A Capability-based Privacy-preserving Scheme for Pervasive Computing Environments

Divyan M. Konidala [†], Dang N. Duc [†], Dongman Lee [‡], Kwangjo Kim [†]
Information and Communications University (ICU),
[†]International Research Center for Information Security,
[‡]Collaborative Distributed Systems and Networks Lab,
103-6, Munji-Dong, Daejeon 305-714, Republic of Korea.
{divyan, nguyenduc, dlee, kkj}@icu.ac.kr

Abstract

In a pervasive computing environment, users interact with many smart devices or service providers (SPs) to obtain some useful services from them. These SPs can be either genuine or malicious. As a result, users privacy is at a greater risk, as they are prone to revealing their location, identity and transactions information to such SPs. On the other hand, user authentication is also required for SPs to provide service access control to only authorized users. In order to protect users privacy, they must be allowed to have anonymous interactions with SPs. But, authenticating and authorizing an anonymous user becomes a challenging task. In this paper, we propose a simple and efficient scheme that allows users to anonymously interact with SPs and the SPs can effectively authenticate and authorize the users based on the anonymous information submitted by the users.

1. Introduction

Among many security requirements of a pervasive computing environment (PCE), this paper focuses on user authentication, authorization, service access control and privacy protection. Only when the user is authenticated and authorized, the SP grants him the services. *e.g.*, the access rights of staff, manager and president are different. This concept is popularly known as “Role-based Access Control(RBAC)”. In the environments with significant concentration of “invisible” computing devices gathering and collecting users identities, their location and transactions information, the user should rightly be concerned for their privacy. This personal information could allow SPs and eavesdroppers to generate detailed profiles of the user, his buying interests, and trace all his actions.

In most PCE, it is desirable that the user interacts anonymously with the SPs or other smart devices. But the catch is, if the user is not revealing his real identity to the SP, how the SP can trust the user to be genuine and check whether he is allowed to access that particular service. Our scheme focuses on resolving this conflicting nature of user authentication, authorization and privacy protection. Our simple and efficient concrete protocol design provides capability/credential based anonymous user authentication and authorization in PCE. It provides user anonymity and the SPs would still be able to authenticate, authorize and provide service access control based on the anonymous information submitted by users. Our scheme assumes minimum amount of trust on the admin server, which issues the capabilities to the users. The user’s service transaction details are hidden from the admin server and even if the SP and the admin server maliciously collude, the real identity of the user is never revealed, further protecting users privacy.

This paper is divided into the following sections: Section 2 briefly describes the background information needed to understand our scheme. Section 3 provides detailed description of our proposed scheme including design considerations, system architecture, and protocol description. Section 4 provides the security analysis of our scheme. Section 5 illustrates the complexity analysis, and Section 6 compares our scheme with related work. Section 7 concludes this paper.

2. Background

Capability-based User Authentication and Authorization: A capability is something that can be used to prove who you are, or prove that you are authorized to do something. It allows RBAC. The capabilities are issued to the user by an admin/Authorized Server (AS), which is trusted both by the user and the SP. The AS issues capabilities to the user depending on his role in the PCE, *e.g.*,

Proceedings of the 3rd Int'l Conf. on Pervasive Computing and Communications Workshops (PerCom 2005 Workshops)
0-7695-2300-5/05 $20.00 © 2005 IEEE

student, professor, staff, manager, visitor, *etc*. Capability is digitally signed by the AS and it is unforgeable. The user should present his capability to a SP whenever he wants to access that service. The SP verifies the AS signature on the capability and accordingly accepts or denies service access control to the user.

Partially Blind Signature: Abe and Okamoto [1] proposed the idea of partially blind signature with security proofs. A partially blind signature scheme is an extension of an ordinary blind signature scheme [5]. It has two portions, one portion consists of the message that is blinded by the user (from the signer) and in the other portion, the signer can explicitly embed some mutually agreed information such as issuing date, expiry date, signer's identity, *etc*.

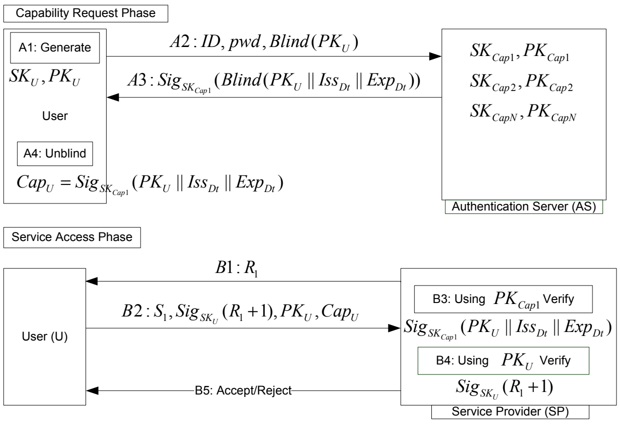
3. Proposed Scheme

3.1. Design Considerations

Our scheme provides user authentication, authorization and privacy protection at the application layer. User devices can still be traced at the link layer via MAC or IP addresses. Approaches like mix-networking [4] address the link layer anonymity. We assume that every user has access rights to more than five services and not just one service. To maintain clarity our protocol does not include some of the trivial and basic data security mechanisms like data confidentiality and integrity.

