Hierarchical trustworthy authentication for pervasive computing

Hierarchical Trustworthy Authentication for Pervasive Computing

Hannan Xiao*, James A. Malcolm*, Bruce Christianson*, and Ying Zhang†
*School of Computer Science, University of Hertfordshire
College Lane, Hatfield, AL10 9AB, UK
Emails: h.xiao, j.a.malcolm, b.christianson@herts.ac.uk
†Department of Engineering, University of Cambridge, Cambridge, CB3 0FA, UK
Email: yz282@cam.ac.uk

***Abstract*— Conventional entity authentication is not enough to build a secure pervasive computing environment. Being sure that you are talking to the expected entity does not guarantee it is going to do what you expect him to do, and only that. This paper introduces a concept of "trustworthy authentication" in pervasive computing which is defined as entity authentication accompanied by an assurance of trustworthy behaviour of the authenticated entity. It discusses how to provide trustworthy authentication in pervasive computing using the example of a roaming customer wishing to print his email on a public printer. A two-level hierarchical trustworthy authentication scheme is proposed where local and higher-level authorization servers issue trustworthiness certificates after receiving trustworthiness records from the printer, signed by its users. The proposed scheme may be generalized for trustworthy authentication of security devices such as firewalls.**

I. Introduction

Pervasive (or ubiquitous) computing presents a world with various kinds of digital devices embedded with certain "intelligence", connected with one another in an unobtrusive way, despite the roaming of some devices with wireless transmission capacities. The normal means of authentication have to be re-examined when applied to the environment of pervasive computing. Pioneering work has been made on solving the problem of lacking a centralized online authentication server to issue tickets as in Kerberos or certificates as in public key infrastructure (PKI), e.g., the "resurrecting ducking" [4], [5]. However, Creese *et al.* argue that the current notions of entity authentication are unsuitable for the pervasive domain [3]. Taking the example of a user wishing to print his email on a unknown public wireless printer, the security assurances achieved by entity authentication that the user is likely to want are:

- The confidential data of the email only goes from the user's PDA to the specific printer chosen by the user, and no other.
- The confidential data is treated by the printer in a "trustworthy" way. Trustworthy here means that the printer ensures that no other party has access to the data while it is resident on the printer and that the printer itself will not use the data in a malicious way (*e.g.*, only pretending to delete the data after finishing printing).

The first requirement might be provided by the conventional concept of entity authentication after being specially tailored for the features of pervasive computing. The second requirement is the concern of this paper. How to ensure that the printer will treat the data in a trustworthy way? As pointed out by Creese *et al.* [3], this type of security assurance seems to fall outside of the scope of traditional entity authentication. Alice is able to authenticate Bob from another company who she never meets; but does Alice believe that Bob will treat the confidential document of her company in a trustworthy manner if she passes it to him? This requirement is common in the pervasive computing domain where entities are mobile, connections are dynamic, and security associations are transient. In this paper we define the entity authentication accompanied by a trustworthy behaviour of the authenticated entity as "trustworthy authentication".

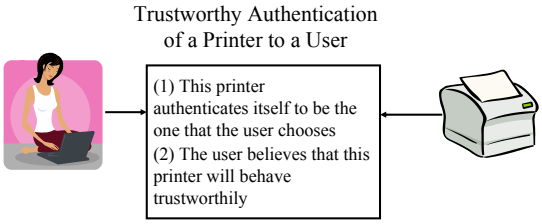

Fig. 1. Definition of Trustworthy Authentication

II. Definition of Trustworthy Authentication

We unpack "trustworthy authentication" in the example above, "a wireless printer provides trustworthy authentication to a user who wishes to print his email on the printer". It means two things (see Fig. 1):

- The printer is able to authenticate itself to the user that it is the printer the user chooses, and
- The user believes that the printer is trustworthy in following communications.

The first part means entity authentication, but may be done in an unconventional way. The second part is a statement of fact on the trustworthiness of the printer – does the printer protect the user's data well, use the user's data in good faith, and always have good printing service, *etc*? The definition of trustworthy authentication of entities means that an entity not only authenticates him to another entity, but also assures the other entity that he is trustworthy in the following communications.

III. A Two-Level Hierarchical Trustworthy Authentication for Pervasive Computing

A. Assumptions

Firstly we assume the availability of a well-known public service/server which provides the basic information such as where are the nearby printers in a similar way that a local council service provides childcare information. We also assume that public printers are reasonably static, *i.e.*, not as mobile as roaming customers, and do not change locations frequently. Secondly we assume an existing PKI in the Internet for the use of pervasive computing and the public keys of authorization servers that will be introduced later are published on public servers too. Thirdly we assume that users are happy to leave a comment on the trustworthiness of the service at the end of their printing job[1].

