A Local Broker Enabled MobiPass Architecture for Enhancing Trusted Interaction Efficiency

Will Tao, Robert Steele

Department of Computer Systems, Faculty of Information Technology
University of Technology, Sydney
PO Box 123, Broadway, 2007, New South Wales

{wtao, rsteele}@it.uts.edu.au

Abstract

While mobile computing provides a potentially vast business opportunity for many industry participants, it also raises issues such as security and performance. This paper proposes a Local Broker enabled MobiPass architecture based on our previous research outcomes. Our MobiPass architecture can convert the unpredictable and highly dynamic mobile environment into a trusted business platform. By setting customised rules against a MobiPolicy, the Mobipass architecture enables fine grained access control without necessarily having a prior knowledge or interaction with other encountered parties and environments. This paper extends our MobiPass architecture by introducing an additional element – the Local Broker, to enhance the architecture's performance and efficiency. A detailed case study has been provided to explain the role that the Local Broker takes in the architecture.

Keywords: Mobile Computing, Ubiquitous Computing, Trusted Interaction

1 Introduction

Recent advances in technology have provided portable devices such as the mobile phone, personal digital assistant (PDA), portable data terminal (PDT) and smart phone with wireless computing capabilities. This kind of wireless computing model is often referred to by the generic term "mobile computing" and has already attained a substantial fundamental role in the business world.

However, to gain wide acceptance and success with this computing model, certain conditions will need to be satisfied before applying mobile computing into a critical, enterprise level system. An example of an inhibitor that deters mobile computing is that it is very difficult to build a trusted environment among all transacting entities within a mobile environment as it is highly dynamic and unpredictable (Satyanarayanan 2000, Ranganathan 2004). Unlike the traditional computing environment that is static and closed, with fixed, well-known entities within the network, mobile computing involves a large number of interactions, co-ordinations and collaborations with a large number of casually accessible yet portable mobile devices. The strategy and approach in building a trusted environment is fundamentally difficult and different when compared with more static networks.

In the case where there is a limited amount of knowledge about different transacting entities, a feasible mechanism that protects sensitive information and determines the level of trust between those entities in the mobile computing network is essential, as a lack of trust can result in failure to implement business models that build on top of this mobile environment. In addition, users will not be willing to participate as they do not have confidence in interacting with each other.

In this paper, based on our previously proposed MobiPass architecture, we put forward an alternative approach to establish a trusted interaction in mobile computing. The new approach introduces the new element, Local Broker (LB) into the architecture that will enhance performance, flexibility and other aspects. The case study, described in Section 4, will clearly illustrate the architecture.

The paper is structured as follows – Section 2 provides a review of the MobiPass architecture with a brief explanation, and Section 3 describes the Local Broker based MobiPass architecture. A case study is examined in Section 4 and in Section 5 related work is discussed followed by future work and the conclusion in Section 6.

2 MobiPass Architecture Review

The purpose of the MobiPass architecture is to help mobile entities to establish trusted interactions and provide a fine grained information access control among those transacting entities. The definition of trust in this paper is defined as "a subjective expectation about other's future behavior" as we believe that the trusted platform is a necessary and non-replaceable condition that can enable mobile computing to achieve a higher level of success.

MobiPass is a generic architecture that creates a flexible and secure environment in mobile computing; it can be applied to a large number of scenarios where trusted interaction is required. As the MobiPass architecture utilises and extends digital certificate technologies to provide more detailed certified information in a *distributed* manner, it is not necessary to have a central server to implement trusted interaction among large numbers of mobile entities. This distributed nature is a critical attribute, given the vast number of potential mobile entity interactions for which trusted interaction must be achieved. To clearly describe the MobiPass architecture, we will use the mobile phone as an example to demonstrate how this architecture works within a mobile computing environment.

By enabling a set of customised preset preferences, the MobiPass architecture allows mobile entities which are previously unknown to each other to interact and communicate in a trusted manner. In the architecture, the mobile entity only talks with and makes itself visible to the trusted entity/environment which satisfies the customised access control rules.

