INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL. 12, NO. 4, DECEMBER 2007, 331-340

The KAA Framework : A History-Based Trust Establishment in Ambient Networks

Samuel Galice, Marine Minier and Stéphane Ubéda

Abstract— **We propose in this paper a global mechanism called KAA framework to handle trust in ambient networks. We assume that each device is equipped with a set of cryptographic tools in order to prove their successful past interactions with other devices. These interactions are stored in a database called History. Each entry of this History can be verified by a third party if this latter has yet interacted with both devices: this mechanism is built on Identity-Based Cryptosystems. The decision-making process relies on the following statement: when two stranger devices meet for the first time, they exchange their respective Histories. If it appears that they have a sufficient number of common trusted devices, they trust each other. In this paper, we also show how context awareness could be a strategic tool to adjust security policy via the use of an adapted trust metric. Privacy and anonymity are also discussed.**

Index Terms— **Trust Establishment, Ambient Networks, Privacy, Anonymity.**

1. INTRODUCTION

NOWADAYS, wireless communication is an important research field and the outcome of this research already started to impact real life: smart-phones, Personal Digital Assistants (PDAs) and other highly communicating devices have ubiquitous capabilities. Embedding computation into the environment would enable people to move around and interact with other devices more naturally than they currently do. Large intelligent environments, be it an office, a building, or a town, are envisioned to offer services to users. For instance, provide services such as printing, opening doors, or guiding visitors within a campus. This, however, may mean many things, and indeed authors have used these terms in significantly different ways since computation and communication facilities can be spread in many manners.

1.1. Ambient networks

Ambient networks describe the concept of integrating computation into all surrounding communicated devices, a situation in which various devices are present and felt everywhere: each device can sense changes in their environment and thus can automatically adapt and act based on these changes and on user's needs and preferences. Without fixed infrastructure, intelligent appliances like vending machines or printers are expected to offer wireless interfaces in order to provide (or request) services. Federations of appliances and devices such as PDAs are emerging, and thus pairing mechanisms, which enable the establishment of a secure link between two devices, become important. However, these communicating devices must have powerful enough capacity for supporting all the various cryptographic schemes necessary for these links. Finally, ambient networks must bring together several different research topics: context awareness (sensing and monitoring), social-organized networking, etc.

The large scale deployment of ambient networks heavily depends on the assurance of essential security properties for users. Indeed, distributed appliances impact daily interactions that often involve assets of users and service providers. This evolution entails unforeseen threats and vulnerabilities, and thus protecting users in terms of security and privacy is becoming a major concern. Strong security - as usually attempted in wired network - can not be achieved in an environment where the active compromising of devices is a real threat. Moreover, due to the lack of infrastructure, there is no permanent global connectivity that means that during some interactions it is impossible to rely on remote trusted third parties such as certification authorities or revocation lists. In addition, mostly interactions occur among unknown parties, there is thus potentially a lack of a priory trust among parties. Overcoming these issues is done in general by the development of trust-based establishment protocols which are necessary to minimise the risk [1], [2]. So, mechanisms to build trust among parties have to be designed. This approach of course does not provide security in the traditional sense of ensuring protection of resources, rather its purpose is to mitigate the risks that are present in these interactions.

1.2. Trust

According to [3], trust management systems are classified into three categories: credential and policy-based trust management, reputation-based trust management, and social network-based trust management. This approach depends on the way trust relationships between devices are established and evaluated. In credential and policy-based trust management system [4], [5], [6], a device uses credential verification to establish a trust relationship with other devices. The concept of trust management is thus limited to verifying credentials and restricting access to resources according to application-defined policies: they aim to enable access control [7]. A resource-owner provides a requesting device access to a restricted resource only if it can verify the credentials of the requesting device either directly or through a web of trust [8]. This is useful by itself only for those applications that assume implicit trust in the resource owner. Since these policy-based

CITI Laboratory - INRIA ARES project - INSA Lyon
21, avenue Jean Capelle - 69621 Villeurbanne Cedex - France
Tel. (+33) 4 72436418 - Fax. (+33) 4 72436227
E-mail: {samuel.galice, marine. minier, stephane. ubeda}@insa-lyon.fr

access control trust mechanisms do not incorporate the need of the requesting peer to establish trust in the resource-owner, they by themselves do not provide a complete generic trust management solution for all decentralized applications. Reputation-based trust management systems on the other hand provide a mechanism by which a device requesting a resource may evaluate its trust in the reliability of the resource and the device providing the resource. Trust value assigned to a trust relationship is a function of the combination of the devices global reputation and the evaluating devices perception of that device. The third kind of trust management systems, in addition, utilize social relationships between devices when computing trust and reputation values. In particular, they analyze the social network which represents the relationships existing within a community and they form conclusions about devices reputations based on different aspects of the social network. Examples of such trust management systems include Regret [9], [10] that identifies groups using the social network, and NodeRanking [11] that identifies experts using the social network.

1.3. The KAA project

The Knowledge Authentication Ambient project (KAA[1]) mimics human society in order to propose trust management mechanisms for ambient networks. Our trust policy depends of the underlying social patterns [12], [13], [14]. We also avoid trust dissemination: each entity is viewed as an autonomous device and trust assessment is based only upon direct link between trustworthy devices. These devices may have no a priori relationship for establishing trust among them. This leads naturally to a decentralized approach that can tolerate partial information through there is an inherent piece of risk for all trusting entity. We also objects the idea of a recommendation mechanism, the trust is considered as a non transitive relation. Consequently, a device could not receive a recommendation for a device that it has never met before since there is no simple and local means to prove the semantics of such a recommendation.

The key concept of our framework relies on the notion of History of past interactions: all successful interactions are represented by cryptographically provable elements and stored in the History of participant devices. Then, two unknown devices could exchange some services if they share a sufficient number of common trustworthy devices: the trust level is thus evaluated by the number of common trusted devices. Our framework includes several security protocols ensuring the robustness of the model [15], [16], [17]. Verifying the identity of each device is realized by a particular trusted station: the imprinting station. The burden of the decision-making process, which is somehow complex and inductive in general, is also significantly reduced by the use of proved data since the decision is based on provable past behaviors.