3.2. Protocol

Our proposed scheme includes three entities: Authorized server (AS), User and his portable mobile device (U), and Service Provider (SP). Generally a PCE or parts of a huge PCE are managed by a AS. AS issues capabilities to the user depending on his role in the environment.



Capability Request Phase
A1: Generate
SK_U, PK_U
User
A4: Unblind
$Cap_U = Sig_{SK_{Cap1}}(PK_U \| Iss_{Dt} \| Exp_{Dt})$
$A2: ID, pwd, Blind(PK_U)$
$A3: Sig_{SK_{Cap1}}(Blind(PK_U \| Iss_{Dt} \| Exp_{Dt}))$
SK_{Cap1}, PK_{Cap1}
SK_{Cap2}, PK_{Cap2}
SK_{CapN}, PK_{CapN}
Authentication Server (AS)
Service Access Phase
User (U)
$B1: R_1$
$B2: S_1, Sig_{SK_U}(R_1+1), PK_U, Cap_U$
B5: Accept/Reject
B3: Using PK_{Cap1} Verify
$Sig_{SK_{Cap1}}(PK_U \| Iss_{Dt} \| Exp_{Dt})$
B4: Using PK_U Verify
$Sig_{SK_U}(R_1+1)$
Service Provider (SP)

Figure 1. System architecture

As shown in Fig. 1, our proposed scheme involves two phases, "Capability Issuing phase" and "Service Access Phase". Initially we assume that a user "Alice" personally registers with the AS by producing her true ID (social security number, student ID, employee ID, *etc*.), a password of her choice, officially certified documents (from school administrative dept., company's human resource dept., *etc*.), which prove her true ID. AS after verifying the original documents, identifies Alice with a particular role in the PCE. The AS then stores its public key PK_{AS} in Alice's mobile device. This is done, so that Alice can make use of PK_{AS} to verify future digitally signed communications from the AS and to secure the communications between them. The above registration procedure can be carried out securely via online, or offline.

The AS stores a set of public and secret-key pairs of different versions of capabilities. *E.g.*, SK_{cap1} and PK_{cap1} are the secret and public key pair of capability 1 ($Cap1$) respectively. $Cap1$ includes permission to access services with service IDs $\{S_1, S_3, S_6, S_{12}\}$ and $Cap1$ may be suitable to all the students in the campus. Similarly SK_{cap2} and PK_{cap2} represent capability 2 $Cap2$, which includes permission to access services IDs $\{S_1, S_6, S_{28}, S_{30}\}$. $Cap2$ may be suitable to all the managers in the office. AS must have already distributed PK_{Cap1} of $Cap1$ to all the SPs providing services $\{S_1, S_3, S_6, S_{12}\}$.

3.2.1 Capability Issuing Phase

One fine day, after the registration procedure, Alice decides to make use of certain services in the PCE. In order to do so, she needs to have her own "capability" issued by the AS. As a result, Alice undergoes the following steps to get a capability from the AS.

Step A1: In this phase, Alice generates a public-key PK_U and secret-key SK_U pair. Alice's mobile device can generate public and secret-key pair well in advance (during the idle time) and store in the memory.

Step A2: Alice securely logs-into PCE by sending her real ID, password pwd to the AS. AS verifies the ID and pwd, If correct, it retrieves the role of the ID in the organization. Consider Alice is a student, the AS replies Alice with PK_{Cap1} of $Cap1$. Utilizing PK_{Cap1} and a partially blind signature scheme, Alice blinds PK_U as $Blind(PK_U)$. PK_U is Alice's public-key "value" for temporary use only, and not a publicly verifiable certificate having user's ID (as in PKI). Alice sends $Blind(PK_U)$ to the AS, thus requesting for a capability to be issued.

Step A3: The AS signs $Blind(PK_U)$ with SK_{Cap1} of $Cap1$, then Alice, who happens to be a student, can access the services $\{S_1, S_3, S_6, S_{12}\}$. While signing $Blind(PK_U)$, the AS utilizes partially blind signature scheme to embed some essential information like Capability Issue Date (Iss_{Dt}) and Capability Expiry Date (Exp_{Dt}) into the signed message.