The core elements in the MobiPass architecture are: The Central Registry, MobiPolicy, Extended Certificated Authority (ECA), MobiPass and the MobiManager (Figure 1), for a more detailed description of its architecture and functionality, see (Steele, Tao 2006, Tao Steele 2006):

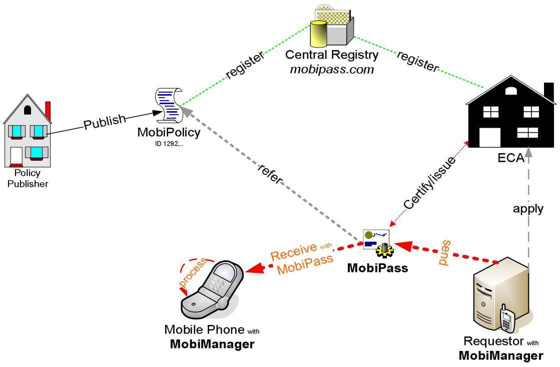

Figure 1: The overall MobiPass architecture

Figure 1 shows that the ECA is an extension of the currently known certificate authority which issues MobiPasses. MobiPass works like a passport in our architecture which is described by XML and complies with the corresponding XML schema, represented in a MobiPolicy. It contains the real data describing a particular service and/or mobile entity in relation to a certain service. Due to the diversity of ubiquitous computing, it is impossible to have one universal specification to model all sorts of services and entities. Therefore a MobiPolicy is introduced to distinguish individual services. It provides a flexible and extensible approach to describe the service and/or mobile entities based on relevant information for this particular service, and MobiPolicy is represented by XML Schema in our architecture. MobiPolicy can be published by any organisations for any services, but the procedure in issuing a corresponding MobiPass is controlled by the ECA, which can also be the same entity as this policy publisher. Moreover, as there is no restriction for any organisation to be an ECA, a non-mandatory Central Registry is introduced to manage all these ECAs. It should be noted that the word *central* in our architecture is only a logical concept. The implementation of a central registry can be totally distributed. The MobiManager is an extra module which is installed on handset devices such as the mobile phone to perform all necessary operations, for example: sending and receiving MobiPass, parsing an incoming MobiPass, helping users to do their preference settings and detecting other surrounding MobiPass devices.

In the MobiPass architecture, multiple ECAs are allowed in the MobiPass with different levels of trustworthiness. Any entity within the MobiPass architecture can act as the ECA to issue certified evaluation results, also, multiple policies are used for different services. A customised policy can be published by any entity to meet the requirement for their particular service.

3 Local Broker Enabled MobiPass Architecture

3.1 Architecture Overview

As described in Section 2, the MobiPass architecture allows previously unknown entities to communicate with each other in a trusted manner. However, in some cases, the performance can be improved if we introduce a Local Broker (LB) into the architecture. As asymmetric key encryption is relatively resource consuming, the performance might be an issue for MobiPass architecture adoption (Diffie 1998, Lenstra & Verheul, 2000). Based on our current research, we have found that in many cases, where mobile entities are previously unaware of each other's existence, there is usually a broker that links all these mobile entities and the broker knows how to deal with each other. Consider the following analogy. Bob organised a party and he invites group of people. However, his guests may not know each other because some of them are Bob's classmates while some of them are his business associates and the rest of them may be his relatives.

Although they do not know each other, one common thing is that they all know Bob and he knows how close he is with each one of them (authentication). If we assume that all guests trust Bob, hence they will assign each other a minimal level of trust, until they have received further information from Bob. The same scenario happens quite often in the area of mobile computing, i.e. the host (referred as the LB in this paper) which provides the service has enough knowledge of participating mobile entities and knows how to assign privileges to different mobile entities with fine grained access control level, and all these mobile entities fully trust the LB (see case study in Section 4). This means that as long as these mobile entities can establish a trusted relationship with the LB, that trust can be expanded across that service network.

Based on our previously published MobiPass architecture, we have developed a variant, the LB enabled MobiPass architecture. This architecture introduces a new element - the LB, and the assumption in this architecture is that the LB is fully trusted during the entire interaction. The LB is the core in this architecture as it does the initial authentication and authorisation and it needs to decide how to assign privileges to different entities. The LB is also responsible for announcing the service so when a MobiPass enabled device enters into the vicinity, it can easily discover the desired service which is being advertised and attempts to initiate communication. To improve the availability, LB can be deployed in a clustering mode which means that multiple LBs can share a single access point address and synchronise the application data in real time. Also, when mobile devices cannot communicate with the LB, devices will try to skip the LB and interact with other transacting parties directly.