Our approach is quite closely related to one discussed in [18], but it differs from several points: in the cited paper, the author proposed a human trust management model and framework in order to facilitate the construction of trust-aware mobile systems and applications. The proposal consists in the construction of a trust management model which involves trust formation, trust dissemination and trust evolution; such a construction is based upon a portfolio containing credentials of previous agents interactions.

Even if we are designing an automatic trust management system to be embedded in smart device, we are facing both technical and social challenges. For instance, in a family-like pattern, we can reasonably suppose that only a unique recent interaction with another device is sufficient to allow further interaction. On contrary, in a very big organization, this threshold should be higher in order to authorize various services.

This paper specifically addresses context awareness and privacy in ambient networking of the KAA framework. The trust model is described together with an overview of the cryptographic protocol and its security analysis. The trust function is also analyzed through simulations.

This paper is organized as follows: section 2 describes the KAA framework. Section 3 presents the cryptographic protocol carried by each device that permits the creation and verification of the trust links. Section 4 focus on the analysis of our proposition whereas section 5 treats the privacy of the KAA framework. Finally, we conclude in section 6.

2. THE KAA FRAMEWORK

Throughout the paper, the device which provides the service is called the *trustee* whereas the requesting device is denoted the *trustor*.

2.1. Social patterns

Exchange is a central and traditional object within the social sciences, notably in economics science where *market exchange* analyzes circulation of goods and services between agents (exchange is trust-regulated, that is to say mostly unknown individuals are implicated), thus in sociology and in anthropology where the key concept is *social exchange*, which gathers all kinds of non-economics exchange between individuals. Social patterns may be distinguish themselves on two strongly differentiating variables.

In one hand, the social distance that separates two individuals: this social distance can be loose in the case of a market or an organization (this is the reason why the contract - commercial or labor - is so important to support exchange between unknowns). Or, at the opposite, this distance can be strong as often in the case of the family (included friends, neighbors, and other kind of strong social bonds and where exchange is gift-regulated) and network (as a community of individuals that share something like a life experience, an interest in something, ...) where familiarity, real or virtual, allows individuals to exchange without contracts.

On the other hand, the degree of structure of the institution defines the degree of liberty of which the actors can dispose in order to exchange (notably the choice of the partner and the nature of exchanged things). This degree can be loose, as in a network or a market where individuals have all latitude to choose themselves and to exchange what they want to,

[1] http://citi.insa-lyon.fr/kaa/

or strong as in a family or an organization/institution where exchange is more constrained by formal hierarchies and rules.

- **Family:** a community with a strong social distance and a strong degree of structure.
- **Network:** a community with a strong social distance and a loose degree of structure.
- **Market:** a community with a loose social distance and a loose degree of structure..
- **Organization:** a community with a strong social distance and a loose degree of structure, as a company.

2.2. The proposed framework

The KAA framework only permits a local reputation mechanism, locally computed, which is based upon provable elements. The interactions could happen between devices of the same community or between devices from different communities. The computation of the reputation value of the device A for the device B could be performed using the following informations computed at two levels:

- **Direct** The device A keeps the memory of past interactions between A and B, using the semantics generated from the elements of History. This information could be proved by a third party who has already met B. We suppose that any element is marked by a time stamp or the trust link is definitely acquired. An element will only indicate the quality of the sequence and the number of past interactions with a single device ('trustor proof', 'reciprocal proof' or both in case of an exchange). This quality could not be transmitted because it is not provable. All elements are cryptographically proved. So a device A that is convinced by those elements could derived a reputation value for the device B.
- **Indirect** When A interacts with a new device C, they exchange whole or parts of their Histories. If they have B as a common device, A could receive from C the semantics of the past interactions between B and C ('trustor proof', 'reciprocal proof' or the both). The device A could take into account those values to update its own reputation value for B. Of course, there is no information about the quantity of previous interactions that C has built with B. This bonus of reputation could be taken into account only once for each correspondent other than B and that presents B in its own History.

We could then notice that the trust indirect policy does not violate the fundamental principles of the KAA framework: we only use trust values that have been cryptographically proved. It is impossible to create by its own an entry in the History for a particular device never met before. Moreover, this reputation model is an incitation to be reciprocal and let a proof of good behavior in the History of previously met devices. Thus, the devices could hope to have a good reputation for a lots of devices.

History-Based Trust management has been yet studied in the past. In [19] such a trust management is presented but it is dedicated to a group signature and using trusty environment to generate elements of History. History-Based Trust management may be regarded as a consisting alternative proposal to a pairing model requiring frequently intervention of users. Pairing model is primarily relevant in the case of long term association between devices. For example, this approach is deployed in the built-in security mechanism of the Bluetooth chip. In this mechanism, the same identification information (a PIN code for example) has to be physically entered on each device. After the generation of a symmetric link key, devices are able to authenticate each other and to encrypt communication. The share PIN is the weakness of the model since it is prone to simple off-line attack [20]. In [21], a complete trust management framework is also proposed although it is only applied to a strongly closed environment where trust is a boolean value: the principle relies on removal or banishment of devices.

2.3. Trust parameters

For sake of simplicity, we consider a unique community which is characterized by its *size* n. The community dynamics depends on the *interaction rate* of each individual device, i.e. the average number of interactions by unit of time. We denote H_{max} the maximal size of a History which is considered as the same for all devices and BL_{max} the maximal size of a blacklist which is also considered as the same for all devices. Each device (for example A) stores its trusted devices in its History (H_A) and reciprocally, it stores untrustworthy devices in its blacklist (BL_A). Hence, for a trusted device B there is a corresponding element $h_A(B)$ in the History H_A of A. Moreover, each element is tagged with two fields: the first one, denoted by $I_A(B)$, represents the *intensity* of the relation with B, the second, denoted by $U_A(B)$, represents the *utility* of B, i.e. the usefulness of the device B with respect to the device A. This last notion is related to the number of times this element contributes in the computation of common elements.

