Combining User Authentication with Role-Based Authorazition Based on Identity-Based Signature

Jin Wang[1], Jia Yu[1,2], Daxing Li[1], Xi Bai, and Zhongtian Jia[1,3]

[1] Institute of Network and Information Security, Shandong University, Jinan 250100, China
[2] College of Information Engineering, Qingdao University, Qingdao 266071, China
[3] School of Information Science and Engineering, Jinan University, Jinan 250022, China
{wangjin06,jiayu}@mail.sdu.edu.cn, xibai@email.jlu.edu.cn

Abstract. Authentication and authorization are crucial for ensuring the security of information systems. Role-based access control (RBAC) can act as an efficient method of managing authorization of system resources. In this paper, we apply identity-based signature (IBS) technique to cryptographically provide user authentication and role-based authorization. To achieve this, we first extend the RBAC model to incorporate identity-based cryptography. Our access control architecture is derived from an identity-based signature scheme on bilinear pairings and eliminates the use of digital certificates. In our suggestion, the manager checks the validity of a user's identity and user's activated roles simultaneously by verifying a corresponding signature, thus the user authentication and role-based authorization procedures can be combined into one operation. We also prove the security of the proposed scheme in the random oracle model.

1 Introduction

1.1 Background and Related Work

In proportion to the spread of computation and communication technologies, how to provide security services, especially authentication and authorization , is becoming even more crucial than ever.

Role-Based Access Control. Role-based access control [1,2] is an effective access control method for protecting information and resources in large-scale and enterprise-wide systems. In RBAC, access rights (*permissions*) are associated with *roles*, and *users* are assigned appropriate roles thereby acquiring the corresponding permissions. Moreover, RBAC allows for roles and permissions to be activated within a user's *session*, thus access privileges can be given only when required. RBAC provides administrators with a means of managing authorization of system resources. In the implementation phase, access control should

Y. Wang, Y. Cheung, and H. Liu (Eds.): CIS 2006, LNAI 4456, pp. 847–857, 2007.

be strong and efficient based on user authentication information, so the RBAC mechanism often requires user authentication as a prerequisite.

Identity-based Cryptography. Certificate-based PKI (Public Key Infrastructure)[11] is widely applied to provide user authentication, but there exists grievous management cost expanding problems for public key certificates. Identity-based cryptography (IBC) can eliminate the need for certificates and overcome those hurdles of PKI by allowing a user's public key to be derived from its identity, such as an email address. The idea of identity-based cryptography was first introduced by Shamir [3], and the first practical identity-based encryption scheme was proposed by Boneh and Franklin [4] based on bilinear pairings. Identity-based cryptosystem fits very well to cryptographically support RBAC. Firstly, it is possible to use arbitrary string values, including a user's identity, a role's identity as a public key. And secondly, a user can just get the corresponding private key from the PKG (Private Key Generator) if the user is currently playing the requested role. There is no need to share or store any certificates of the user.

Related Work. There have been several approaches about cryptographic support of access control involving identity-based cryptography. Smart presents a simple mechanism [5] to drive access control to broadcast encrypted data using a variant of identity-based encryption scheme. Nali et al. [6] extend a mediated identity-based encryption scheme to support RBAC. But due to the encryption-based access control method, previous approaches cannot support flexible access rights, and are not suitable for widely application environment.

1.2 Our Contribution

In this paper, we propose a scheme that cryptographically provides user authentication and role-based access control for large organizations based on identity-based signature (IBS) technique. To achieve this, we extend the elements *user* and *role* in RBAC model [1,2] to cooperate with identity-based cryptography. Our suggestion is that each role is associated with a pair of public/private keys. Each user uses his/her identity as a public key, and has a set of private keys (called *assigned key*) corresponding to the roles assigned to him/her. A role's private key is used to generate a user's assigned key while the administrator assigns this role to the user. Our access control architecture is based on a pairing-based identity-based signature scheme [7]. In our proposed scheme, the manager can check the validity of a user's identity and activated roles by verifying the user's signature, so there is no need to authenticate users in an independent procedure.