Step A4: Alice receives A3 and verifies the signature on message A3 using PK_{AS}. Since PK_U is blinded by the user, the AS does not know the value of PK_U. User un-blinds A3 to obtain her capability $Cap_U = Sig_{SK_{Cap1}}(PK_U||Iss_{Dt}||Exp_{Dt})$. For security reasons the capability is issued for a day. After the capability expires, Alice has to restart this capability issuing phase with a new public and secret-key pair.

3.2.2 Service Access Phase

After receiving a capability Cap_U from the AS. Alice undergoes the following steps to get services from the SPs.

Step B1: Using her mobile device, Alice requests for service S_1. A SP in charge of providing S_1 generates a unique random number R_1 and sends it to Alice's mobile device. To keep the explanation of our proposed scheme to-the-point we avoid the details of Alice authenticating SP via AS issued digital certificate.

Step B2: Alice digitally signs $R_1 + 1$ using SK_U. Alice then sends $S_1, Sig_{SK_U}(R_1+1), PK_U, Cap_U$ to the SP. Where: S_1 is the service ID for which Alice wants to obtain access control from the SP. As mentioned before, neither PK_U nor Cap_U contain any information that can be used to expose the true identity of Alice.

Step B3: The SP receives the message B2 from Alice. It first retrieves PK_{Cap1} from its database and verifies the signature of AS on Alice's capability Cap_U. If satisfied it proceeds to check if PK_U sent in open equals PK_U in Cap_U. Later it verifies Exp_{Dt}. If the SP does not posses PK_{Cap1}, then it is not entitled to provide the services $\{S_1, S_3, S_6, S_{12}\}$ of Cap_1. Such is the case, Alice is duly directed to another SP.

Step B4: Using PK_U sent in open, the SP then verifies the signature: $Sig_{SK_U}(R_1 + 1)$. If successful, the SP realizes that only the user whose PK_U is signed by the AS can only correctly sign $R_1 + 1$ using his SK_U.

Step B5: Thus the SP without knowing the real identity of the user, concludes that this particular user has the capability Cap_U issued by the AS and hence is authorized to access the service S_1. If any one of the above checks fail, Alice is denied access to S_1.

4. Security Analysis

User Privacy Protection

In step A4 of capability issuing phase, Alice's capability Cap_U does not contain her real ID. Therefore the SP does not know the real ID of Alice. On the other hand the AS does not know what services the user has accessed, because the SP and the AS never communicate with each other during the service access phase. This provides complete anonymity and privacy to Alice.

Since the SP receives Alice's PK_U, Alice can still be tracked with her PK_U usage. But the real identity of Alice is never revealed, because PK_U acts as a pseudonym for Alice. And also the Time to Live value of Alice's capability is for a day. So the SPs receive different PK_U's from the same user on a daily basis. The more frequently PK_U and SK_U are changed, the better the anonymity/privacy, but this induces high computational overhead on Alice's mobile device. It is a trade-off issue between perfect unlinkable anonymity and performance degradation. However, capabilities issued for only one day are reasonably secure, and provide required level of partial un-linkable anonymity/privacy.

Our scheme also prevents the SP and the AS from maliciously colluding with each other in order to reveal the transactions of the user and expose his privacy. The SP receives PK_U from Alice via message B2, it can send PK_U to the AS hoping to obtain the real ID of Alice. However, at the AS end, there is no match between the real ID of the user and his/her PK_U. Because in message A1 of capability issuing phase, PK_U is blinded as $Blind(PK_U)$, the AS never knows the value of PK_U. SP may try to correlate the capability's dates of issue and expiry by collaborating with AS. So these ought to be distinct dates.

User Authentication, Authorization, Access Control

Via steps B3 to B5 of the service access phase, it can be noticed that without using the real ID of Alice, she is effectively authenticated, authorized and provided/denied service access control.

In Step B1 of the service access phase, it can be noticed that the SP sends a unique random number R_1 to Alice's mobile device. Even though Alice's real ID is never included in her capability Cap_U, still the capability can be anonymously linked to Alice as follows: In message B2, only Alice using her SK_U can correctly sign on $(R_1 + 1)$. $Sig_{SK_U}(R_1 + 1)$ can eventually be verified by the SP using only Alice's PK_U. PK_U is included in Alice's capability Cap_U. And finally, Cap_U is digitally signed by the AS. Even if an adversary captures message B2: ($S_1, Sig_{SK_U}(R_1+1), PK_U, Cap_U$), he cannot impersonate Alice as he does not know SK_U.