In a more general framework, with different communities, our trust policy could be different according to either an interaction takes place with a member of its community or not. Thus, we have to take into account the *community parameter* $C(A, B)$.

More deeply, the internal structure of a community could also modify our trust policy: for instance, through the social degree of the community, initial trust may be total: each device is valid and active in the community (for example for objects belonging to a same family). On the contrary, the initial trust may be partial or even non-existent if the social degree of these communities is loose (for example for objects belonging to members of a national sporting federation with several thousand of members). In this case, the weight given to the *community parameter* could not be anymore the same and the behavior of a mobile in such a community depends essentially of its own experiences through its History.

To sum up, a device A evaluates the trustworthiness in a device B using only local information: its History H_A, the intensity $I_A(B)$ of its relation with B, its own blacklist BL_A and the History H_B transmitted by B. With the help of cryptographic algorithms (see section 3), device A can check the validity of any element of History in H_B as soon as it has the knowledge of the public identity of the involved devices.

As explained later, the verification is restricted to $H_A \cap H_B$ which corresponds to the known common devices between A and B. Conserving an element of History relating A and B means *device A recommends device B* but contrary to a classical recommendation framework, this assumption can be verified.

2.4. The lifespan of elements of History

In an environment where exists neither a central regulating entity nor authorizing accreditations or the revocation of objects, a fair assumption is let's make the time: the data elements are automatically revoked after their lifespans expire [22]. A temporal semantics can easily be added to an element of History if both parties agree on a creation/expiration date. This information is simply concatenated with existent data before the signature. Nevertheless, nothing guarantees that the both entities will choose correct values for this information: the reality may be different (dishonest devices or simply malfunction). But there is no real benefit to cheat on these values. Indeed, each entity may filter a received element of History according to its local trust policy: an element can be rejected if its creation date is too old, its validity period is considered to be abnormally long although being still valid or if its lifespan is of course expired. No information having an infinite lifespan in the system is guaranteed by this timestamp.

2.5. Trust policy

Another aim of the KAA framework is to build a global structure which includes many systems rather than a specific one. With the help of the two basic mechanisms - namely Common History Extraction (see section 3) and local reputation - one may build a wide range of trust policy. Let us sum up first all parameters that can be adapted. Adaptation can be static, depending of the social pattern of the user's smart device is belonging to or dynamic depending of the context of use.

Obviously, the trust establishment policies and how the system behave are to be considered at the same time while taking into account the issue of risk assessment. Indeed, according to the number of common devices and the nature of these elements in the respective Histories, the fulfillment of a service might be conditioned by a certain number of criteria. For example, it is less risky to share its bandwidth with other devices than offering a read (or write) access to confidential files. In order to have a coherent model, the various issues involving in risk assessment must be studied.

The main parameter is the size of the History that a device requires in order to accept any interaction. Note that, there exists an asymmetry in interaction and the size of the common knowledge required to be receiver or to be provider not needs to be the same. Also the fact that the corresponding device belongs or does not belong to the same community (security domain) may also have an impact on that. For intra-domain relationship, the required size of the common History is clearly related to the size of the community.

2.6. Probabilistic analysis

We have computed in [17] the probability that two randomly picken devices have enough common elements in their respective Histories to automatically trust each other. This approach is closely related to the birthday paradox. We observe that for a given group of size n, a History of size $k \sim n/\ln(n)$ and a threshold of common knowledges $p \sim \sqrt{n/\ln(n)}$, then the probability of automatically create a trust link is greater than 50%. For instance, if $n = 100$, $k = 22$ and $p = 5$, this probability is about $56,6\%$ (for the same parameters and $p = 3$, this probability reaches 92%). We see that the size k of the History is reasonable and could be easily carried by each device and the number of verifications to perform given by the p value is also not too excessive.

2.7. Eviction policy of elements of the History

We have also simulated in [17] an ambient network with power law distribution and uniform random distribution to describe the interaction between each device. Our analysis shows that the Least Frequently Used (LFU) policy mode was more efficient to preserve the cohesion of regular interacting devices than the First In, First Out(FIFO) policy. This result is what expected since the FIFO policy does not take into account the importance of the elements of History. On the contrary, if we only keep the most active and useful elements, the chances to find them in other Histories increases. The threshold can be fixed at 3 or 4 for communities with about 100 or 120 devices. Beyond, the protocol would require too many user interventions to be viable.

3. The Common History Extraction Protocol

In the KAA framework, the *Common History Extraction* protocol forms the core of the system and it is based on cryptographic methods. Although, the detailed cryptographic aspects are beyond the scope of this paper, we give here a brief description of the general scheme.

3.1. The CHE Protocol

A device in the KAA framework is equipped at least with a *cryptographic package* what will make it compatible de facto with any other entity of our model, i.e. other objects which implicitly accept the model. When an object received this package and the initial parameters, it can then initiate sessions of communication by the means of the CHE protocol (detailed in [16]). If a session is accepted, the two involved devices estimate, considering their security policies, that they can trust each other during this interaction.

Starting from an empty History, a device records all the successful interactions made with other devices in order to support the future spontaneous interactions. To prove the past interactions, it creates with each met device an element of History related to their respective identities and signed by the two parts. Before any interactions, devices must build a *trust germ*, by counting the number of common devices they have in their History, or by manually forcing the relation: this is the *bootstrap* phase of our model. If the number of common

interactions is sufficient (greater than a threshold p which is a function of the size n of the community and the maximum size H_{max} of the History), they can then interact.