Imagine roaming customer Bob gets the location information of public printers from a public server and approaches the one nearest to him. Bob's PDA and the printer exchange their public keys via a location-limited channel (*e.g.* Infrared) as suggested in [1]. By direct physical contact at the location-limited channel, Bob is sure that his PDA is communicating with the printer he chooses – *i.e.*, the first requirement in "trustworthy authentication". Next, how does the printer convince Bob that it is trustworthy?

B. Bootstrapping

If Bob sees a brand-new printer with no trustworthiness records, he may believe that the printer is trustworthy because it is made by a famous printer-company such as HP, and/or located in a Starbucks Cafe where the user considers as a safe place, and/or both the user's PDA and the printer are configured using Microsoft software. This sounds like a trend towards monoculture; however, it may be inevitable as we have to use some trustworthiness bootstrapping. This kind of bootstrapping based on common sense can also be used if the customer is suspicious of the trustworthiness records presented by the printer, since he can always retreat to this initial judgement of trustworthiness.

C. Trustworthiness Record

If customer Alice is happy to leave a comment on the service at the end of printing, she writes a trustworthiness record as follows:

$$E_{K^+_{localAS}}(ID_P||K^+_P||M||Timestamp||ID_{CS}) \\ ||E_{K^-_A}(H(M))||E_{K^-_{CS}}(T||ID_A||K^+_A) \quad (1)$$

where the notations are:

- ID_P, K^+_p: the printer's identity registered at the local AS, and the printer's public key. These information is exchanged between Alice's PDA and the printer via the location-limited side channel [1].
- M: the comments of Alice.
- $Timestamp$: the time when the record is generated.
- ID_{CS}: the identity of a public key certification server (CS) at Alice's home domain.

This data is encrypted by $K^+_{localAS}$, the public key of a local authorization server (AS). Alice gets the key from the local notice board or tourist information center. The trustworthiness record proper is followed by:

- $E_{K^-_A}(H(M))$: Alice's signature of her comments.
- $E_{K^-_{CS}}(T||ID_A||K^+_A)$: the key certificate issued to Alice by a public key CS.

Alice writes the comments about the printer (which is identified by its ID and public key), notes the time when writing her comments, and includes the identity of her certification server. Alice then encrypts the above message by using the public key of a local AS. This is to guarantee the confidentiality of the message so that the printer is not able to know whether it has good comments or bad comments from a customer. Alice also signs the comments, and attaches a copy of her public key certificate issued by a public key CS at her own domain. We call (1) a trustworthiness record. It tells the local AS: "I here put this comment (M) about the printer (ID_p, K^+_P) at this time ($Timestamp$). Here is my signature and you will know who I am from the public key certificate issued by the certification server (ID_{CS}).

The trustworthiness record is forwarded from the printer to the local AS, rather than directly sent by Alice. This is because we want to put minimized requirements on customers and it benefits the printer to pass the trustworthiness records to the local AS in order to obtain a trustworthiness certificate. The local AS decrypts the first cyphertext by applying his private key and sees the comments about the printer. The local AS also gets the ID of the public key CS of Alice, by which it finds the CS's public key from its local database or somewhere else. Next, by using the public key of CS, the local AS obtains the public key of Alice from the public key certificate. Finally, the local AS uses Alice's public key to verify her signature.

D. Local Trustworthy Certificate

After the local AS receives certain number of trustworthiness records of the printer during a certain period, it issues a trustworthiness certificate as:

$$TWGrade||ID_P||ID_{localAS}||Timestamp \\ ||E_{K^-_{localAS}}(H(TWGrade)||ID_P||Timestamp) \quad (2)$$

where $TWGrade$ is the trustworthiness grade given to the printer by the local AS. $E_{K^-_{localAS}}[H(TWGrade)||ID_P||Timestamp]$ is the signature of the local AS. The use of timestamp prevents the printer from using an old certificate of higher TWGrade since the customer will consider it less valuable than a more recent good grade certificate.

The trustworthiness certificate in (2) is much less complicated than the trustworthiness record in (1). It simply says: "Here

[1]We are aware that this assumption is strong, since some users may want to stay anonymous. But users can be authenticated as a member of group (faculty staff) without revealing who they are - *e.g.* using ring signature [2].

I, a local authorization server ($ID_{localAS}$), give this trustworthiness grade ($TWGrade$) to the printer (ID_P) at this time ($Timestamp$). Here is my signature."

The printer may collect trustworthy certificates from more than one local AS, possibly because of the maintenance needed by local ASs or the movement of printer although we assume that public printers do not have frequent movement.