The rest of this paper is organized as follows. Section 2 introduces some related preliminary information; Section 3 presents our RBAC scheme based on identity-based signature; in Section 4 we analyze the security of the proposed scheme; we conclude in Section 5.

2 Preliminaries

In this section, we briefly review some of the properties of bilinear pairings, and recall an identity-based signature scheme proposed by Cha and Cheon[7], which is the basis of our proposed scheme.

2.1 Bilinear Pairings and Gap Diffie-Hellman Groups

Bilinear Pairing. Let G_1 be an additive group of prime order q and G_2 be a multiplicative group of the same order q. A bilinear pairing is a map $\hat{e} : G_1 \times G_1 \rightarrow G_2$, with the following properties.

1 *Bilniearity:* $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, for all $P, Q \in G_1$, $a, b \in Z_q^*$;
2 *Non-degeneracy:* There exist $P, Q \in G_1$, such that $\hat{e}(P, Q) \neq 1$;
3 *Computability:* There is an efficient algorithms to compute $\hat{e}(P, Q)$ for all $P, Q \in G_1$.

At the same time, we are interested in the following mathematical problems. Let P, Q be elements of G_1 and a, b, c be elements of Z_q^*.

Discrete Logarithm Problem (DLP). Given P, Q, find an integer n such that $P = nQ$, where such n exists.

Computational Diffie-Hellman Problem (CDHP). Given (P, aP, bP), compute abP.

Decisional Diffie-Hellman Problem (DDHP). Given (P, aP, bP, cP), decide whether $c = ab$ in Z_q^*.

We call G a GDH group if DDHP can be solved in polynomial time but no probabilistic algorithm can solve CDHP with non-negligible advantage within polynomial time. Such group can be found on super singular or hyper elliptic curves over finite field. The Weil pairing and the Tate pairing [13] are admissible applications satisfying the properties mentioned above.

2.2 Identity-Based Signature

An Identity-based signature scheme consists of four phases namely *Setup*, *Extract*, *Sign*, and *Verify*. The PKG initializes the system in the Setup phase by generating the system public parameters. The PKG also chooses a master key and keeps it secret. The master key is used in the Extract phase to calculate private keys for the participating users in the system. A signer signs a message in the Sign phase using a private key given by PKG corresponding to his/her identity. To verify a signature of a user with identity ID, a verifier just uses ID in the Verify phase. An identity-based signature scheme proposed by Cha and Cheon[7] is introduced as follows.

Setup: The PKG specifies two groups G_1 and G_2 of prime order q, a generator P of G_1, a bilinear map $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$, and two hash functions $H_1 : \{0, 1\}^* \rightarrow G_1$

and $H_2 : \{0,1\}^* \times G_1 \rightarrow Z_q^*$. It also chooses $s \in Z_q^*$ randomly as its master secret key and computes the global public key P_{pub} as sP.
System params:$\langle G_1, G_2, \hat{e}, P, P_{pub}, H_1, H_2 \rangle$. Master-key: $\langle s \rangle$.

Extract: The PKG verifies the given identity ID, and computes the secret key for the identity as $S_{ID} = sH_1(ID)$. The component $Q_{ID} = H_1(ID)$ plays the role of the corresponding public key.

Sign: To sign a message $m \in \{0,1\}^*$ using the private key S_{ID}, the signer chooses $r \in Z_q^*$ randomly and calculates:

1 $U = rQ_{ID}$
2 $h = H_2(m, U)$;
3 $V = (r + h)S_{ID}$.

Signature: $\sigma = \langle U, V \rangle \in G_1 \times G_1$.

Verify: To verify a signature $\sigma = \langle U, V \rangle$ for an identity ID on a message m, a verifier checks whether $(P, P_{pub}, U + hQ_{ID}, V)$ is a valid Diffie-Hellman tuple. This can be accomplished by the equation below: $\hat{e}(P, V) = \hat{e}(P_{pub}, U + hQ_{ID})$. Notice that this check can be performed because of the assumption that the group G_1 is a GDH group.