Each device receives an initial *trust germ* from its *imprinting station*. It is composed by the following information: an identifier ID_u chosen by the device owner (eMail address or IP address or just a simple name or pseudonym) supposed to be unique within the security domain built by this imprinting station, an identity which is obtained from this identifier by concatenating it with a date d and a lifespan T ($\mathrm{ID} = \mathrm{ID}_u||d||T$), a first pair of private/public key $(S_{\mathrm{ID}}, Q_{\mathrm{ID}})$ for cipher operations, a second pair of keys $(S^S_{\mathrm{ID}}, Q^S_{\mathrm{ID}})$ for the signature and a set representing all the public parameters of the elliptic curves required along computations:

Params:

$$\Omega := \langle \mathbb{F}_p, a, b, P, h, G_1, G_2, e, H_1, H_2, H'_1, H'_2; P_{pub,\Omega} \rangle$$

where: a and b are the parameters of a particular elliptic curve $y^2 = x^3 + ax + b$ on $\mathbb{F}_p$; P, a particular point of this curve of prime order q; h, the cofactor defined as $h = \#E(\mathbb{F}_p)/q$; G_1, is a first additive cyclic group of prime order q built using the P point; G_2, a multiplicative cyclic group of the same order; e, a bi-linear pairing from $G_1 \times G_1$ to G_2; $H_1 : \{0,1\}^* \rightarrow G_1^*$ and $H_2 : G_2 \rightarrow \{0,1\}^n$, two map-to-point hash functions required for the Boneh-Franklin's Identity Based Encryption (BF-IBE) (see [23] for more details); and $H'_1 : \{0,1\}^* \times G_1 \rightarrow G_1$ and $H'_2 : \{0,1\}^* \times G_1 \rightarrow \mathbb{Z}_q$, two hash functions required for the Chen-Zhang-Kim IBS signature scheme (CZK-IBS) (see [24] for more details). Notice that the device public keys are directly derived from their identities due to the use of Identity-Based cryptosystems.

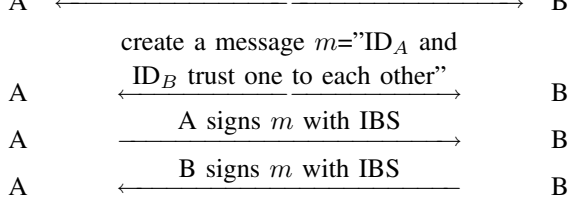
Another important point is that each smart device shares the same following cryptographic algorithms and protocols downloaded from the imprinting station: a fingerprint algorithm, a signature algorithm, a zero-knowledge protocol, a protocol to construct secure channel and the public parameters.

Ω-values are the domain identifier values provided to each device imprinted by the same imprinting station. Every imprinting station possesses the same Ω-values except $P_{pub,\Omega} = sP$ varying along the parameter s, the master key of a station. This value depends on each station and must be absolutely kept secret by it. None of these imprinting stations is supposed to be certified by any authority. Moreover, an independent mobile imprinting itself may be its own standalone security domain. The only values that each smart device has to keep secret is S_{ID} and S^S_{ID} as usually in cryptosystems.

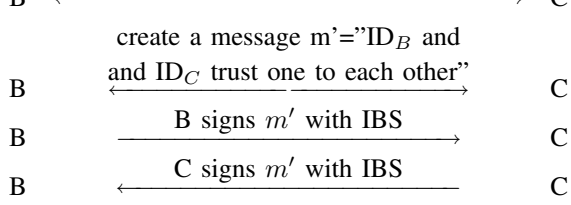
Notice that if a first identity is $\mathrm{ID}_1 = (\mathrm{ID}_u||d_1||T_1)$ where ID_u represents the name or a pseudonym, d_1 a date and T_1 a lifespan, this identity allows to generate the first corresponding key pairs. Then, the updated identity ID_2 is equal to $\mathrm{ID}_2 = ((\mathrm{ID}_u||d_2||T_2)||MAC((\mathrm{ID}_1||\mathrm{ID}_u||d_2||T_2), P_{pub,\Omega}))$ where d_2 represents a second date, T_2 another lifespan and MAC is a MAC algorithm. And so on, the next identities are created using the same operations, generating a MAC chain.

Once the initialization phase is done, a device may interact with other devices without any contacts with its imprinting station. This forms a second phase in the protocol.

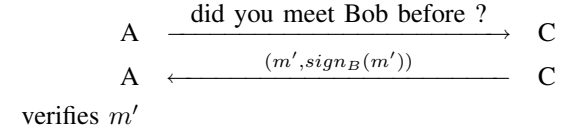
The first step of our protocol supposes that both entities Alice and Bob have already interacted at least once and have built a trust bond: this is a message m signed by Bob that Alice publishes in the public part of her History $(m, sign_B(m))$ while Bob publishes $(m, sign_A(m))$ in its own History. This bond could be created by forcing by the hand the beginning interaction as in a Bluetooth like system if the number of common elements of their history were insufficient. This forms the *bootstrap* phase. Let us note that if Alice and Bob have already met and if this new interaction is successful, they just have to modify the respective values of the intensity and to rebuild a new history element to replace the old one because it contains a timestamp. Suppose now that in the same way Bob and Charlie have built a secure channel to exchange a common message of mutual trust m'.



In the same way Bob and Charlie have built a secure channel to exchange a common message m'.



When Alice meets Charlie and one to each other want to prove that they have respectively met before Bob, they exchange a public part of their histories and Charlie, first, proves to Alice that Bob trust him using m'.



The second step of our protocol describes a trust bond establishment using the contents of History between two devices (here Alice and Charlie) that have never met. Thus, when Alice meets Charlie for the first time, they exchange the concatenation of all the public keys Q_{ID} contained in their history. Once this first exchange carried out, Alice and Charlie realize that they have both met before Bob and want to mutually prove this common meeting. Charlie, first, proves to Alice that Bob trusts him using the message m'. Alice could verify the contents of m' because she knows Bob's public keys

from her own previous meeting. The same process is repeated for Alice.