E. Higher level Trustworthy Certificate

After a printer collects several consistently good trustworthiness certificate from a local AS, the printer sends these certificates to a higher level AS, and exchanges them for a trustworthiness certificate signed by it:

$$TWGrade||ID_P||ID_{higherAS}||Timestamp \\ ||E_{K^-_{higherAS}}(H(TWGrade)||ID_P||Timestamp) \quad (3)$$

Here we assume that the user has obtained the public key of the higher level AS from somewhere first. In pervasive computing, a domain usually contains more local ASs and just few higher level ASs. A higher AS is able to see a lot of trustworthiness certificates issued by many local ASs; and it applies much stricter criteria of issuing a certificate than a local AS. Thereby a user considers a printer with such a certificate more trustworthy than the one with just a certificate from a local AS. The user is more likely to recognise the higher level AS, and also obtains a more balanced view of the trustworthiness of the printer from a higher level AS than from a single local AS.

F. Typical Procedure

Figure 2 shows a typical procedure of the proposed solution for trustworthy authentication in pervasive computing in the example of roaming users using a public printer.

1) Alice gets the public keys of a local AS and a higher level AS from a local notice board.
2) Alice's PDA and the printer exchange their public keys via a location-limited channel, or a secret password firstly via the location limited channel, and then the public keys via wireless connection.
3) Alice prints her job, leaves trustworthiness record about the printer, by wireless communication.
4) The printer forwards the comments from various customers to the local AS.
5) The local AS issues a trustworthiness certificate to the printer.
6) The printer may decide to forward the trustworthiness certificates from the local AS to a higher level AS.
7) The higher level AS issues a trustworthiness certificate to the printer.
8) The printer presents the trustworthiness certificates to a new customer Bob.

IV. Related Work, Open Questions & Summary

Related works are in three major areas: authentication in pervasive computing, trust establishment in pervasive computing, and trust authentication in enterprise digital rights management (DRM). When going to the detail of the scheme, there are

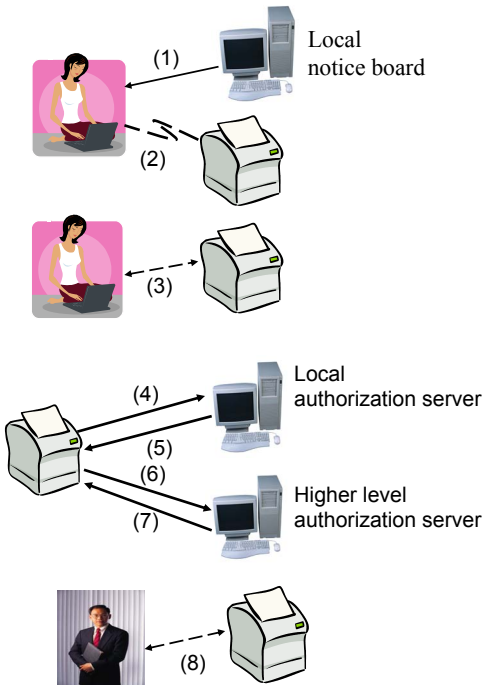

Fig. 2. Two-level Hierarchical Trustworthy Authentication for Pervasive Computing

obvious questions. How does the user know that no other party has access to his data while the email is being printed? How does the user know that the printer itself does not save a copy of the data for future use? And what is the threat model of the proposed scheme?

In summary, this paper puts forward a concept of "trustworthy authentication" in pervasive computing which is defined as entity authentication accompanied by a trustworthy behaviour of the authenticated entity and a two-level hierarchy of authorization servers is proposed. The proposed scheme may be generalized for trustworthy authentication of security devices such as firewall and IDS. In the future, further attack analysis on the proposed trustworthy authentication structure will be carried out. It remains an open question whether one can be assured of correct behaviour without authenticating the participants at all.

References

[1] D. Balfanz, D. K. Smetters, P. Stewart, and H. Chi Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proceedings of Network and Distributed Systems Security Symposium*, 2002.

[2] P. Das Chowdhury and Bruce Christianson. Uncorrelatable electronic transactions using ring signatures. In *Proceedings of the Wholes Workshop of Multiple Views of Privacy*, Sweden, 2004. Swedish Institute of Computer Science.

[3] S. Creese, M. Goldsmith, B. Roscoe, and I. Zakiuddin. Authentication for pervasive computing. In *Proceedings of the 1st International Conference on Security in Pervasive Computing, IEEE Computer Society Press, 2003*, pages 116–129, 2003.

[4] F. Stajano. The resurrecting duckling - what next? In *Proceedings of 8th Security Protocols Workshop, Lecture Notes in Computer Science 1361*, pages 204–214, Berlin, 2001. Springer-Verlag.

[5] F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues in ad-hoc wireless networks. In *Proceedings of 7th Security Protocols Workshop, Lecture Notes in Computer Science 1796*, pages 172–182, Berlin, 2000. Springer-Verlag.