3 Our RBAC Scheme Based on IBS

In this section we present a scheme that cryptographically enforces user authentication and role-based access control based on the extension of above Cha-Cheon's scheme. Hereafter we refer our proposed scheme as IRBAC (Identity& Role Based Access Control) scheme.

3.1 Notations

We extend the elements *user* and *role* in RBAC model [1,2] to cooperate with identity-based cryptography.

- User: In our suggestion, each user can be represented as $u = \langle ID, USKS \rangle$. ID is an identity information of the user and is used as a public key. $USKS = \{S_{IDr_1}, ..., S_{IDr_n}\}$ represents a set of *assigned keys* corresponding to the roles assigned to the user.
- Role: A role is described as a set of permissions to access system resources, each role can be represented as $r = \langle rpk, rsk \rangle$. rpk and rsk are defined as a pair of public/private keys belonging to the role, where rsk is randomly chosen from Z_q^* and $rpk = rsk \cdot P$. Here our system parameters are identical to Cha-Cheon's scheme, where P is a generator of G_1. Each role can be considered as be associated with a PKG, which generates user's assigned key as a function of its rsk and a user's identity while assigning the role to the user.

3.2 System Architecture

The entities participating in the scheme and their responsibilities are described as follows.

- System Manager (SM): The SM is responsible for generating system parameters and defining roles. When a new role is added in the system, the SM generates a public/private key pair for the role, and keeps the private role key secret.
- Role Manager (RM): The RM is responsible for assigning roles to users. As mentioned above, each role is corresponding to a PKG as in the IBS scheme, but it is unpractical to build as many PKGs as roles. In our scheme, the RM receives all of the role's private keys securely from the SM and uses them to issue assigned keys while assigning corresponding roles to users.
- Access control Enforcement Facility (AEF) and **Access control Decision Facility (ADF):** The AEF and the ADF are responsible for managing the system's resources. The AEF mediates access request, and passes the user's notation to the ADF. The ADF makes the access control decisions based on the system security policies. The AEF enforces access decisions made by the ADF.

3.3 Framework

Definition 1. *Our scheme is specified by five algorithms (*Gen_{Sys}*,* Add_{Role}*,* $Asgn_{User}$*,* Gen_{Sig} *and* $Auth_{User}$*) such that:*

- Gen_{Sys}*: It takes as input the security parameter* k*, and returns system parameters.*
- Add_{Role}*: It takes as input a new role's identity. It generates a pair of public/private keys for the role.*
- $Asgn_{User}$*: It takes as input a user* $\mathcal{A}$*'s identity and a role* r_i*'s private key. It assigns* r_i *to* $\mathcal{A}$*, that is, it generates an assigned key for* $\mathcal{A}$ *corresponding to* r_i*.*
- Gen_{Sig}*: It takes as input* $\mathcal{A}$*'s identity, a set of assigned keys of* $\mathcal{A}$ *and an access request message* Q*. It generates a signature on* Q *for* $\mathcal{A}$*.*
- $Auth_{User}$*: It takes as input* $\mathcal{A}$*'s identity, a set of roles' public keys, an access request message* Q *and a signature for* $\mathcal{A}$*. It decides to allow* $\mathcal{A}$*'s access request or not.*

3.4 IRBAC Scheme

Our proposed scheme is driven from Cha-Cheon's identity-based signature scheme [7], we describe each algorithms of our scheme. We assume that all the users agree on a set of public parameters. The RM generates system parameters as follows.

Gen$_{Sys}$**:** the SM
Chooses a generator P of G_1, two hash functions $H_1 : \{0,1\}^* \rightarrow G_1$ and H_2 :

$\{0,1\}^* \times G_1 \rightarrow Z_q^*$. The SM also picks its master key $s \in Z_q^*$ at random and computes the system public key $P_{pub} = sP$.
The system public parameters are $params = \langle P, P_{pub}, H_1, H_2 \rangle$.