This protocol guarantees the traditional cryptographic properties: authenticity (as Charlie knows the Bob's public key, he could authenticate his signature), integrity is guaranteed by the hash function used in the identity based signature scheme as in the classical case of a certificate, confidentiality is guaranteed by the secure channel. The secure channel built at the beginning of the exchange in the first step also prevents a man in the middle attack: the reciprocal authentication between Alice and Bob is done by their own trust link and the secret is only known by the concerned devices and has been carefully exchanged using a public key cipher method based on identity based encryption.

3.2. The generation mechanism for the elements of History

The first step consists of deciding at which moment the generation of an element of History will begin. Of course, this generation could not happen before the exchange of services due to the fact that a cheater device could build a History without providing services and in this case, the model would not be anymore a trust framework. So, the generation of the elements of History must be performed after the service realization. In this part, we do not integrate a semantics for all possible services. We just consider here a successful interaction where the service is really provided.

It seems logical that the device that first gives an element of History is the device that receives the service from another. Thus, a terminal with a rich History could interact more easily with others.

Then, our algorithm could be written as follows:

- **First step (1):** suppose that the device A (the receiver) asks the device B (the provider) for a particular service.
- **Second step (2):** The device B could provide the service to the device A. In this case, the message formed to be signed by the two parties in presence will be m ='B provided a service to A'. In the case where the service is not furnished by B, the algorithm stops here. We call this property $sp_A(B)$='service property'.
- **Third step (3):** The device A could enrich the History of B by signing and sending to B the message m. In this case, we say that A is a trustor and we call this property the $tp_B(A)$='trustor property'. If the device A does not provide the element of History, the algorithm stops due to the 'non trustor property'.
- **Fourth step (4):** The device B has previously received from the device A the $tp_B(A)$ and it could be reciprocal: it could provide to the device A a element of History signing and sending the message m. We call this property $rp_A(B)$ ='reciprocal property'. Of course, the device B is not obliged to provide this element and could verify the 'non reciprocal property'.

This algorithm is illustrated on the following figure:

$$\text{A} \xrightarrow{\text{asks B for a service}} \text{B} \quad (1)$$

$$\text{A} \xleftarrow{\text{B provided the service}} \text{B} \quad (2)$$

$$\text{A} \xrightarrow{\text{A signs } m\text{=B provided a service to A}} \text{B} \quad (3)$$

$$\text{A} \xleftarrow{\text{B signs } m\text{=B provided a service to A}} \text{B} \quad (4)$$

In step (3), B keeps $sign_A(m)$ while A keeps $sign_B(m)$ in step (4).

3.3. Trust function

We discuss here the KAA trust function. Suppose that A and B are two devices belonging or not to the same security domain or community. The main parameter of the KAA trust function is the threshold p (a positive number) of common elements of History.

First of all, let us introduce the direct trust function:

$$d(A,B) = \alpha|H_A \cap H_B| + (\alpha - 1)|BL_A \cap H_B|$$

where α varies in the interval $[0,1]$. This coefficient indicates the weight of the number of common elements $|H_A \cap H_B|$ with regard to the number of untrustworthy devices of A that B considers trustfulness. The direct trust function obviously admits negative values due to the term $(\alpha - 1)|BL_A \cap H_B|$, otherwise we consider that if it exceeds the threshold p, then the direct trust level $T(A,B)$ will equal to one, otherwise it will equal to 0.

The KAA trust function is context-dependent, so we need to combine the direct trust value with other values given by the context of the interaction, limited here to the *intensity* of the relation between the two involved devices and to the *community parameter*. The device B compute the global trust function regarding the device A by using the formula:

$$TF_A(B) = \frac{\beta_A T(A,B) + \gamma_A \frac{I_A(B)}{I_{max}(A)} + \delta_A C(A,B)}{\beta_A + \gamma_A + \delta_A}$$

with $T(A,B) = 1$ if $d(A,B) \geq p$, 0 otherwise and with $C(A,B) = 1$ if $\Omega_A = \Omega_B$, 0 otherwise; where β_A, γ_A and δ_A are three parameters that belong to $[0,1]$; and where $I_{max}(A)$ represents the maximal intensity admitted by A. Then, the KAA trust function gives a trust notation that belongs to $[0,1]$. According this value and the trust threshold t_{ID} defined by each device, the involved devices could decide to interact or not.

β, γ and δ are coefficients which represent the weights of each parameter we want to take into account. They depend on the trust policy of each device. For example, a device will prefer, if it has never met a device C (then, the corresponding $I_A(C)$ value is equal to 0), to take into account the number of encountered devices represented by the β coefficient than the community one (for instance, they belong to the same tennis club). The δ coefficient represents the degree of structure of a community and will depend on the type of the community.

4. Analysis of the KAA framework

4.1. Semantics

Semantics attached to an element of history plays a crucial role in the KAA framework since all interactions are stored in the History and thus, this latter gives easily precious information on the previous behavior of a device: it permits to derive richer and more accurate information than if we only consider the cross-checking of Histories. By the way, it is possible to pre-establish a level of trust based not only on the *quantity* of interactions, but also on its *quality*.

Semantics brings a specific role for each device and permits to take into account the human reciprocal behavior, as described in experimental economic games or in peer-to-peer networks. For instance, the element of History received by a device could contain the following semantics:

- B provided a service to A: $sp_A(B)$ property.
- A was a trustor for B: $tp_B(A)$ property.

The element of History received by A could contain the following semantics:

- B provided a service to A: $sp_A(B)$ property.
- B was a trustee for A: $tp_A(B)$ property.
- B was reciprocal for A: $rp_A(B)$ property.

In our framework, each device is incited to be a trust(or)/(ee) since this is the only way to enrich its own History. This capacity of course do not exclude the *free rider* behavior. As well, B has not a real advantage to be reciprocal except if it takes into account the general enrichment of the network and of its personal reputation with other devices.

