Computers and Mathematics with Applications 60 (2010) 234–244

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

An inter-domain authentication scheme for pervasive computing environment

Lin Yao [a,b,*], Lei Wang [b], Xiangwei Kong [a], Guowei Wu [b], Feng Xia [b]

[a] School of Electronics and Information Engineering, Dalian University of Technology, Dalian 116023, China

[b] School of Software, Dalian University of Technology, Dalian 116023, China

ARTICLE INFO

Keywords:
Inter-domain authentication
Key establishment
Biometric Encryption
Signcryption
Pervasive computing

ABSTRACT

In a pervasive computing environment, mobile users often roam into foreign domains. Consequently, mutual authentication between the user and the service provider in different domains becomes a critical issue. In this paper, a fast and secure inter-domain authentication and key establishment scheme, namely IDAS, is proposed. IDAS adopts Biometrics to guarantee the uniqueness and privacy of users and adopts signcryption to generate a secure session key. IDAS can not only reduce the burden of certificates management, but also protect the users and authentication servers against fraud. Compared with some other authentication methods, our approach is superior with faster key exchange and authentication, as well as more privacy. The correctness is verified with the Syverson and Van Oorschot (SVO) logic.

1. Introduction

Pervasive computing is the next level of computing environment with information and communication technology anywhere, anytime for everyone [1]. Pervasive computing holds the promise of simplifying daily life by integrating mobile devices and digital infrastructures into our physical world. With such abundant service users and service providers, authentication becomes quite important prior to the access of services, not only because it is a basic security service to assure service users and providers that they are interacting with the intended entities, but also because mutual authentication can prevent information leakage, avoid service abuse, and defend against malicious attacks.

Pervasive computing applications are characterized by the following basic elements [2]: (1) ubiquitous access, (2) context awareness, (3) intelligence, (4) natural interaction. In response to the features above, the traditional security mechanism cannot be applied straightforwardly, because traditional security mechanism is based on a static network or closed system with a central control. Communication parties in a pervasive computing environment are unpredictable and dynamic. Consequently, dynamic mutual trust between service users and providers should be established. Due to the high mobility in pervasive computing environment, communicating parties often roam into different domains. Hence, fast and secure inter-domain authentication as well as intra-domain authentication should be highly emphasized in pervasive computing environments.

To secure inter-domain authentication in pervasive computing environments, the first challenge is that the users and the service providers should authenticate mutually, which secures the subsequent interactions. The two classical authentication approaches are either based on knowledge or token. However, token and knowledge are prone to be forgotten, lost, stolen or duplicated. Neither approach is able to represent the uniqueness of a user and impersonation attacks are possible. It is

* Corresponding author at: School of Electronics and Information Engineering, Dalian University of Technology, Dalian 116023, China.
E-mail addresses: yaolin_yl@hotmail.com (L. Yao), kongxw@dlut.edu.cn (X. Kong).

0898-1221/$ – see front matter
doi:10.1016/j.camwa.2010.01.010

necessary to provide an unobtrusive and convenient mutual authentication mechanism. The second challenge is to consider the balance between security and the capabilities of computing, storage and communication, because devices in pervasive computing environment are usually mobile and embedded or portable, which have limited resources.

To sum up the above arguments, the inter-domain authentication scheme for pervasive computing environment should meet the following requirements:

(1) Entity authentication: It can prevent an intruder from impersonating a legitimate user to register to the authentication server and vice versa.

(2) Confidentiality: Only authenticated users can access the services.

(3) Low computation, storage and communication cost: Devices in pervasive computing environment have limited resources. Therefore, as few as possible resources should be used during inter-domain authentication.

In this paper, a fast and secure inter-domain authentication and key establishment scheme for pervasive computing environment is proposed – namely IDAS (Inter-Domain Authentication Scheme) – which involves four entities: A, B, SA and SB, where A and B represent communication entities, and SA and SB represent authentication servers in the domains of A and B respectively. IDAS adopts Biometric Encryption technique [3] to authenticate a mobile user. Biometric Encryption, instead of a certificate, frees the devices from heavy burden of certificates management. IDAS adopts signcryption technique [4] to establish a secure and fresh session key for A and B, which achieves both the functions of digital signature and public key encryption, while using far less resources than that required by "digital signature followed by encryption". The security analysis and the comparisons with other methods show that IDAS meets the above requirements.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 presents Biometric Encryption and Signcryption technique. IDAS is described in Section 4. We analyze the security of the protocol, verify its correctness with the SVO logic [5], and compare it with other protocols in Section 5. A conclusion is given in Section 6.

2. Related work

According to the number of participators, the existing inter-domain authentication protocols can be divided into three classes.

The first inter-domain authentication protocol class includes three entities: A, B and SB. Summit [6] et al. proposed an inter-domain authentication protocol based on a proof token. A proof token binds a subject's identity with a public key as a digital certificate does. Additionally, a proof token also proves the fact that the subject has been successfully authenticated in the issuer's domain when it is issued. The protocol avoids the interaction between the current domain server and the user's registration domain server, and lowers the time delay of authentication. But it requires to issue a roaming certificate between servers, which brings some cost for administrators.

The second inter-domain authentication protocol class includes four entities: A, B, SA, and SB. Ford [7] et al. proposed an inter-domain authentication and key negotiation protocol based on identity encryption. This scheme avoids the administration and verification process of certificates, but the whole interaction process is based on identity encryption. Identity-based encryption cannot protect the user's privacy well. A large amount of message exchange brings about long time delay. Yeh and Sun [8] proposed two four-party password-based authentication and key establishment protocols, which need public key infrastructure to distribute and verify the servers' public keys for the clients. This is a significant requirement for standard password-based authentication protocols in wired network applications, but less desirable for lightweight computing environments.