Replay Attack Detection

Even if an adversary captures message B2, he cannot mount replay attack. Because during another service access phase the SP sends an unique random number say R_3 and the captured message B2, does not contain R_3.

Capability Non-transferability

Our scheme discourages Alice to transfer her capability to another user say "Bob". In step B2 of the service access phase, only Alice can correctly produce $Sig_{SK_U}(R_1 + 1)$ using SK_U. As a result, if Alice wants to transfer her capability to Bob, then Alice should give away her secret key to Bob, but Bob may misuse it. And Alice's capability may include access rights to her very personal services including some financial services. It is also possible that Alice may try to request the same capability twice, one for himself and another for his friend Bob. To prevent such malpractice, we can introduce a policy to issue only one capability a day for a particular user. This can be verified by the issue date Iss_{Dt} included in the capability.

5 Complexity Analysis

In this section, we analyze the proposed scheme in terms of storage requirement and execution time. In a PCE, we are mostly interested in storage, computational and execution time overhead at user's end because they generally carry low-computing and resource-poor portable devices like mobile phones, and PDAs.

The heart of our scheme is the Abe-Okamoto's partially blind signature (PBS) scheme, whose security is based on the intractability of discrete-logarithm problem. So we can use Elliptic Curve (EC) group here. In the signature issuing protocol of the Abe-Okatomo scheme, the detail computational complexity of two parties, user and server are as follows: Server: 3 scalar multiplications a one point addition. User: 4 scalar multiplications and 4 point additions.

The message to be signed is $(PK_U||Iss_{Dt}||Exp_{Dt})$. This message together with server's signature form a capability Cap_U. Iss_{Dt}, and Exp_{Dt} each consume 7 bytes. If we use 163-bit long secret key for PBS scheme, then the signature length is 4*163 bits. Also in the capability issuing phase, users have to generate a public-key and secret-key pair, we strongly recommend Elliptic Curve Cryptography (ECC) as the most suitable approach to implement this, because it provides small key size and faster execution than other public-key cryptosystems. A 80-bit key is sufficient in ECC for cost effective short term security. Therefore the size of a single capability is given by: length of the signature + length of PK_U + length of SK_U + $2 * 7 * 8$ bits, which equals to $4 * 163 + 80 + 80 + 2 * 7 * 8 = 924$ bits. This indicates low storage requirement to store a capability in the user's mobile device.

To illustrate how well the capability issuing protocol perform, we show here some implementation results of primitive operations (scalar multiplication and point addition) on ECC using embedded platforms such as Palm Pilot and SmartCard. As we know, these primitive operations are the most computationally expensive operations in cryptographic schemes based on EC group. They dominate other operations like hash computation, modular addition, *etc.* in terms of time required to complete. Table 1 shows time taken to carry out primitive operations on EC group [11, 12, 13, 7, 8]. From this table we can infer that our scheme has tolerable execution time.

Table 1. Performance Evaluation

	Siemens Crypto Smartcard *		Palm V Dragonball **	
	Time	Remark	Time	Remark
Point Add. on EC	39.56 ms		NA	
Scalar Multi. on EC	1830 ms	Pre-compute	0.79 s.	Pre-compute
ECDSA Key Gen.	NA		514 ms.	163-bit
ECDSA Sig. Gen.	185 ms	135-bit sig.	713 ms.	163-bit sig.
ECDSA Sig. Verif.	360 ms	135-bit sig.	1740 ms.	163-bit sig.

* 5MHz SLE66CX160S, **16.6 MHz

As in PBS, on the user side, it is required to execute four scalar multiplications and four point additions while on the server side, it is required to execute three scalar multiplications and one point addition. So if user uses a Palm Pilot device, it will take him about 4 seconds to finish all computations to get one capability. This is a reasonable performance for user satisfactory. User's mobile device can generate ECC-based public-key and secret-key pair in advance (during idle time) and store in the memory.

For service accessing phase, we are interested in how well signature generation and verification operations perform. The above table also shows timing information for such operations with respect to Digital Signature Algorithm (DSA) based on elliptic curve group (EC-DSA). The performance of these operations is clearly far better than one session of the capability issuing protocol.

6 Comparison With Related Work

Identity Management Approach

Identity Management is well described in [10]. In this approach users interact with other smart devices through pseudonyms or Virtual Identities (VID). [9] describes the drawbacks of this method. This approach is not user friendly as it involves many pre-settings and creates burden on the user's mobile device. Our scheme provides user anonymity, authentication and authorization with out using many pseudonyms (only one PK_U per day). User's involvement is also very low and it provides user authorization.