When two devices enter in an interactive session, they inverse the two roles during the session and at the end, each one possesses an element of History. A trustor proof is an element of History constructed by the first statement and a reciprocal proof, by the second. All these properties are hardened by the identity-based cryptosystemes used in our framework. This cryptographic protocol also guarantees that the elements of History could only be used by the device that conceives it. Each device is not obliged to provide the element $tp_B(A)$. The same remark holds for the element $rp_A(B)$.

The use of a blacklist process is highly desirable to maintain malevolent devices under control. In an ambient network, the only reasonable possibility to sanction theses devices is to add them in a blacklist whenever what they did not respect a chart of good behavior. We suppose that a blacklist operation will only take place after a threshold of non trusted interactions.

In a risk evaluation approach, it can appear logical to penalize the devices which present recommendation coming from devices which are blacklisted. The disadvantage of a weak blacklist policy is to prohibit some interactions with honest devices only because some elements of their history have been locally blacklisted. There is thus a balance to find between a very strict policy which will have a very negative impact on the whole of the community and too permissive policy which will imply engage a too important risk.

Since the KAA framework supposes that there is no recommendation model, then no central authority could act for any sanction. The only sanction that a device could undertake if it is not satisfied by a device (the non sp, non tp or non rp properties) is to blacklist it. This sanction implies that the untrustworthy device will not anymore interact with the deceived device.

4.2. Security requirements

The following traditional cryptographic properties are guaranteed by our protocol: an offline authentication (users performs each other a weak authentication using the IBE scheme and as Charlie knows the Bob's public keys, he could authenticate his signature), integrity is guaranteed by the hash function used in the IBS scheme as in the classical case of a certificate, confidentiality is guaranteed by the use of the cryptographic IDs. Those IDs also permit to guarantee that the first phase of our protocol was correctly done. The secure channel built at the beginning of the exchange in the first phase also prevents a man-in-the-middle attack.

4.3. Security analysis

In the KAA framework, an intermediate malicious or corrupted device should not be able to destroy a complete chain of trusted relations since the trust establishment is not transitive, contrary to any PGP-like trust establishment framework. Our protocol CHE prevents any subgroup of devices from easily destroying a particular device reputation since each element of History is cryptographically proved and thus, cannot be used by another party. A trust notation or reputation principles as proposed for example in [25] may be also added to the global architecture.

Suppose now that an attacker has stolen the identity and the History of a device, but he was unable to steal the secret key: he does not know the secret S_{ID} provided by the imprinting station otherwise he could create all desired clone of this device. So, even if this attacker knows all the content of the History, he could never prove by a zero-knowlegde mechanism that it is the real device.

4.4. Classical attacks

As mentioned in [16] and due to the use of the IBE-scheme, the well known key escrow drawback is inherently present in our protocol. We then suppose that all the imprinting stations must be trusted entities. Otherwise, they can read and send messages instead of devices. However, the signature scheme used here prevents such an attack from happening because the signature key pair generated is unknown from the imprinting station.

Our trust management framework is a cross-domain protocol: two devices, not belonging to the same domain (or to the same imprinting station) could nevertheless interact by comparing the contents of their respective Histories once they exchange the public key of their security domains (we suppose here that all the other parameters are the same). Our protocol also guarantees the non-transferability of the History because only the knowledge of the secret keys allows to use the content of the History (the secure channel initially built prevents the use of the elements of History). Then, stolen identities or pseudonyms or histories could not be useful.

4.5. Sybil-like attacks

A user could utilize our protocol to forge several identities or pseudonyms from the same or different imprinting stations and then uses them in a Sybil-like attack. However, in our framework a unique identity or pseudonym could only be linked with a particular History.

For example, suppose that an attacker (Eve) wants to attack another device (Alice), then she creates first several pseudonyms $\mathrm{ID}_1, \cdots, \mathrm{ID}_n$. Alice asks her a particular service, they realize that they have enough common elements of History to interact. Suppose now that Eve does not provide the corresponding service to Alice with her ID_1 pseudonym, Alice then decides to blacklist the ID_1 Eve's pseudonym. So, Eve must use another pseudonym, ID_2 for example, if she wants to interact and attack Alice again. To manage this operation, she must build another time a sufficient number of elements of History common with Alice. Even if, she knows the pseudonyms of devices to meet again with her second pseudonym, she must play an active and positive role inside the ring of Alice's trusted devices. So, this attack consisting to employ several pseudonyms is very expensive and requires a hard social engineering.

4.6. Clone attacks

As mentioned in [16], one major attack against our model is the clone attack in which Alice clones her identity among several other terminals. Those clones have exactly the same keys and thus could build a very large History if they gathering all their recorded elements: the resulting History could interact more easily than the others.

However, Alice cloned devices should be carried by different persons visiting different places in order to have different Histories. This fact thwarts the risk since it is a social engineering attack which is difficult to conduct as well as difficult to surround by cryptographic methods.

5. Privacy and anonymity in the KAA framework context

Defining an access control model taking into account the lack of trust and communication infrastructure, and the requirements for privacy and context awareness is a challenging task. To decide whether an entity can be authorized to access a service, it is necessary to rely on trust relationships. When no trust is defined, this means establishing trust based on observation, recommendation, or reputation mechanisms. The fact that the communication media is not reliable disables access control based on trusted third parties and promotes the use of the KAA framework.

Privacy, in terms of anonymity of the credential holder and linkability of the elements of History lead to contradictual and hard constraints. For instance the location of a device at a given time and what it did.

5.1. Context awareness

In an ambient network the context refers to any information on the local environment: time, number and presence of other devices, etc. A solution for monitoring the context and subsequently using this measure as a proof would be useful. However, only results signed by a trusted device could be seen as proof, and, even in this case, it is easy to tamper with context. Due to this fact and that all contextual information can be manipulated by an attacker, only self-evaluation is taking into account by each device in our framework. However, the intersection of History is not always sufficient for spontaneous interaction. The objective of introducing a context awareness evaluation for the device is to detect as quick as possible in which environmment this device is really embedded.