The third inter-domain authentication protocol class includes five entities: A, B, SA, SB and P, where P is the father server of SA and SB. Ren-Junn [9] et al. proposed an inter-domain authentication protocol based on symmetric encryption and hash functions. Hung-Yu Chien [10] et al. also proposed a similar protocol, in which symmetric encryption is replaced by public-key encryption. Both methods adopt hash functions instead of certificates. The overhead of maintaining certificates is reduced, but certificate scrambling can easily be brought about, and the additional server increases the time delay.

All the protocols mentioned above show disadvantages either in time delay, or in computation and storage cost, which are not suitable for pervasive computing environments. In our inter-domain authentication scheme, we adopt Biometrics to guarantee the uniqueness and privacy of users and adopt Signcryption to generate a secure session key. IDAS can provide an unobtrusive and convenient authentication mechanism, and protect users and authentication servers against fraud. Biometric encryption and Signcryption techniques are introduced in the following section.

3. Biometric Encryption and signcryption

3.1. Biometric Encryption

Compared with the two classical personal authentication approaches, knowledge based approach and token based approach, Biometrics can represent the uniqueness of a user through electronic examinations of his or her physiological characteristics such as iris, fingerprint, or face, and/or through behavioral characteristics.

Conventional biometric identification typically consists of an enrollment stage and a verification stage. During the enrollment stage, a user's biometric template is gathered and stored in the authentication server. During the verification stage, the user's biometrics sampled on the spot are matched against the stored biometric template to verify his or her

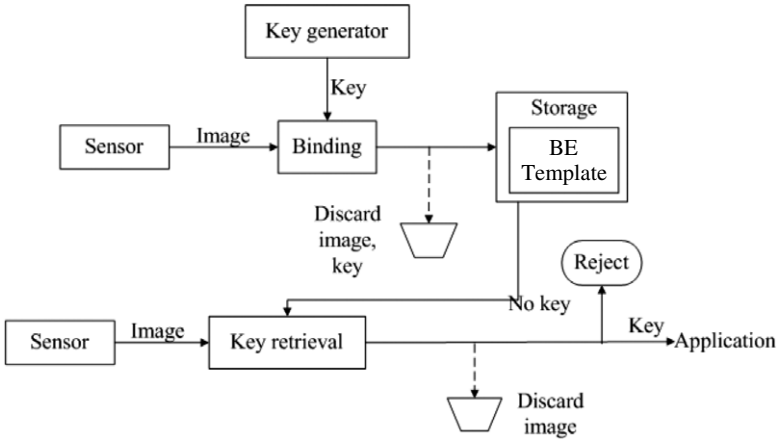

Fig. 1. Biometric Encryption process.

Table 1
Signcryption implementation.

Sender encrypts message m		Receiver decrypts to verify
$x \in [1 \ldots q-1]$ $(k_1, k_2) = hash((y_b{}^x) \bmod p)$ $c = E_{k_1}(m)$ $r = KH_{k_2}(m)$ $s = (\frac{x}{(r+xa)} \bmod q)$	$\Rightarrow c, r, s \Rightarrow$	$(k_1, k_2) = hash((y_a \cdot g^r)^{(s.xb)} mod p)$ $m = D_{k_1}(c)$ Only when $KH_{k_2}(m) = r$, m can be accepted.

identity. If the stored biometric template of a user is compromised, there could be severe consequences for the user because the biometric template lacks revocation mechanisms.

With the proliferation of information exchange across the Internet and the storage of sensitive data in an open network environment, cryptography is becoming an increasingly important feature of computer security. Regardless of whether a user employs a symmetric or a public-key system, the security is dependent on the secrecy of the secret or private key (passcode). Because of the large size of a cryptographically-strong key, it would clearly not be feasible to require the user to remember and enter the key each time. It is necessary to develop a method so that the user need not remember the passcode, and only the valid user can release the key. Bodo proposed a method in a German patent, where the data derived from the biometrics could be used directly as a cryptographic key. However, the leakage of a cryptographic key can lead to the disclosure of the biometrics.

In order to protect the users' biometric templates and keys, Mytec Technologies Inc proposed Biometric Encryption technique (BE) [3]. The Biometric Encryption solution also has a two-stage process as shown in Fig. 1: the enrollment stage and the verification stage. During the enrollment stage, the biometrics is bound with a cryptographic key to create some secret data as Bioscrypt. During the verification stage, the sampled biometrics on the spot are combined with the Bioscrypt to recover the key. Bioscrypt does not reveal any information about the key or the biometric features. It is computationally hard to decode the key without any knowledge of the user's biometrics, and vice versa. Consequently, Bioscrypt can provide excellent privacy protection. The key itself is completely independent of biometrics. Therefore, it can be changed or updated when required. Even if the key is ever compromised, the biometric cannot be leaked. In conclusion, Biometric Encryption can not only secure a cryptographic key, but also can protect a user's biometrics, which meets the four properties described in [11]: diversity, revocability, security and performance.

Bioscrypt can reduce the storage requirement for mobile devices and the management requirement for certificates. Besides, it is faster to generate the Bioscrypt with face than with fingerprint or iris, which can reduce the registration time.