An important point is that the LB only works locally, i.e., there is no centralised broker. To establish a trusted link that facilitates interaction between devices, different elements are required to collaborate with each other.

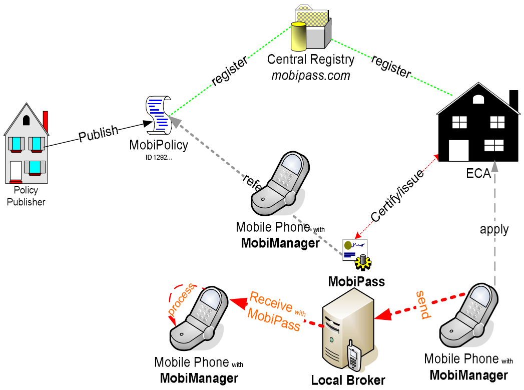

Figure 2: Overview of the Local Broker enabled MobiPass Architecture

Figure 2 demonstrates how the service works. MobiPass enabled devices will keep on discovering the available services.

Compared to the normal MobiPass architecture, there are no changes in how MobiPass is applied by MobiPass users and is granted by ECAs. The users will still go through the normal processes, i.e., first, users are required to find the right MobiPolicy then fill in all necessary information for that particular MobiPolicy. Next, the MobiPolicy will then need to be sent to the ECA and apply to the MobiPass. If all information has been verified as true and genuine, the ECA will issue a MobiPass to this user which contains a valid ECA's digital signature.

The difference with the LB based MobiPass architecture is that rather than having a direct communication with each individual mobile entity, the MobiPass enabled device will talk to the LB instead of directly to the targeted entity. Additionally, the LB can act as a pure "forwarder" or fully on behalf of the involved mobile entities. It should be noted that the LB also holds a special role, it needs to apply a MobiPass from the relevant ECA, which indicates that it is from the MobiPolicy publisher, and the ECA will also verify the relevant documents to make sure that all information in this MobiPass is genuine. It will then sign this LB's MobiPass using the ECA's private key. When the MobiPass is installed in the transaction entity's mobile device, a XML schema based user interface generating system, Xplorer, (Steele et al 2005) will generate preference settings interface based on MobiPolicy's XML schema and users will be required to fill in the options based on the incoming MobiPass's data. For example, in a Mobile social introduction service, users can set his/her preference for their device to only look for software engineers whose age is between 30 and 35. Once the preference is set, users can activate his/her MobiPass enabled service on his/her mobile device. It should also be noted that this LB enabled MobiPass device will keep on discovering any available services which match the MobiPolicy. On the other hand, unlike the MobiPass enabled device, LB enabled MobiPass devices will not advertise themselves; it will only discover the service. The LB is responsible for advertising the available services as all communications are surrounding the LB, so the LB will keep on announcing the services with their corresponding MobiPolicy IDs.

Once the MobiPass enabled device has found the right MobiPolicy ID through the LB, the following actions will take place:

1 The mobile device will ask the LB to send the MobiPass to check whether this LB is a genuine ECA signed Broker.

2 After receiving a MobiPass from a LB, a check will take place to assess whether or not this LB is a Known Local Broker (KLB). A KLB is one that has a public key already existing in the mobile device or its MobiPass can be found in the list containing existing trusted MobiPasses.

3 In the case where a LB is not a KLB, the normal procedure for verifying MobiPasses will be applied onto this broker's MobiPass. Once this has been done, the LB can be confirmed as a true LB for the advertised MobiPolicy.

4 The mobile device will send a list of supported symmetric key algorithms such as DES or BlowFish, it will also send out the MobiPass which contains the user's public key, as well as the ECA's public key and signature to the LB for the random security number.