The introduction of such an evaluation is quite simple in our framework. Each device can carry out easily a passive listening of the elements of the Histories exchanged by two other devices in its radio range. Then, it can compute the cardinality of common trusted devices it shares with these both devices. By this possibility to regularly sense its environment, the device can evaluate the social situation in which it evolves. If the cardinality of intersected Histories is always high, the device could consider that it acts in a well known environment and may adapt its security policy to this context.

The use of the CHE protocol confers two major advantages. First of all, context awareness evaluation is based upon provable data. Secondly, with only some few interceptions of communication, devices obtain a large list of elements of History. This latter point is important to reduce the cost of such a listening. The context awareness could detect any change in the environment. For instance, it is valuable to detect any gap in the ratio of known trusted devices over unknown ones. Known trusted devices mean here those stored in the respective Histories. This ratio may be sufficiently accurate with few sampling communications. This point can be proved analytically by using a method similar to the birthday paradox.

5.2. Privacy

Privacy is defined as the right of an individual to be secure from unauthorized disclosure of information about oneself. Different services are associated with privacy: anonymity, pseudonymity, unlinkability, and unobservability.

Privacy is a major concern in the KAA framework since applications designed to work in ambient networks do not reduce the privacy of users: public interaction between all devices can be automatically logged in order to make a profile of the owner. The gathering and combination of this data is an important threat against privacy that can lead to targeted attack. Moreover, real-time location of these devices can also be achieved thanks to their unique identity during some period. The future development of ambient networks will affect more and more daily interactions and thus if no privacy protection is provided, it will be possible to log any interaction. Indeed, wireless communicating mobile devices could feed huge databases logging daily interactions of users and enable to accurately profile users leading to violation of their privacy. At the application layer, the attributes that are revealed have to be carefully chosen in order to avoid traceability based on the attributes. It is obvious that identity certificates enable traceability of holders even when the elements of History is

unlinkable. Statistical disclosure control aims at controlling what information can be revealed without threatening users privacy.

Our architecture for protecting the untraceability of users in such a context relies on pseudonymous data (see next subsection). After the acces control phase, privacy at network level is required in order to prevent tracing of user behavior based on the monitoring of network traffic. Unlikely, the elements of History delivered in the exchange phase can be traced, though they serve to sense the context (see subsection 5.1). Indeed, private user data (e.g. his own identity) can be directly exposed through our trust protocol since the mechanism is based on identification and authentication.

5.3. Anonymity

The CHE protocol is based on identity and this point could constitute a risk against the preservation of privacy and anonymity. Anonymous data is non-personal data which, by itself, has no intrinsic link to an individual user. For example, hair color or height (in the absence of other correlating information) does not identify a user. Similarly, system information such as hardware configuration (e.g., CPU and memory size) is anonymous when it is not tied to an individual. Data can also lose its anonymity as the volume of data collected increases. The more information that is known, the greater the chance a link to an individual can be made, especially in situations where there is a small population of possible candidates. Privacy and anonymity are important to avoid that all daily interactions of users be logged. However, privacy (as well as security) always has a cost in terms of communication and delays, computational power, memory, etc. Moreover, even with unlinkable data, it is necessary that the communication channels protect the privacy and that the application carefully controls the disclosure of attributes.

Obviously, at first sight the CHE protocol does not deal with anonymous data since a device identity could be related to its owner. Personally identifiable information means any information that identifies or can be used to identify, contact, or locate the person to whom such information pertains, or from which identification or contact information of an individual person can be derived. It includes, but is not limited to: name, address, phone number, fax number, email address, financial profiles, medical profile, social security number, and credit card information. If a unique identifier is introduced that ties the data to an individual, the data is no longer anonymous. Each device should use pseudonymous data as identity since such data is unique information that by itself does not identify a specific person (e.g., unique identifiers, biometric information, and usage profiles that are not tied to an individual), but could be associated with an individual. Once this data is associated with an individual it must be treated as personal information. Until that time, it may be treated as anonymous. When it is necessary to store sensitive personally identifiable information on the user's device, the user must provide consent, and the data should be stored only for the shortest amount of time necessary to achieve the specific communication purpose. This data must be stored with the appropriate safeguards and mechanisms to prevent unauthorized access.

A user could preserve its anonymity because he is free to choose his own pseudonyms according to the context and he may have several pseudonyms transmitted by different imprinting stations. Those pseudonyms are certified through the use of identity-based cryptosystems schemes and are necessary to preserve the real identity of their owner, even if his meetings when he acts in the network are known with other peers with pseudonyms. Moreover, each pseudonym defines its own History and all the pseudonyms are certified, thus tackling the well-known *Sybil attack*. Our model also guarantees the non-repudiation: each user is preventing from denying previous meetings or actions. Revocation is also possible using the timestamp linked with an element of History and thus included in the key pairs (as previously described in 3).

6. Conclusion

We have proposed a distributed framework that produces trust establishment based on direct proved experience. Our cross-domain scheme supports a weak authentication process, user anonymity and resists to various attacks, especially the Sybil-like attack. We have designed a trust notation that takes into account local blacklist process and context awareness. Finally, this framework is suitable for large communities and the bootstrap phase is not an obstacle. Malicious behaviors of some peers are prevented by the use of a blacklist process wich protects trusted devices. As part of a future work, we will investigate the dynamics of our model behavior for different social cases.

Acknowledgment

This work is done in the *Knowledge Authentication for Ambient* (KAA) project supported by the ACI-SI program of the French ministry of research.

References

[1] J. Mundinger and J.-Y. Le Boudec, "Analysis of a Reputation System for Mobile Ad-Hoc Networks with Liars," in *The 3rd International Symposium on Modeling and Optimization in*, 2005.