3.2. Signcryption

Signcryption was proposed by Yuliang Zheng [4]. Signcryption can achieve both signature and public-key encryption in a rational logic procedure, and the cost is much lower than the traditional way that digital signature is followed by encryption. Hence, it is a perfect method to transmit and preserve information with encryption and authentication.

The signcryption technique based on a discrete logarithm can be described as follows. P is a big prime number. Q is a big prime factor of $p-1$. G selected from 1 to $p-1$ is an integer of factorial q. E represents encryption and D represents decryption. $KH_k(m)$ is a hashing function of message m using the key k. The detailed procedure is shown in Table 1.

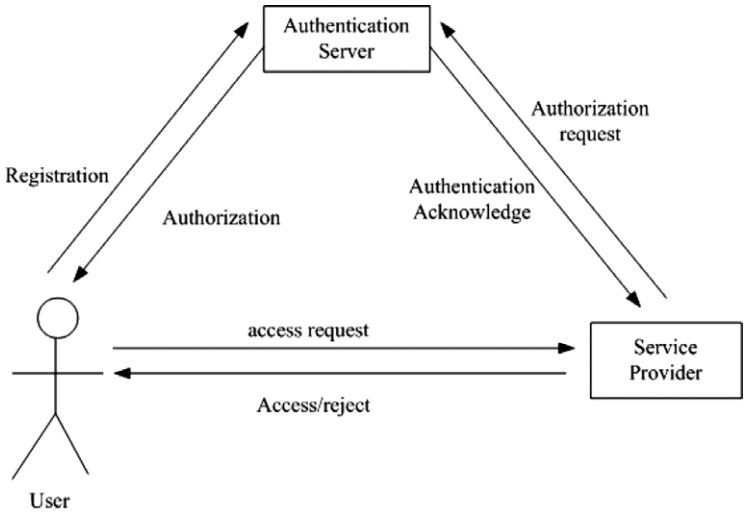

Fig. 2. Mutual authentication architecture.

In order to increase the security of signcryption, c, r and s are transmitted secretly and random numbers xa and xb are used to resist replay attacks. Moreover, after finishing signcryption, both sides begin to negotiate a session key to secure the subsequent traffic.

4. Inter-domain authentication scheme

The proposed system IDAS includes four entities: two users A and B, and two authentication servers SA and SB. A and B are located in different domains. SA and SB are servers in the domains of A and B respectively. In IDAS, it is required that only legal users in one domain can request to access other services in other different domains. Therefore, IDAS includes two parts: intra-domain authentication and inter-domain authentication. Intra-domain authentication sets up mutual trust between a user and a service provider in the same domain, and generates Bioscrypt and a secret key for inter-authentication. Inter-authentication builds mutual trust between a user and a service provider in the different domains, such as between A and B. In this section, we first present intra-domain authentication briefly, and then we describe inter-domain authentication in detail.

4.1. Intra-domain authentication

The intra-domain authentication system architecture similar to that in [12] is shown in Fig. 2, consisting of three types of entities: the mobile user (U), the service provider (SP), and the authentication server (AS). It includes three phases: the registration phase between U and AS, the access service phase between U and SP, and the authentication phase between SP and AS. U first registers to AS. When U wants to access some service, U sends an access request to SP. SP forwards the request to AS in order to get some acknowledgment on U from AS. If U is illegal, the access request is rejected. During the registration phase, Biometric Encryption algorithm is adopted to generate a certificate-like item named Bioscrypt. Bioscrypt represents a user's identity and can be used to perform mutual authentication with the service provider.

4.1.1. Biometric Encryption algorithm

In IDAS, the Biometric Encryption cryptosystem is shown in Fig. 3, including the key linking and retrieving phases.

In the key linking phase, the face is first processed by discrete-hashing [12] based on the iterative inner-product between a user's face and a tokenized random number r which can be produced from a seed in a secure device or remembered by the user. Discrete-hashing is described as follows.

(1) Feature extraction. Fisher Discrimination Analysis (FDA) [13] is used to extract the face features represented in a vector format, $w \in R^n$ with n denoting the feature length of w.
(2) Use r to generate m orthogonal pseudo random vectors, $\{r^i \in R^n | i = 1, 2, \ldots m\}$ and $m \leq n$.
(3) Compute the inner product $\{t^i \in T | i = 1, 2, \ldots, m\}$ between r and w.
(4) Compute a m bit FaceHash code: $b^i = 0$ if $t^i \leq 0$; $b^i = 1$, if $t^i > 0$.

Next, Reed–Solomon codes are designed to correct the errors (bit differences) within the reference and test FaceHash.

Then, the biometric template is protected by XOR operation as shown in Eq. (1). σ represents biocode. $H(key)$ is stored in the authentication server and H is the hash function.

$$b \oplus key = \sigma. \tag{1}$$

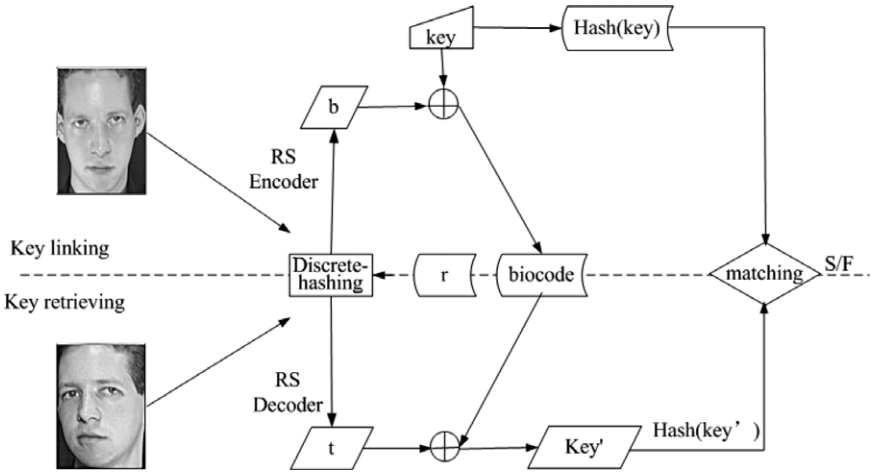

Fig. 3. The key binding cryptosystem.