5 The LB will try to validate the incoming MobiPass by evaluating ECA's signature to see whether this MobiPass is valid, assuming that LB has a very good knowledge of ECAs, especially for ECA's public keys. If the incoming MobiPass is considered as a valid MobiPass and matches the MobiPass's holder, the LB will then choose a supported encryption algorithm, generate a security token (e.g. a large random number), which will then be encrypted by the MobiPass holder's public key that is extracted from the MobiPass. The encrypted security token will then be encrypted by the LB's private key and then sent back to the mobile device.

6 Once the mobile device has received the enhanced MobiPass from the LB, it will decrypt the message by the LB's public key and the MobiPass holder's private key to get the original sender's MobiPass and run the normal MobiPass interaction procedure to establish the trusted interaction.

After this, both LB and the mobile device know the secret key and the secret key is only shared between these two parties. To perform the service smoothly for interacting mobile entities, a session timeout value can be set at the LB to prevent the interacting entities accidentally dropping out of the service. As it is not unusual that the mobile device may roam out of the service vicinity temporarily, once the LB receives the incoming MobiPass and this MobiPass cannot be delivered, this MobiPass will be kept until the session timeout value has been reached. This means that after this time, the LB will consider that this transacting entity has formally quit the session (service).

As previously mentioned, the LB runs under two modes, they are:

1. Forward only mode
2. Access Level Control (ALC) Mode

Depending on the current condition, the LB will run in either of the modes, the following sub-sections will explain these two modes in detail.

3.2 Forward Only Mode

The Forward Only Mode means that the LB will only pass and forward MobiPasses among mobile entities, no further operations/processing will be made.

One advantage of the Forward Only Mode is that all mobile devices will only communicate to the LB. Once a successful handshake has been initiated, there is no need for the asymmetric encryption anymore in ongoing communications. The mobile device will automatically discover each other, and track down the address. Therefore when they want to start the communication, they only need to forward the MobiPass to the LB, along with the destination address. Once the LB receives the MobiPass, it will decrypt the MobiPass using the secret key which is shared by the sender. When decryption is successfully finished, the LB will encrypt the MobiPass content using another secret key which is shared by the receiver. Therefore during the message transmission stage, only the sender, receiver and LB can read the message, and as the LB is fully trusted, it means that during the transaction of the message, the MobiPass will be safe and there is no need to contact the ECA to ensure that the MobiPass and the content in the MobiPass is authentic. This mode allows mobile devices within the entire network to each do one public key encryption operation and the rest of the operations will be conducted by private key encryption. This greatly improves the performance and eases the communication especially when there are a large number of transacting entities.

When the mobile entity receives the incoming MobiPass, they will run the whole workflow as discussed in our previous paper, please note that even if the MobiPass has successfully been delivered it does not guarantee that a transaction will be conducted. Interactions between devices will only be conducted when the MobiPass matches the receiver's profile to present access control rules.

3.3 Access Level Control (ALC) Mode

ALC mode is a more advanced mode for the LB. Rather than just simply forwarding the incoming MobiPass, it actually runs the authentication and authorisation for the MobiPass.

Once the LB and the mobile entities have finished the handshake, the LB will request the copy of the mobile device's preference for this MobiPass/MobiPolicy profile, and this preference setting will be transmitted by using the shared symmetric key between this mobile device and the LB. In this case, no public key encryption is required anymore as the shared secret key and algorithm is sufficient to identity the sender's ID.

The entire handshake is finished once the LB has successfully received the preference settings. Extending the forward mode, the LB not only forwards the MobiPass, it also runs the preference check on users' behalf every time a transaction occurs. As discussed previously, after the MobiPass has been verified, the LB will request the encrypted preference settings from the mobile device, as the LB has full knowledge of this MobiPolicy, it is very easy for the LB to check all incoming MobiPasses. Under this mode, the LB is responsible to perform the authentication and authorisation for building the trusted interaction between two mobile entities. The steps for communications are explained below:

- Receive the sender's MobiPass.
- Decrypt the incoming MobiPass by using a shared secret key with the sender.
- Detect the designated device from the MobiPass
- Look up the preference settings of this destination device, if the preference setting can not be found, then it will contact the designated device to initiate a handshake, then the symmetric key will be shared and the encrypted preference setting will be acquired.
- If the destination device is no longer in the network, the sender will be notified, otherwise, the LB will extract all the values and compare to the receiver's preferences. If the incoming MobiPass matches the receiver's preference settings, it will be forwarded to start conducting a trusted transaction, otherwise, a request will be sent out.