When a role r_i is added to the system, The SM carries out Add_{Role} as follows.

Add$_{Role}$**:** The SM
1. Picks a random $s_i \in Z_q^*$ as r_i's private key, and sets $P_i = s_iP$ as r_i's public key. If s_i is equal to other existing role's private key, the RM randomly picks another value from Z_q^* as r_i's private key.
2. Assigns specified permissions to r_i.The SM maintains a permission-assignment list (PAL) to record the assignment relationships between roles and permissions.
3. Sends (s_i, P_i) to the RM via secure channel.

In order to authorize users to access system resources, the RM must issue assigned keys stating the roles being granted. If a user A with identity ID_A wants to be a member of role r_i, he submits the request message to the RM. To assign r_i to A, the RM carries out Asgn_{User} as follows:

Asgn_{User}: The RM
1. Checks validity of A's identity.
2. Computes $Q_{ID_A} = H_1(ID_A)$.
3. Generates A's assigned key corresponding to $r_i : S_{ID_A r_i} = s_i Q_{ID_A}$, where s_i is r_i's private key.
4. Sends $S_{ID_A r_i}$ to A via secure channel.

We suppose that A wants to access system resources, he initiates a session by interacting with the AEF. Then A performs Gen_{Sig} as follows.

Gen$_{Sig}$**:** A
1. Selects a role or role set to activate in the current session, assume the activated role set is $AR = \{r_1, ..., r_k\}$.
2. Generates the query message Q and the signature $SigQ$ on Q using assigned keys corresponding to AR. Let $Q = ID_A|AR|p$, where ID_A is A's identity, p is the permission that A wants to enforce. To generate the signature on Q, A chooses a random number $r \in Z_q^*$ and computes:

a) $U = rQ_{ID_A}$.
b) $h = H_2(Q, U)$.
c) $S_{ID_A AR} = \sum_{i=1}^{k} S_{ID_A r_i}$, where $S_{ID_A r_i}$ is an assigned key of A corresponding to the role r_i.
d) $V = (r + h) S_{ID_A AR}$.

Signature: $SigQ = \langle U, V \rangle$.
3. Submits Q and $SigQ$ to the AEF.

After receiving Q and $SigQ$, the AEF and the ADF carries out Auth_{User} as follows.

Auth$_{User}$: The AEF
1. Checks the validity of $SigQ$ using ID_A and the public keys of $r_1, ..., r_k$. This can be accomplished by the equation below:

$\hat{e}(P, V) = \hat{e}(P_{AR}, U + hQ_{ID_A})$, where $h = H_2(Q, U)$, $P_{AR} = \sum_{i=1}^{k} P_i$, P_i is the public key of the role r_i.

2. The ADF maintains a permission-assignment list (PAL) to record the assignment relationships between roles and permissions. If $SigQ$ is valid, the ADF retrieves permissions assigned to the roles of AR, and makes the decision whether Alice's request should be allowed or denied according to the assigned permissions and system security policies. The ADF returns the decision to the AEF, and then the AEF enforces the ADF's decision.

For any valid signature produced by a user, we obtain

$$\begin{aligned}
&\hat{e}(P_{AR}, U + hQ_{ID}) \\
&= \hat{e}(\sum_{i=1}^{k} P_i, rQ_{ID} + hQ_{ID}) \\
&= \hat{e}(\sum_{i=1}^{k} sP, (r + h)Q_{ID}) \\
&= \hat{e}(P, (r + h) \sum_{i=1}^{k} s_i Q_{ID}) \\
&= \hat{e}(P, (r + h)S_{IDAR}) \\
&= \hat{e}(P, V)
\end{aligned}$$

So the correctness of our scheme can be easily verified.

Of course, we can choose other identity-based signature schemes as the basic signature scheme, such as [8,9,10].