At last, *key* and biometrics are discarded.
In the key retrieving phase, the fresh face sample is first discretized into biocode, which is represented by t.
Next, Reed–Solomon codes are used to correct the error within the reference and test FaceHash.
Then, key' is retrieved by computing Eq. (2).

$$\sigma \oplus t = key'. \tag{2}$$

At last, if $hash(key) = hash(key')$, then the user is legal. The face sample is discarded again.

4.1.2. Intra-domain authentication

The primary intra-domain authentication steps are shown in Fig. 4. For ease of reference, the notations used in the protocol description are listed in Table 2.

(1) First, when a user comes into a domain, he must submit his face biometrics to his AS before he accesses some services.
(2) AS generates a new key K_{sid} for the SID requested by the user and binds the user's biometrics with the new key to generate biocode as Bioscrypt. AS stores $h(K_{sid})$ and its corresponding SID. Moreover, biometrics, Bioscrypt and K_{sid} are discarded. In this step, AS will distribute a different number to every user in order to match the corresponding credential. A user's real identity cannot be deduced by the number, which helps to protect a user's privacy.
(3) A user sends an access request encrypted with K_{sa} to SP. This message includes the number generated in step (2), Bioscrypt, SID, biometric feature vectors, and a time stamp T_u.
(4) SP simply forwards this request to AS.
(5) After receiving the request message from SP, AS first decrypts the message with its private key. Second, it verifies the time stamp to make sure that a replay attack has not happened. Third, AS checks whether SP is the right service provider by comparing the SP's SID list with the SID received.
(6) If there is a match in step (5), a key is recovered from Bioscrypt and biometric feature vectors.
(7) AS fetches the hash value of the key according to the user's number and SID in step (2). If the stored $h(key)$ matches the hash value of the key in step (6), the user is authenticated successfully. Then, AS sends an encrypted acknowledgment to SP, including a time stamp T_a, SID and $h(K_{sid})$.
(8) After receiving the acknowledgment from AS, SP checks the time stamp to judge whether it is replayed by decrypting this message with SK_{sa}.

At the end of the mutual authentication, both the user and the SP hold a new session key $h(K_{sid})$ to secure the subsequent traffic.

4.2. Inter-domain authentication and key establishment

In this section, we describe our inter-domain authentication and key establishment protocol. Suppose that the public key of each entity is open to all, A is a user requesting services and B is a service provider in a different domain. A and B have finished intra-domain authentication. Each message includes the sender's and the receiver's ID, and here user ID is omitted for simplicity. The detailed interactions are shown in the Fig. 5.

(1) A sends an access request to B, which includes SID, a random number N_a, and an authentication factor to SA. The authentication factor encrypted with K_{sa} includes the number assigned by SA, the corresponding Bioscrypt, the face feature vectors, and a random number N_a. The whole request is encrypted with K_{sb}.
(2) B only appends his service list (SID-List) and his random number N_b encrypted with K_{sb} to the access request.

Table 2
Notations.

Symbols	Meaning
U	A service user.
AS	Authentication server which authenticates the user for access control.
SP	Service provider.
SID	A service type identifier. Different users may apply for the same service type.
$KH()$	A keyed hash function.
$E()$	Encryption function.
K_a, K_a^-1	Public and private key pair of entity A.
SK_{sa}	Shared secret key between entities A and S.
h()	A secure one-way hash function.
T	Time stamp face User's biometric feature vectors.
ack	Access acknowledgment.

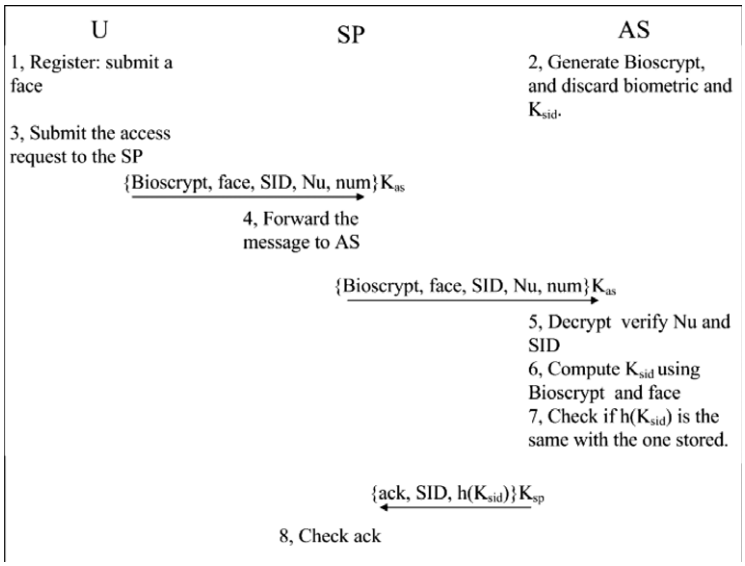

Fig. 4. Mutual authentication.