The ALC mode will greatly reduce the load of the mobile devices, as the computational part has been successfully transferred to the LB. As the LB is not necessarily a mobile device, it can in all probability easily handles the load and perform the authorisation as well. In the next section, a detailed case study will be given to clearly describe the LB enabled MobiPass architecture.

4 Case Study

In this section, a university community based case study will be used to explain how the LB enabled MobiPass architecture can assist in establishing a trusted interaction as the research community is familiar with the university environment. It should be noted that the scope of the LB enabled MobiPass architecture is not limited to the university environment, any environment which requires a trusted interaction between several mobile entities and satisfies the requirements of the LB i.e. transaction entities might not know each other but they all trust the LB, can benefit from this architecture.

These are the facts that exist in most public universities:

- There are a large number of students in the university; the number of student can exceed 100,000, and many different units coexists in the university, such as faculties, departments, service units, student unions and clubs.
- Most students only know a limited number of other students in the university. However

collaborations are often required, even though students/staff do not know each other.

- Every staff/student is supposed to trust the administration unit in the university.

From these facts, we can derive that there is a demand for a trusted interaction and it is not an easy task to implement such an interaction within the university as there might be a large number of staffs and students, all with a different background e.g. language and culture. Moreover, most of them do not have a previous knowledge of each other. For example, a group of students from different faculties doing some outdoor activities together, or they are looking for a flat mate to share accommodation with. A trusted interaction is required for transactions within all the above mentioned cases. We will now use the Accommodation Finder Service (AFS) as an example to demonstrate how the LB enabled MobiPass architecture works.

As there are many rural, inter-state and international students in the university, it is necessary for them to find their own accommodation as it might not be financially feasible for them to travel to the university from home everyday, so some students will go and find others to share accommodation. Also, due to security concerns, students like to share with other students from the same university; some students even like to share accommodation with others who have the same background or interest. Currently the main approach which has been used in many universities to find a flat mate is to read notes or advertisements posted on bulletin boards, this way can be dangerous as by just referring to the information given out in the post, there is no way in telling whether the information is true. Also, as this is not a real time interaction, a student will need to arrange and meet with a potential flat mate somewhere else. This kind of appointment can be dangerous, especially for female students. By using the LB enabled MobiPass architecture, it is very easy to enable a trusted interaction for the AFS.

In this case, the ECA and MobiPass publisher will be the Student Service Union (SSU) and as students currently trust the SSU; we can assume that trust will extend to the SSU published ECA and MobiPass. SSU can provide a online form which has an equivalent schema as the AFS MobiPolicy, and students can carry out and apply for their AFS MobiPass from this portal. Students will only be required to fill in extra information such as their interests and hobbies, as the SSU already has part of the student's personal information which has already been authenticated, such as their real name, gender, age, major and nationality. The extra information will only act as a supplement to help students to find flat mates who are more compatible with them and therefore certification is not required. Once they have filled in the forms, the SSU will generate the MobiPass for the AFS for this student.