3.5 Discussion

Our scheme has several advantages over the previous approaches [5,6]. Firstly, our scheme prevents a service from having to provide system resources to any users in an encrypted form, which can be an expensive task. Secondly, since the encryption-based access control method is avoided, our scheme fulfills the requirement of supporting multiple types of operations and objects in RBAC model. And thirdly, in our scheme, both aspects of the user authentication and checking the activated role's validity can be combined into one operation of verifying a signature of the user, so there is no need to check the user's identity in an independent procedure.

4 Security Analysis

4.1 Authenticity

Since an assigned key is generated as a function of a role's private key and a user's identity, it is uniquely corresponding to the user and the assigned role. The signature $SigQ$ is generated using the sum of assigned keys corresponding to the roles activated by the user, so the validity of $SigQ$ can prove the user's possession of the activated roles and authenticate the user's identity. There is no need to check the user's ID in an independent procedure.

4.2 Unforgeability

Our IRBAC scheme can be regarded as an identity-based signature scheme with multiple PKGs, each PKG is associated with a role. In order to activate role set $AR = \{r_1, ..., r_k\}$, a user has to generate a valid signature using the sum of assigned keys corresponding to all the roles of AR on the user's ID. We use similar technique in [7] to prove the unforgeability of our scheme. Suppose the hash functions H_1 and H_2 are random oracles. The following attack model appropriate to IRBAC scheme may be considered.

Definition 2. *We say that our IRBAC scheme is secure against existential forgery under adaptively chosen message and ID attack if no polynomial time adversary $\mathcal{A}$ has a non-negligible advantage against challenger $\mathcal{C}$ in the following game:*

1. Assume that performing specified permissions need to activate roles set $AR = \{r_1, ..., r_k\}$. Adversary $\mathcal{A}$ first chooses $k-1$ roles of AR which it wants to corrupt. Without loss of generality, let $SR = \{r_2, ..., r_k\}$ be the roles chosen by $\mathcal{A}$. $\mathcal{C}$ runs the System Setup algorithm and the resulting system parameters are given to $\mathcal{A}$.
2. $\mathcal{A}$ issues a number of the following queries as it wants, every request may depend on the answers to the previous ones:
- Hash Function Query: *$\mathcal{C}$ computes the value of the hash function for the requested input and sends the value to $\mathcal{A}$.*
- Extract Query: *$\mathcal{A}$ can issue two type of extract queries:*

a) $\mathcal{A}$ selects an identity ID and a role $r_i \in AR$, $\mathcal{C}$ returns the corresponding assigned key S_{IDr_i} which is obtained by running $Asgn_{User}$ algorithm.

b) $\mathcal{A}$ selects an identity ID, $\mathcal{C}$ returns the sum of all of assigned keys $\sum_{i=1}^{k} S_{IDr_i}$ (with $r_i \in AR$).

- Activate Query: *Given an identity ID and a message m, $\mathcal{C}$ returns a signature which is obtained by activating all the roles of AR, namely the signature is generated using the sum of all of assigned keys $\sum_{i=1}^{k} S_{IDr_i}$ (with $r_i \in AR$).*

3. $\mathcal{A}$ submits a target identity ID, such that ID is not equal to any input of Role Extract queries, and receives from $\mathcal{C}$ $k-1$ assigned keys S_{IDr_i} (with $r_i \in AR$) corresponding to the target ID.
4. Finally, $\mathcal{A}$ outputs (ID, m, σ), where ID is target identity chosen in phase 3, m is a message and σ is a signature such that (ID, m) is not equal to any input of Activate queries. $\mathcal{A}$ wins the game if σ is a valid signature of m using the sum of all assigned keys $\sum_{i=1}^{k} S_{IDr_i}$ (with $r_i \in AR$).

Our IRBAC scheme is based on Cha-Cheon's identity-based signature scheme, and Cha-Cheon's scheme is completely secure against existential forgery under adaptively chosen message and ID attack [7] in the random oracle model assuming the hardness of CDHP. The security proof of Cha-Cheon's scheme is given in [7].