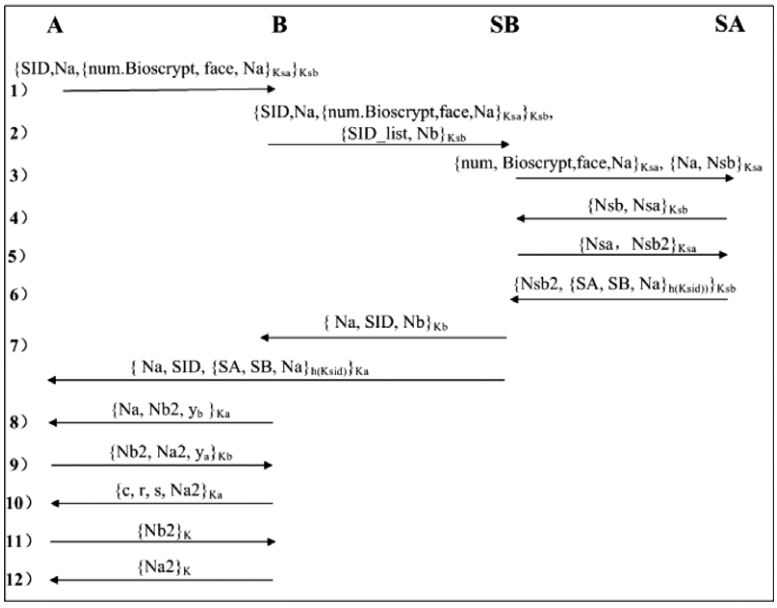

Fig. 5. Inter-domain authentication and key establishment.

(3) SB decrypts the message from B. If A is one of SB's registered users, inter-domain authentication starts. Otherwise, SB sends an authentication request encrypted with K_{sa} to SA, which includes A's authentication factor and a random number N_{sb} generated by SB.
(4) SA decrypts the message from SB and looks up the corresponding $h(K_{sid})$ according to the number. If $h(K_{sid})$ matches $h'(K_{sid})$ recovered from Bioscrypt and face feature vectors, it shows that A is legal. SA sends a message including N_{sa} and N_{sb} encrypted with K_{sb} to SB.
(5) If N_{sb} decrypted by SB is correct, it shows that SA is legal. Then SB sends a feedback to SA, which includes N_{sa} encrypted with K_{sa}.
(6) If N_{sa} is verified to be correct, it shows that SB is legal. If SA trusts SB, SA sends the authentication result of A to SB. The whole message is encrypted with $h(K_{sid})$, which can only be decrypted by A.
(7) If N_{sb} is correct, SB sends a feedback to B and A. The message sending to B includes N_a, SID, N_b and a new random number N_{sb2}, which is encrypted with K_b. The message sending to A includes N_a, SID and some message from SA to A, which is encrypted with K_a.
(8) Based on the validity of N_b from step (7), B can authenticate SB and A. If SB and A are legal, B begins to prepare for the key establishment. First, B selects its own secret x_b and uses public parameters g and p to calculate $y_b = g^{xb} \ mod \ p$. Then, B sends an acknowledgment to A encrypted with K_a, which includes y_b and N_{b2} produced newly.
(9) A checks the validity of SB through N_a from step (7). If N_a decrypted with $h(K_{sid})$ is correct, SA is legal. A selects its own secret x_a and calculates the public information $y_a = g^{x_a} \ mod \ p$. A checks the validity of B through N_a from step (8). If B is legal, A sends a feedback to B encrypted with K_b, which includes y_a and N_{a2},.
(10) B checks the validity of A through the correctness of N_{b2}. Until now, the inter-domain authentication is finished and a session key establishment begins. B selects a new *key* with L length and an integer x from 1 to $p-1$ randomly. B calculates $(k_1, k_2) = h(y_a^x \ mod \ p)$, $c = E_{k_1}(key)$, $r = KH_{k_2}(key, N_{a2})$, $s = \frac{x}{(r+x_b)} \ mod \ q$, and then sends a message to A encrypted with K_a, which includes c, r, s and N_{a2}.
(11) A decrypts the message from (10). Then A calculates $(k_1, k_2) = h((y_b \cdot g^r)^{(s.x_a) \ mod \ p})$, $key = D_{k_1}(c)$ and $KH_{k_2}(key, N_{a2})$. If $KH_{k_2}(key, N_{a2})$ is equal to r, the key is proved to be secure. $K = KH_{key}(N_{a2})$ is calculated as the new session key, and N_{b2} is encrypted to B with K.
(12) If N_{b2} is correct, B begins to calculate K in the same way as A and gives a feedback.
(13) A verifies the correctness of N_{a2}. Until now, the inter-domain authentication and key establishment process has finished. The new session key K is used to secure the subsequent traffic for A and B.

5. Protocol analysis and verification

5.1. Protocol analysis

5.1.1. Mutual authentication

The main aim of mutual authentication is to verify the identity of a service user and a service provider. According to the domains where a service user and a service provider are located, there are the following mutual authentication situations in IDAS.

Mutual authentication between A and SA: SA recovers $h'(K_{sid})$ from the Bioscrypt and the face feature vectors of A. A authenticates SA based on the comparison result between $h(K_{sid})$ and $h'(K_{sid})$. Only SA and A know $h(K_{sid})$, so A can decrypt the message. As a result, SA can prove the validity of A. Mutual authentication between A and SA is achieved by messages in steps (1), (2), (3), (6) and (7) of inter-domain authentication.