For instance, Alice is a 20 years old first year international student who is studying computing science, and she is currently looking for a flat mate. Alice would like to share accommodation with another female international student of a similar age and background because this makes her feels comfortable and safe. Therefore her preferred flat mate will be a female student, aged between 20 to 25 years old, and can speak her language. Alice does not want to use the traditional approach to find flat mates as Alice can only gather potential flat mate's information by reading posts and there is no way to tell whether this information is true or false. Also, after she has contacted the poster, she might go somewhere else with this potential flat mate to find accommodations. This can happen at night and Alice feels that it is dangerous to meet with a stranger in an unknown location. So she signs onto the SSU portal site and applies for a MobiPass for the AFS. To ensure the level of security, Alice must hold her public/private key pair before applying for the MobiPass from the SSU. Once she has signed in, she would found that most information about her has already been filled and cannot be modified; only the self explanation and descriptions are left for her to fill. At the same time, Alice's public key for the AFS is uploaded to the SSU. Once submitting the form and reviewed by the SSU staffs, Alice will receive a MobiPass for the AFS from SSU. The message snippet is shown in Figure 3.

```
<MobiPass>
  <meta>
    <digestValue>RjzP...DGY8=</digestValue>
    <signatureValue>=</signatureValue>
  </meta>
  <certified>
     <expired>2007-08-05</expired>
    <issuer>
      <ECA>
        <ECA-ID>124..626</ECA-ID>
          <ECA-name>AFS, Univ of Techo, Sydney </ECA-name>
<publicKey>https://ssu.mobipass.uts.edu.au/afs.pub.key</publicKey>
      </ECA>
       <policy>
          <policy-ID>11...34</policy-ID>
          <description>....</description>

<schemaLocation>https://ssu.mobipass.uts.edu.au/afs.schema.xsd</schemaLocation>
      </policy>
    </issuer>
    <studentInfo>
      <publicKeyOfHolder>Daz==.z==</publicKeyOfHolder>
      <gender>female</gender>
      <age-range>20-25</age-range>
      <department>computer science</department>
      <faculty>information technology</faculty>
       <nationality>Chinese</nationality>
          …
     </studentInfo>
  </certified>
  <nonCertified>
     <selfDescription><![CDATA[… easy going, nice person!.... ]]></selfDescription>
<interests>
        <element>fishing</element>
        <element>reading</element>
        <element>...</element>
      </interests>
<smoker>false<smoker>
<hasPet>false<hasPet>
  </nonCertified>
  <timestamp>
```

```
      <notBefore>1132622517640</notBefore>
      <notAfter>1132622519640</notAfter>
      <timestampSignatureValue>skz...==z</timestampSignatureValue>
   </timestamp>
</MobiPass>
```

Figure 3: MobiPass Message Snippet for AFS

After receiving this AFS MobiPass, Alice imports this MobiPass to her mobile phone and tries to setup the service correctly. We can assume that Alice already has SSU's public key in her mobile phone, and the MobiPolicy for this AFS has been downloaded onto her mobile phone. So Alice runs her MobPolicy setup to load her AFS service, and fill in all other criteria for this service. Such as:

- Gender: Female
- Flat mate Age Range: 20-25
- Nationality: {Chinese, Korean, Japanese, Australian}
- Flat mate Major: Accounting, Business, Music
- List of Suburbs, which are within walking distance to the university
- Monthly rental budget, for Alice, the limit is $150 per week.
- Furnish – fully furnished etc

The service is activated once the setup is finished. So when Alice walks into the campus, her mobile phone will try to find an AFS LBs to run the service. For example, the AFS LB reception might cover the central common areas within the university, and the LB will run the service in ALC mode. This means that when Alice enters the campus, the AFS service will be up and running on her mobile phone, then her MobiPass application will try to contact the LB for the AFS service. When Alice's mobile device has found the SSU's LB, it will then try to authenticate the LB; initiate a handshake with the local SSU for exchanging the symmetric key. As the local SSU is running in ALC mode, the SSU's LB will also be asking for Bob's AFS service settings, so Alice's mobile device will send his settings to the SSU's LB. The SSU's LB will then act on Alice's behalf and announces to the entire wireless network that a new member has joined the network and this new member is looking for a flat mate. After receiving the message, students who are using the same service might try to contact Alice by their MobiPass through the LB, and the SSU LB will try to authenticate the incoming MobiPass for Alice, i.e, whether this sender has a valid MobiPass e.g. a student from the university. If the MobiPass is valid, the LB will try to match their profiles and if it matches Alice’s criteria, she will receive a notification that there are people around who are interested to share an accommodation with her.

This notification by her MobiPass-enabled device will allow Alice to meet with potential matching students within a very short time in the university common area, so that they can speak face-to-face, therefore allowing Alice to make a final decision on whether the potential student is a match. In this way, trusted interaction for mobile devices supported by the MobiPass architecture can provide greater immediacy and functionality than other electronic interactions.

