Mihaela Ion , Andrea Danzi , Hristo Koshutanski and Luigi Telesca. (2007) A Peer-to-Peer Multidimensional Trust Model for Digital Ecosystems. Computer Science Department, University of Malaga, Spain.
11. Yan Lindsay Sun, Wei Yu, Zhu Han, and K.J. Ray Liu. (2004) Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks.
12. Yonglin Ren and Azzedine Boukerche. (2008) Modeling and Managing the Trust for Wireless and Mobile Ad hoc Networks. In: ICC 2008 proceedings.
13. Junhai Luo, Xue Liu, Mingyu Fan. (2009) A trust model based on fuzzy recommendation for mobile ad-hoc networks. In: Computer Networks 53, pp 2396-2407.
14. Judith Masthoff. (2007) Computationally Modelling Trust: An Exploration. University of Aberdeen, Aberdeen, Scotland, UK.
15. Felix Gomez Marmol, Gregorio Martinez Pe rez. (2009) Security threats scenarios in trust and reputation models for distributed systems. Computers Security 28 (2009), pp 545-556 .
16. Abi-Char Pierre, Mhamed A., El Hassan B. (2007) A Fast and Secure Elliptic Curve Based Authenticated Key Agreement Protocol For Low Power Mobile Communications. In: The International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007).
17. Moloney Maria, Weber Stefan. (2005) A Context-aware Trust-based Security System for Ad Hoc Networks. In: Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks, 2005.
18. Surie Ajay, Perrig Adrian, Farber David J. (2007) Rapid Trust Establishment for Pervasive Personal Computing. Published by the IEEE Computer Society. In: Pervasive Computing, IEEE, Volume 6 Issue 4, pp 24-30.
19. Lagesse Brent, Kumar Mohan, Paluska Justin Mazzola, Wright Matthew (2009) DTT: A Distributed Trust Toolkit for Pervasive Systems. In: IEEE Conferences, 2009.

20. Sheikh I. Ahamed, Moushumi Sharmin, Shameem Ahmed (2008) A Risk-aware Trust Based Secure Resource Discovery (RTSRD) Model for Pervasive Computing. In: Sixth Annual IEEE International Conference on Pervasive Computing and Communications. IEEE, 2008.
21. Ghorbel M., Mhamed A., Mokhtari M. (2009) Secured and Trusted Service Provision in Pervasive Environment. In: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2009.
22. Daniela Godoy and AnalÃa Amandi (2011) Enabling topic-level trust for collaborative information sharing. In: Personal and Ubiquitous Computing Journal, OnlineFirst, 6 August 2011.
23. A. El Husseini, A. Mhamed, B. El Hassan, M. Mokhtari (2011) A Novel Trust-Based Authentication Scheme for Low-Resource Devices in Smart Environments. In: The second International Conference on Ambient Systems, Networks and Technologies (ANT-2011).
24. Borzymek, P. ; Sydow, M. ; Wierzbicki, A. (2009) Enriching Trust Prediction Model in Social Network with User Rating Similarity. In: International Conference on Computational Aspects of Social Networks, 2009. CASON '09, pp 40-47.
25. Yamasaki, S. (2011) A Trust Rating Method for Information Providers over the Social Web Service: A Pragmatic Protocol for Trust among Information Explorers and Information Providers. In: 11th International Symposium on Applications and the Internet (SAINT), 2011 IEEE/IPSJ, pp 578-582.