Mutual authentication between B and SB: SB authenticates B based on the SID-list of B. B authenticates SB based on the random numbers and the public key encryption. Mutual authentication between B and SB is achieved by the messages in steps (2) and (6) of inter-domain authentication.

Mutual authentication between A and SB: SB authenticates A according to the authentication result between SA and A. A authenticates SB according to the authentication result between SA and SB. Both A and B authenticate each other based on the decisions of SB and SA. Mutual authentication between A and SB is achieved by the messages in steps (1), (2), (3), (4),(5), (6) and (7) of inter-domain authentication.

Mutual authentication between SA and SB: Authentication is achieved by the messages in steps (4), (5), and (6) based on the public key encryption and random numbers between SA and SB, which is a typical challenge-response scheme.

5.1.2. Security analysis

Off-line guess: In off-line guess attack, attackers try to infer the privacy information from the captured messages, which is harmful, because attackers have no limitations of time and resources for calculating, and all kinds of resources can be used to decrypt the message at off-line state. In our scheme, all messages are encrypted with random numbers, which makes off-line guess attack more difficult. During the phase of key establishment, it is infeasible for attackers to attempt to guess c, r and s from message (10) of inter-domain authentication, because the message is encrypted. Furthermore, both c and r are encrypted with a random number, which makes the message more secure.

Table 3
Length of the parameters.

Parameter	Length	Annotation
Bioscrypt	LB	The result of biocode.
Face	LF	The length of face sample
Random	LR	The length of the random number
Hash	LH	The result of Hash. The length is 128bit for MD5
Public Key	LP	Suppose the length of all public key is the same

Table 4
Storage requirements.

	Fixed storage	Temporary storage
A	LB+LF+2LR+3LP+LH	3LR+4LH
B	2LR+3LP+LH	3LR+4LH
SA	LH+LP	2LR
SB	3LP+LH	2LR

Table 5
Computation cost.

	Authentication					Key establishment				
	PE	SE	DE	Hash	Ex	PE	SE	DE	Hash	Ex
A	2	0	2	0	0	1	1	3	2	1
B	1	0	1	0	0	2	1	3	2	1
SA	2	1	3	1	0	0	0	0	0	0
SB	4	0	4	0	0	0	0	0	0	0

Online guess: Similar to the off-line guess, there is less probability for attackers to succeed because of the security of cryptography system, random numbers and the limited resources.

Forward stability: Even though some of the previous session keys are obtained by the attackers, the new keys cannot be guessed out. In our protocol, the session key K is relevant to *key* of B which is different and random every time. Thus, attackers cannot predict the new session key K even though they have gotten some previous session keys and the *key* of B.

Backward stability: Even though the attackers know the current session key K, old session keys cannot be guessed out because there are no relevancies among these session keys.

Man-in-the-middle attack: In the worst case, suppose SA or SB is dishonest as a man-in-the-middle attacker. On the one hand, SA or SB wants to decrypt the messages between A and B. After authentication, A and B break away from the control of SA and SB. The session key K is established by A and B without SA's and SB's participation, so SA and SB can not know the session key K and cannot decrypt the messages between A and B. On the other hand, SB or SA wants to carry out impersonation attacks. Because SB knows nothing about Bioscrypt and other secret information shared by A and SA, SB cannot impersonate A or SA. Similar to SB, SA cannot get any secret information shared by SB and B, so SA cannot impersonate B or SB.

5.1.3. Performance analysis

In this section, we analyze IDAS in terms of storage, computation and communication traffic cost. The length of the parameters used in IDAS is listed in Table 3. In IDAS, every entity has temporary and permanent parameters. A needs to store Bioscrypt, face feature vectors, K_{sa} and K_{sb} as permanent parameters. K_a, K_a^-1, and K are stored as temporary parameters. B stores K_b, K_b^-1, K_{sa}, K_{sb} and K as permanent parameters. SB and SA store their own public keys. The storage requirements are listed in Table 4.

If hash function is MD5, the hash result will be 16 bytes. In that case, the maximum permanent parameters storage space of A is 90 bytes plus the length of public key, and the maximum permanent parameters storage space of B is 90 bytes plus the length of public key. The storage requirement is lightweight even for embedded or portable equipments.

The computation overhead focuses on encryption, decryption and hash operations at the stages of authentication and key establishment. The computation cost is listed in Table 5. In Table 5, PE represents the operation times of public key encryption. SE represents the operation times of symmetric encryption. Ex represents the operation times of exponentiations. DE represents the operation times of decryption.

It can be seen that, during the inter-domain authentication and key establishment phases, one needs 12 instances of public key encryption operation, 16 instances of decryption operation, 3 instances of symmetric encryption, 5 instances of hash operation and one exponentiations operation. In IDAS, public-key encryption and decryption consumes the main computation time and resources. With the rapid development of ASIC, the process of public-key encryption is much easier and faster.

The analysis of communication traffic is in terms of the number of messages and the size of messages. During the authentication phase, there are a total of eight messages in the inter-domain communication between A and B, between

B and SB, and between A and SB. There are a total of four messages to establish a secure session key during the key establishment phase. Another advantage of IDAS is that the messages as the output of these encryption algorithms have a fixed size.

From the above analysis on storage, computation and communication traffic cost, we can conclude that IDAS is a lightweight protocol which can well meet the requirements of pervasive computing environments.