Pseudonym Systems

In a pseudonym system [6, 3], a user interacts with multiple organizations in an anonymous manner using some unlinkable pseudonyms. This approach employs high computationally complex number-theoretic operations at the user's end. Our scheme induces less computational burden and execution time at the user's end, appropriate for PCE's low computing devices. Also the above schemes consist of very specific protocols whereas our proposed scheme use cryptographic schemes in black-box manner thus can use any schemes available.

Mix-Network

A mix net [4] consists of several servers, called mixes, which produces a batch of output messages in a permuted (mixed) order. Al-Muhtadi *et al.* [2] proposed Routing through Mist Routers protects authorized users' location privacy. But users have to trust a "Lighthouse". The Lighthouse keeps all information of users registered with it. It also assumes high degree of trust in the mix network. Our scheme is simple, and the computational complexity and execution time at the user's end is also very low. Our scheme assumes minimum amount of trust on the AS, and even if the SP and the AS maliciously collude, the real identity of the user is never revealed,

7 Conclusion

Our scheme can be easily ported on to a public space or large scale PCE, for example airports, train stations, streets, highways, *etc*. Nowadays, in our society we can experience the ubiquitous presence of credit card companies and mobile operators. Consumers and SPs have already put in a great deal of trust in such big organizations. Therefore these organizations can take up the role of AS mentioned in our scheme and issue capabilities to their esteemed customers. In PCE, such organizations may want their customers, gold/platinum card holders and VIP members to access different types of services depending on their privileges.

Our proposed scheme is first of its kind to introduce capability based privacy preserving user authentication and authorization scheme for PCE. The scheme is simple, efficient and cost effective with respect to storage, computation and time complexity. It provides complete privacy and anonymity to the user. The SPs authenticate and authorize the users based on the anonymous information submitted by the users. The SP does not know the user's real identity. The capability issuing server does not know what services the user is accessing. Our scheme discourages capability non-transferability and prevents SP and AS to maliciously collude with each other.

Acknowledgement

We are grateful to Trinh Minh Cuong and Le Xuan Thao for their contributions in the early version of this paper. We also thank anonymous reviewers for their constructive and insightful comments.

References

[1] M. Abe, and T. Okamoto, "Provably Secure Partially Blind Signatures", *Advances in Cryptology - CRYPTO '00*, LNCS 1880, pp. 271-286, 2000.

[2] J. J. Al-Muhtadi, R. Campbell, A. Kapadia, D. Mickunas, and S. Yi, "Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments", *International Conference on Distributed Computing Systems (ICDCS '02)*, pp. 74-83, 2002

[3] J. Camenisch and A. Lysyanskaya, "Effcient Non-transferable Anonymous Multi-show Credential System with Optional Anonymity Revocation", *Advances in Cryptology-EUROCRYPT '01*, LNCS 2045, pp. 93-118, 2001.

[4] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms", *Communications of the ACM*: 24(2), pp. 84-88, 1981.

[5] D. Chaum, "Blind Signatures for Untraceable Payments", *Advances in Cryptology-CRYPTO '82*, pp. 199-203, Plenum Press, 1983

[6] D. Chaum "Security without identification: transaction systems to make Big Brother obsolete", *Communications of the ACM*: 28(10), pp. 1030-1044, 1985.

[7] N. Daswani, "Cryptographic Execution Time for WTLS Handshakes on Palm OS Devices", `http://www-db.stanford.edu/~daswani/papers/WTLSPerformancePaper3.pdf`

[8] H. Handschuh and P. Paillier, "Smart Card Crypto-Coprocessors for Public-Key Cryptography", *Smart Card Research and Applications '00*, LNCS vol. 1820, pp 386 - 394.

[9] C. Hauser, "Privacy and Security in Location-Based Systems With Spatial Models", *Workshop on Requirements for Mobile Privacy and Security'02*, 2002.

[10] U. Jendricke, M. Kreutzer, and A. Zugenmaier, "Pervasive Privacy with Identity Management", *Ubiquitous Computing '02*, 2002.

[11] A. Weimerkirch, C. Paar, and S.C. Shantz, "Elliptic Curve Cryptography on Palm OS Device", *Australasian Conference on Information Security and Privacy '01*, 2001.

[12] A.D. Woodburry, D.V. Bailey, and C. Paar, "Elliptic Curve Cryptography on SmartCards without Coprocessor", *Smart Card Research and Advanced Applications '00*, 2000.

[13] D. S. Wong, H. H. Fuentes and A. H. Chan, "The Performance Measurement of Cryptographic Primitives on Palm Devices" - *Computer Security Applications Conference '01*, pp. 92-101, 2001.