[2] B. Yu and M. P. Singh, "A social mechanism of reputation management in electronic communities," in *Cooperative Information Agents*, 2000, pp. 154–165. [Online]. Available: citeseer.ist.psu.edu/yu00social.html

[3] G. Suryanarayana and R. N. Taylor, "A survey of trust management and resource discovery technologies in peer-to-peer applications." [Online]. Available: citeseer.ist.psu.edu/suryanarayana04survey.html

[4] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis, "The KeyNote Trust-Management System Version 2 - RFC 2704," RFC 2704, Available from http://www.faqs.org/rfcs/rfc2704.html, September 1999.

[5] M. Blaze, J. Feigenbaum, and A. D. Keromytis, "The role of trust management in distributed systems security." in *Secure Internet Programming*, ser. Lecture Notes in Computer Science, J. Vitek and C. D. Jensen, Eds., vol. 1603. Springer, 1999, pp. 185–210.

[6] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management." in *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 1996, pp. 164–173.

[7] T. Grandison and M. Sloman, "A survey of trust in internet applications." *IEEE Communications Surveys and Tutorials*, vol. 3, no. 4, 2000.

[8] R. Khare and A. Rifkin, "Weaving a Web of trust," issue of the World Wide Web Journal (Volume 2, Number 3, Pages 77-112), Summer 1997.

[9] J. Sabater and C. Sierra, "Regret: reputation in gregarious societies." in *Agents*, 2001, pp. 194–195.

[10] J. Sabater, "Reputation and social network analysis in multi-agent systems." in *AAMAS*. ACM, 2002, pp. 475–482.
[11] J. M. Pujol, R. Sangüesa, and J. Delgado, "Extracting reputation in multi agent systems by means of social network topology." in *AAMAS*. ACM, 2002, pp. 467–474.
[12] V. Legrand, D. Hooshmand, and S. Ubéda, "Trusted ambient community for self-securing hybrid networks," INRIA, Research Report 5027, 2003.
[13] S. galice, V. Legrand, P. Neuville, and S. Ubéda, "Identification dans les réseaux spontanés," in *Conférence sur la Sécurité et Architectures Réseaux*, June 2005.
[14] S. Galice, V. Legrand, M. Minier, J. Mullins, and S. Ubéda, "The kaa project: a trust policy point of view," INRIA, Research Report RR-5959, 2006.
[15] S. Galice, V. Legrand, M. Minier, J. Mullins, and S. Ubéda, "A history-based framework to build trust management systems," in *Second International IEEE SECURECOMM Workshop on the Value of Security through Collaboration (SECOVAL 2006)*, august 2006.
[16] S. Galice, M. Minier, J. Mullins, and S. Ubéda, "Cryptographic protocol to establish trusted history of interactions," in *Third European Workshop on Security and Privacy in Ad hoc and Sensor Networks*, september 2006, p. LNCS 4357.
[17] S. Galice, M. Minier, and S. Ubéda, "A trust protocol for community collaboration," in *Joint iTrust and PST Conferences on Privacy, Trust Management and Security*, July-August 2007, p. IFIPTM 2007.
[18] L. Capra, "Engineering human trust in mobile system collaborations." in *SIGSOFT FSE*, 2004, pp. 107–116.
[19] L. Bussard, R. Molva, and Y. Roudier, "History-based signature or how to trust anonymous documents." in *Second International Conference on Trust Management*, O. T. Dimitrakos, Ed., April 2004, pp. 78–92.
[20] M. Jakobsson and S. Wetzel, "Security weaknesses in bluetooth." in *CT-RSA*, ser. Lecture Notes in Computer Science, D. Naccache, Ed., vol. 2020. Springer, 2001, pp. 176–191.
[21] N. Prigent, J.-P. Andreaux, C. Bidan, and O. Heen, "Secure long term communities in ad hoc networks," in *1st ACM workshop on Security in Ad hoc and Sensor Networks (SASN)*, 2003.
[22] D. Quercia, S. Hailes, and L. Capra, "Tata: Towards anonymous trusted authentication." in *iTrust*, ser. Lecture Notes in Computer Science, K. Stølen, W. H. Winsborough, F. Martinelli, and F. Massacci, Eds., vol. 3986. Springer, 2006, pp. 313–323.
[23] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing." in *CRYPTO*, ser. Lecture Notes in Computer Science, vol. 2139. Springer, 2001, pp. 213–229.
[24] X. Chen, F. Zhang, and K. Kim, "A new ID-based group signature scheme from bilinear pairings." in *Information Security Applications, 4th International Workshop - WISA'03*, ser. Lecture Notes in Computer Science, vol. 2908. Springer-Verlag, 2003, pp. 585–592.
[25] J. Liu and V. Issarny, "Enhanced reputation mechanism for mobile ad hoc networks." in *iTrust*, ser. Lecture Notes in Computer Science, C. D. Jensen, S. Poslad, and T. Dimitrakos, Eds., vol. 2995. Springer, 2004, pp. 48–62.
[26] *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy, Proceedings*. ACM, 2002.

Samuel Galice is actually a Ph.D. student at the INSA de Lyon (Institut National des Sciences Appliquées) since December 2004. His main research interests include several aspects of Security and Cryptography, in particular Trust Management Framework and Security in Ambient Networks, Identity-Based Cryptosystems, Authentication Protocols, Anonymity, Privacy.

Marine Minier received the Ph.D. degree in 2002, from the University of Limoges. In 2005, she joined the INSA de Lyon (Institut National des Sciences Appliquées), as an Assistant Professor, in the CITI laboratory, a team working in telecommunications. Her research interests include Symetric Key Cryptography, Security in Sensor Networks and in Ambient Networks.

After a PhD in Computer Sciences from the Ecole Normale Supérieure de Lyon in 1993, **Stéphane Ubéda** was associated professor in the Swiss Federal Institut of Technology until 1994. He was associated professor in the Jean-Monnet University (Saint-Etienne) up to 2000 and went to the Institut Nationale des Sciences Appliquées de Lyon, as full professor in the department Telecommunications. Nowdays, he is the head of the CITI Lab attached to this department. His main interest concerns distributed algorithms and protocols evaluations.