5.2. Correctness proof

In this section, the correctness of IDAS is formally verified with the SVO logic [10,5]. SVO logic is based on a unification of four of its predecessors in the BAN family of logics and is relatively simple to use. Protocol correctness means that, after secure mutual authentication, both parties ascertain that they are sharing a fresh session key, and both are sure that the same trust is held by the other side. In particular, in IDAS, after the execution of the key establishment protocol, both A and B obtain a new session key K. Therefore, the verification goals are as follows.

$$A \mid\equiv A \overset{K}{\longleftrightarrow} B \; B \mid\equiv B \overset{K}{\longleftrightarrow} A.$$

Firstly, we should formalize the messages transmitted between the two entities. Secondly, the premises sets are built, which include the initial trust of the two entities, the interaction messages and so on. The formalized message interaction sequences and premises sets are shown as follows. The marks and symbols can be referred to the definitions of SVO.

Formalized Message Sequence

$M1.\ B \rightarrow A : \{[N_a]_{k_b^{-1}}, N_{b2}, y_b\}_{K_a}$

$M2.\ A \rightarrow B : \{N_{b2}, N_{a2}, y_a\}_{k_b}$

$M3.\ B \rightarrow A : \{c, r, s, N_{a2}\}_{K_a}$

$M4.\ A \rightarrow B : \{N_{b2}\}_K$

$M5.\ B \rightarrow A : \{N_{a2}\}_K$

Premises Sets

$P1.\ A \mid\equiv \#(N_a);\ A \mid\equiv \#(N_{a2})$

$P2.\ B \mid\equiv \#(N_b);\ B \mid\equiv \#(N_{b2})$

$P3.\ A \mid\equiv PK_\delta(A, N_{a2})$

$P4.\ B \mid\equiv PK_\sigma(B, key)$

$P5.\ A \mid\equiv \mid\Rightarrow A \overset{key}{\longleftrightarrow} B$

$P6.\ A \mid\equiv (A \propto \{[Na]_{k_b^{-1}}, N_{b2}, y_b\} \supset A \mid\equiv \{[N_a]_{k_b^{-1}}, N_{b2}, PK_\delta(B, y_b, \#(y_b))\})$

$P7.\ A \mid\equiv (A \propto \{c, r, s, N_{a2}\}_{K_a} \supset A \mid\equiv \{c, r, s, \#(N_{a2}), PK_\delta(B, key), \#(key)\}_{Ka})$

$P8.\ A \mid\equiv (A \supset \{N_{a2}\}_K \supset A \mid\equiv \{\#(N_{a2})\}_K)$

$P9.\ B \mid\equiv (B \propto \{N_{b2}, N_{a2}, y_a\}_{k_b} \supset B \mid\equiv \{\#(N_{b2}), N_{a2}, PK_\delta(A, y_a), PK_\delta(A, N_{a2})\}_{k_b})$

$P10.\ B \mid\equiv (B \propto \{N_{b2}\} \supset B \mid\equiv \{\#(N_{b2})\}_K).$

Assume that both A and B believe their servers' jurisdiction of the legitimacy. Both entities believe their own random numbers and the key parameters generated in IDAS are secure. It also assume that each principal believes the freshness of the random numbers. Other premises are made according to the expressions of the SVO logic.

The first two assumptions P1 and P2 state that both A and B trust the freshness of their random numbers. P3 and P4 state that each principal believes their own agreement key is secure and each principal controls the generation of the agreement key. P6, P7 and P8 show what A's comprehension about the session key, while P9 and P10 show B's complementation about the session key.

After the messages are formalized and the premises sets are built, the verification procedure from A's standpoint is listed as follows. The procedure from B's is similar to that of A. R7 and R8 show that IDAS has reached the anticipated aims.

Verification Procedure

$R1.\ A \mid\equiv B \mid\approx \{c, r, s, \#(N_{a2}), PK_\delta(B, key), \#(key)\}$,by M3, P7, P5, P1, Ax3, MP, Nec

$R2.\ A \mid\equiv PK_\sigma(B, key)$, by R1, Ax15, Ax16, Nec

$R3.\ A \mid\equiv A \overset{K}{\longleftrightarrow} B$, by R2, P3, Ax5, Nec

$R4.\ A \mid\equiv A \overset{K}{\longleftrightarrow} B$, by R3, the fact that A produces K

$R5.\ A \mid\equiv B \mid\sim \{\#(N_{a2}) \wedge B \lhd K\}$, by R5, P1, Ax19, Nec

$R6.\ A \mid\equiv (B \mid\approx \{B \lhd K\})$, by R5, P1, Ax19, Nec

$R7.\ A \mid\equiv A \overset{K}{\longleftrightarrow} B$, by R4, R6, Nec

$R8.\ A \mid\equiv \#(K)$, by the fact that A produces K.

Table 6
Comparison.