This case study has described the steps taken in applying the LB based MobiPass architecture to the AFS, the MobiPass architecture, in this example, provides an excellent platform for university students to find their accommodation and flat mates securely and efficiently. By using the MobiPass architecture, students can very easily distinguish potential flat mates. The scope of potential flat mates is limited to university students with matching profiles. Also, the communication can happen in real time and once they find each other by mobile phone, they can start meeting immediately, such as in the university's common area where the MobiPass-based interaction initially occurs. Students are not required to make appointments and meet somewhere which might be unfamiliar and potentially dangerous to them. Also the architecture is very open and flexible; it is easy to apply this architecture to a more mission critical service to ensure that interactions will take place in a trusted manner.

5 Related Work

This research outcome is based on our previous research on ubiquitous and mobile computing, the MobiPass architecture (Steele, Tao 2006, Tao, Steele 2006). The goal of this research is to provide a highly effective approach to build a trusted interaction between different entities within an open and dynamic environment. There are also other researchers that have focused their efforts in addressing this issue.

Kagal, Finin and Joshi (Kagal, Finin, Joshi 2002) proposed the Centaurus system which provides a fine grained access control in their Smart Office ubiquitous computing scenario, the system utilises the distributed trust approach and extends the Role Based Access Control(RBAC) to allow foreign users from another security domain to be granted the proper privileges in order to gets access. Based on their implementation, Hong and Landay (2004) propose the architecture to perform the authentication and authorisation in ubiquitous computing by assigning tags to pieces of information; information is associated with a policy and indicated by the tag. Park and Sandhu (1999) proposed a concept named smart certificate for improving scalability in web servers, which has some interest to our work. The smart certificate is an extended version of X.509 certificate with several remarkable features. These previous research works are more focused on authentication which can help interacting entities to identify the transacting parties; however, fine grained access control is not covered comprehensively, which, is a very important aspect in mobile computing.

6 Future Work and Conclusion

To build a trusted environment in mobile computing, the MobiPass architecture is introduced to allow mobile devices to be recognised by only presenting their MobiPass and also it allows one entity to judge other entities by examining their respective MobiPasses. However, for performance considerations, a variant that introduces the LB to reduce the load for each mobile device has been introduced in this paper. The LB enabled MobiPass architecture works under the condition that a group of mobile devices do not know each other, but they all have a solid relationship with the LB. According to our research, there are a large number of scenarios in which the device has such good knowledge of the LB. Our future work will focus on extending the LB enabled MobiPass architecture to enable arbitrary MobiPass processing in real time, i.e. do not need to have manually imported the ECA's public key, and proposing a good mechanism for allowing to reuse a MobiPolicy by multiple service providers. Efforts will also be made to improve our service discovery protocol, i.e., how different entities can discover each other and how to negotiate and send the MobiPass during the discovery. Our ongoing research will be to refine the MobiPolicy and to make the MobiPass architecture generic enough to be pluggable into most trustworthiness mobile systems.

References

Steele, R., Tao, W. (2006) MobiPass: A Passport for Mobile Business. *Personal and Ubiquitous Computing*, Springer, 11 (3): pp.157-169

Tao, W., Steele, R. (2006) Mobile Trust Via Extended Digital Certificates, *Proc of The 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC 06)*, pp. 284-292.

Satyanarayanan, M. (2001), Pervasive computing: vision and challenges, *IEEE Wireless Communications, 8 (4)*, 10-17

Ranganathan, K. (2004), Trustworthy Pervasive Computing: The Hard Security Problems, *Proc of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops* 119

Diffie, W. (1998), The first ten years of public-key cryptography, *Proc of IEEE, 76 (5)*, 560- 577

Steele R, Gardner W, Rajugan R, Dillon TS (2005) A design methodology for user access control (UAC) middleware. *Proc of the 2005 IEEE international conference on e-technology, e-commerce and e-service (EEE'05)*, pp 385–390

Kagal L, Finin T, Joshi A (2002) A security architecture based on trust management for pervasive computing systems". *Grace Hopper Celebration of Women in Computing*, October 2002

Hong J, Landay J (2004) An architecture for privacy-sensitive ubiquitous computing" *Proc of the 2nd international conference on mobile systems, applications, and services*, pp 177–189

Park J.S, Sandhu R (1999) RBAC on the Web by smart certificates *Procof the 4th ACM workshop on role-based access control. ACM Press*, New York, pp 1–9