Theorem 1. *Suppose that there exists a polynomial-time adversary $\mathcal{A}$ that can attack our scheme in the game described in Definition 2 with a non negligible advantage $Adv^{IRBAC}(\mathcal{A})$. Then we have an adversary $\mathcal{B}$ that is able to gain advantage $Adv^{CCIBS}(\mathcal{B}) = Adv^{IRBAC}(\mathcal{A})$ against Cha-Cheon's scheme under the adaptively chosen message and ID attack model.*

Proof. We use $\mathcal{A}$ to build algorithm $\mathcal{B}$ that can attack Cha-Cheon's scheme under the adaptively chosen message and ID attack model.

1. At first, $\mathcal{B}$ receives a random system parameter $K_{pub} = \langle G_1, G_2, \hat{e}, P, P_{pub}, H_1, H_2 \rangle$, which is generated by its challenger of Cha-Cheon's scheme. The system private key s is kept unknown to $\mathcal{B}$. $\mathcal{B}$ works by simulate $\mathcal{A}$'s environment as follows. $\mathcal{B}$ chooses $a \in Z_q^*$ randomly, and supplies $\mathcal{A}$ with the IRBAC system parameters $\langle G_1, G_2, \hat{e}, P, aP, H_1, H_2 \rangle$, where $G_1, G_2, \hat{e}, P, H_1, H_2$ are taken from K_{pub}. $\mathcal{B}$ informs $\mathcal{A}$ the role set $AR = \{r_1, ..., r_k\}$ to be activated. $\mathcal{A}$ chooses $k-1$ roles in AR it wants to corrupt, let $SR = \{r_2, ..., r_k\}$ be the roles chosen by $\mathcal{A}$. Then $\mathcal{B}$ randomly selects $s_i \in Z_q^* (i = 2, ..., k)$ as r_i's private key$(i = 2, ..., k)$, the corresponding role public key is $P_i = s_i P (i = 2, ..., k)$. Let r_1's private key $s_1 = s - \sum_{i=2}^{k} s_i$, public key $P_1 = P_{pub} - \sum_{i=2}^{k} P_i$. s_1 is kept unknown to $\mathcal{B}$. $\mathcal{B}$ sends $P_i (i = 1, ..., k)$ to $\mathcal{A}$.
2. $\mathcal{A}$ has access to the random H_1, H_2, Extract and Activate oracles. H_1 and H_2 are taken from Cha-Cheon's scheme, for every query made by $\mathcal{A}$ to random oracles H_1 and H_2, $\mathcal{B}$ forwards it to its challenger and sends the answer back to $\mathcal{A}$. $\mathcal{B}$ simulates the Extract oracle and Activate oracle as follows.

Extract-queries

a) $\mathcal{A}$ chooses a new ID_j, a role $r_i \in AR$, and issues an assigned key extract query. If $r_i \neq r_1$, $\mathcal{B}$ reply to $\mathcal{A}$ with $S_{ID_j r_i} = s_i H_1(ID_j)$. Otherwise, $r_i = r_1$, $\mathcal{B}$ forwards ID_j as its extract query to its challenger and gets the reply $S_{ID_j} = sH_1(ID_j)$. $\mathcal{B}$ computes $S_{ID_j r_1} = (s - \sum_{i=2}^{k} s_i) H_1(ID_j) = S_{ID_j} - \sum_{i=2}^{k} s_i H_1(ID_j)$, and returns $S_{ID_j r_1}$ to $\mathcal{A}$.

b) When $\mathcal{A}$ chooses a new ID_j , and query the sum of assigned keys corresponding to AR, $\mathcal{B}$ first forwards it to its Extract oracle and gets the reply $S_{ID_j} = sH_1(ID_j)$. $\mathcal{B}$ computes the sum of assigned keys $S_{ID_j}r_i$(with $r_i \in AR$) as: $S_{ID_jAR} = \sum_{i=1}^{k} S_{ID_jr_i} = \sum_{i=1}^{k} s_iH_1(ID_j) = sH_1(ID_j) = S_{ID_j}$, so $\mathcal{B}$ returns S_{ID_j} to $\mathcal{A}$.