Items	Protocols						
	Entity	Message	M with RS	Privacy	Computation	Storage	Assumption
Summit's protocol	3	10	0	N	PE	Proof token and roaming-certificate	Pre-established trust between domains
ID-4-PAKE	4	12	6	N	PE and SE	Initial passwords	No special requests
Ren-Junn's protocol	5	10	4	Y	SE and hash	Certs and symmetric keys	Symmetric keys between the entity and its server
Our protocol	4	12	4	Y	PE and SE	Bioscrypt	No special requests

5.3. Comparisons

In this section, IDAS is compared with some other schemes in terms of security, privacy and performance. The comparison results are shown in Table 6. In Table 6, the column 'Entity' represents the numbers of entities that participate in the protocol, and the 'Message (M)' column shows the total numbers of messages that are transmitted during the protocol. The column 'M with RS' shows how many times the entities should communicate with their register servers (RS). The column 'Privacy' shows whether the protocol protects the user's privacy or not. While the 'Computation' column compares the main computing operations, the 'Storage' column shows the storage requirements for the entities. The last column presents the assumption conditions of each protocol. In Table 6, PE represents public key encryption, SE represents symmetric encryption, BE represents Biometric Encryption, and Cert represents certificate.

In Table 6, though there are only three entities involved in Summit's protocol and no interactions with the register server, the users need to maintain a large amount of proof tokens and the servers need to maintain a large amount of roaming-certificates. All these operations bring great storage burdens to pervasive computing devices. What is more, Summit assumes that the servers have already built a trusting relationship. ID-4-PAKE protocol is based on identity encryption; the user's privacy cannot be protected well. Moreover, large amount of messages interacting with the register server results in long time delays. Due to adopting an extra server, Ren-Junn's protocol becomes simple. However, more servers mean more time delay and more resource requirements for the environment. In addition, the hash chain used in Ren-Junn's protocol may bring about the problem of wrong sequences. In that case, the certificates are not coming in order and will require extra processes and resources. IDAS replaces a certificate with Bioscrypt, generated by Biometric Encryption. Bioscrypt instead of the certificate can not only reduce the computation and storage cost, but also protect the user's privacy well. IDAS has no special assumption conditions. Besides, IDAS reduces the interactions between users and their register servers, and has no special requirements for the environment.

6. Conclusion and future work

In this paper, we present an inter-domain authentication and key establishment scheme (IDAS) for pervasive computing environments. In IDAS, Biometrics and Signcryption techniques are adopted to protect users' privacy and protect users and authentication servers against fraud. Bioscrypt can reduce the storage requirement for mobile devices and the complexity of certificates. The correctness of IDAS is proved with SVO logic. Compared with other schemes, IDAS is superior with higher security, higher performance and more privacy. The primary contributions of this paper are summarized as follows:

- A new mutual authentication and key establishment scheme for pervasive computing environment is presented.
- Biometric Encryption is applied to protect users' privacy during mutual intra-domain authentication and inter-domain authentication.
- Signcryption technique is applied to establish secure session keys, which can secure the subsequent traffic.
- SVO logic is used to verify the correctness of IDAS.

As part of our future work, we intend to do more research on Biometric Encryption algorithms to decrease the false acceptance rate to zero. Then we would like to extend our scheme to the mobile wireless network environment, where location-based services are one of the most desirable classes of services to be provided for mobile users. A user's identity can be verified based on his face biometrics captured by a high speed camera. We will also evaluate IDAS by simulations and testbed experiments based on the vehicles that will incorporate intersection behavior. It is also valuable to provide fault-tolerant features for IDAS in order to prevent some of the communication messages from being lost during the authentication phase.

References

[1] Nguyen Ngoc Diep, Sungyoung Lee, Young-Koo Lee, Heejo Lee, A privacy preserving access control scheme using anonymous identification for ubiquitous environments, in: RTCSA, IEEE Computer Society, 2007, pp. 482–487.
[2] Alois Ferscha, Pervasive computing — kurz erklärt, Datenbank-Spektrum 7 (2003) 48–51.
[3] Randall K. Nichols, Icsa Guide to Cryptography, McGraw-Hill Professional, 1998.

[4] Yuliang Zheng, H. Imai, Compact and unforgeable key establishment over an atm network, in: INFOCOM '98, Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings of the IEEE 2 (1998) 411–418.
[5] P.F. Syverson, P.C. van Oorschot, On unifying some cryptographic protocol logics, in: 1994 IEEE Computer Society Symposium on Research in Security and Privacy, 1994, pp. 14–28.
[6] Summit R. Tuladhar, Carlos E. Caicedo, James B.D. Joshi, Inter-domain authentication for seamless roaming in heterogeneous wireless networks, in: SUTC '08: Proceedings of the 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, SUTC 2008, IEEE Computer Society, Washington, DC, USA, 2008, pp. 249–255.
[7] Ford Long Wong, Hoon Wei Lim, Identity-based and inter-domain password authenticated key exchange for lightweight clients, in: AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops, IEEE Computer Society, Washington, DC, USA, 2007, pp. 544–550.
[8] Her-Tyan Yeh, Hung-Min Sun, Password authenticated key exchange protocols among diverse network domains, Computers & Electrical Engineering 31 (3) (2005) 175–189.
[9] Ren-Junn Hwang, Feng-Fu Su, A new efficient authentication protocol for mobile networks, Computer Standards & Interfaces 28 (2) (2005) 241–252.
[10] Catherine Meadows, Formal methods for cryptographic protocol analysis: Emerging issues and trends, IEEE Journal on Selected Areas in Communications 21 (2) (2003) 44–45.
[11] Anil K. Jain, Karthik Nandakumar, Abhishek Nagar, Biometric template security, EURASIP Journal on Advances in Signal Processing 28 (2008) 1–17.
[12] Andrew Teoh Beng Jin, David Ngo Chek Ling, Alwyn Goh, Biohashing: Two factor authentication featuring fingerprint data and tokenised random number, Pattern Recognition 37 (11) (2004) 2245–2255.
[13] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.