Activate-queries

When $\mathcal{A}$ chooses (ID_j, m), and makes a query to the Activate oracle, since the signing structure of IRBAC is identical to Cha-Cheon's scheme and $S_{ID_jAR} = S_{ID_j}$, $\mathcal{B}$ forwards (ID_j, m) as its sign query to its challenger of Cha-Cheon's scheme, and returns the reply to $\mathcal{A}$.

3. At some point, $\mathcal{A}$ submits a target identity ID*. $\mathcal{B}$ generates $k-1$ assigned keys for ID* corresponding to SR as $S_{ID^*r_i} = s_iH_1(ID^*)(i = 2, ..., k)$, then sends $S_{ID^*r_i}(i = 2, ..., k)$ to $\mathcal{A}$. $\mathcal{B}$ also regards ID* as its own target identity.

4. Finally, $\mathcal{A}$ outputs (ID^*, m^*, σ^*). $\mathcal{B}$ also takes (ID^*, m^*, σ^*) as its output because $S_{ID^*AR} = sH_1(ID^*) = S_{ID^*}$ and IRBAC uses an identical signing structure to Cha-Cheon's scheme. From $\mathcal{A}$'s viewpoint, the above simulation is indistinguishable from the real protocol, and $\mathcal{B}$ is successful only if $\mathcal{A}$ is successful. Thus $Adv^{CCIBS}(\mathcal{B})$=$Adv^{IRBAC}(\mathcal{A})$.

5 Conclusion

In this paper, we apply identity-based signature technique to address user authentication problem in the role based access control systems. To achieve this, we extend the elements *user* and *role* in RBAC model to cooperate with identity-based cryptography. In our scheme, the manager can check the validity of a user's identity and activated roles simultaneously by verifying the user's signature, so the independent authentication procedure is eliminated. As we know our scheme is the first scheme that realizes user authentication and role-based access control in one operation using identity-based signature technique.

References

1. Sandhu, R., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-Based Access Control Models. IEEE computer 29(2), 38–47 (1996)
2. Ferraiolo, D.F., Sandhu, F., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST Standard for Role-Based Access Control. In: ACM Trans. Information and System Security, vol. 4(3), pp. 224–274. ACM Press, New York (2001)
3. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) Advances in Cryptology. LNCS, vol. 196, pp. 47–53. Springer, Berlin Heidelberg New York (1984)
4. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) Advances in Cryptology - CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)

5. Smart, N.P.: Access control using pairing based cryptography. In: Joye, M. (ed.) Topics in Cryptology - CT-RSA 2003. LNCS, vol. 2612, pp. 111–121. Springer, Heidelberg (2003)
6. Nali, D., Adams, C., Miri, A.: Using mediated identity-based cryptography to support role- based access control. In: Zhang, K., Zheng, Y. (eds.) Information Security. LNCS, vol. 3225, pp. 245–256. Springer, Heidelberg (2004)
7. Cha, J., Cheon, J.H.: An Identity-Based Signature from Gap Diffie-Hellman Groups. In: Desmedt, Y.G. (ed.) Public Key Cryptography - PKC 2003. LNCS, vol. 2567, pp. 18–30. Springer, Heidelberg (2002)
8. Hess, F.: Efficient identity based signature schemes based on pairings. In: Nyberg, K., Heys, H.M. (eds.) Selected Areas in Cryptography. LNCS, vol. 2595, pp. 310–324. Springer, Heidelberg (2003)
9. Paterson, K.G.: ID-based signatures from pairings on elliptic curves. Cryptology ePrint Archive, Report 2002/004, (2002) `http://eprint.iacr.org/2002/004`
10. Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems based on pairing. In: Symposium on Cryptography and Information Security-SCIS'00 (2000)
11. Public-Key Infrastructure(X.509), `http://www.ietf.org/html.charters/pkixcharter.html`
12. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. In: Boyd, C. (ed.) Advances in Cryptology - ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)