# An Improved ID-based Proxy Signature Scheme with Message Recovery

Article *in* International Journal of Security and its Applications · September 2015

DOI: 10.14257/ijsia.2015.9.9.14

Caixue Zhou
Jiujiang University

International Journal of Security and Its Applications
Vol.9, No.9 (2015), pp.151-164
http://dx.doi.org/10.14257/ijsia.2015.9.9.14

# An Improved ID-based Proxy Signature Scheme with Message Recovery

Caixue Zhou

*School of Information Science and Technology, University of Jiujiang, JiuJiang, 332005, JiangXi, P.R. China*
*Charlesjjjx@126.com*

## *Abstract*

*In 2012, Singh and Verma proposed an ID-based proxy signature scheme with message recovery. In this paper, we show that their scheme is vulnerable to the forgery attack, and an adversary can forge a valid proxy signature for any message with knowing a previous valid proxy signature. In addition, there is a security flaw in their proof. Furthermore, we propose an improved scheme that remedies the weakness of their scheme, and the improved scheme can be proved existentially unforgeable-adaptively chosen message and ID attack assuming the computational Diffie-Hellman problem is hard.*

**Keywords:** *identity-based proxy signature; signature with message recovery; bilinear pairing; unforgeability; forking lemma*

## 1. Introduction

The digital signature scheme with message recovery was first introduced by Nyberg and Ruppel [1] in 1993. In such a scheme, the original message of the signature is not required to be transmitted together with the signature since it can be recovered from the signature by the receiver. It is very suitable for small message to be signed or bandwidth to be one of the main concerns.

In 1984, Shamir [2] proposed the concept of ID-based cryptography. In this paradigm of cryptography, a user's identifier information such as his/her name, e-mail address or IP address can be used as a public key. Thus it simplifies the key management and removes the necessity of public key certificates.

Proxy signature was first proposed by Mambo et al. [3] in 1996, which allows a designated person, called proxy signer, to sign on behalf of an original signer on the message m. The proxy signature scheme plays an important role in many practical applications and has been received great attention after it was proposed. Proxy signature schemes can be used in distributed shared object systems [4], grid computing [5], mobile agent applications [6] and global distribution networks [7], etc.

In 2012, Singh and Verma [8] proposed an ID-based proxy signature scheme with message recovery, which combines the merits of ID-based signature scheme and signature scheme with message recovery, and they proved their scheme is existentially unforgeable under adaptively chosen message and ID attack. However, in this paper, we disprove their claim and show that their scheme is forgeable, and that anyone after getting a valid proxy signature can forge another proxy signature with any message under the same original signer and proxy signer. Meanwhile, we point out a security flaw in their proof. After that, we propose an improved scheme that remedies the weakness of Singh and Verma's scheme, and give the security proof of our improved scheme.

Proxy signature with message recovery can have many practical applications in real life. For example, a person wants to buy some goods or services on the Internet. There is a lot of work to do to find the suitable goods or services, so he/she will delegate a mobile

ISSN: 1738-9976 IJSIA

agent to help him/her do it. In order to achieve non-repudiation, the mobile agent must sign on such suitable goods or services. In this scenario, it can use our improved proxy signature with message recovery scheme to finish these tasks. Thanks to the message recovery feature, there is no need to transmit the messages, and thus the transmission bandwidth is saved.

The rest of this paper is organized as follows. In Section 2, we survey some related works. In Section 3, we give the preliminaries. In Section 4, we review the Singh and Verma's scheme, and show their scheme is forgeable, and point out a security flaw in their proof. In Section 5, we present an improved scheme to resist our attack, In Section 6, we give the security proof and efficiency analysis of our improved scheme. Finally, the conclusion and future work are given in Section 7.

## 2. Related Work

To categorize delegation types, Mambo, et al. [3] defined three levels of delegations: (1) Full delegation. The original signer gives his secret key to the proxy signer. The proxy signer uses the key to sign documents. So, it has the main weakness that the proxy signature cannot be distinguishable from the original signer's signature. (2) Partial delegation. The original signer generates a proxy signature key from its private key and gives it to the proxy signer. The proxy signer uses the proxy key to sign. However, the proxy signer can abuse his delegated rights because partial delegation does not restrict the proxy signer's signing capability. (3) Delegation by warrant. The original signer signs the warrant which describes relative rights and information of the original signer and proxy signer. The final proxy signature includes two parts: one is the signed warrant, and another is the proxy signature produced by the proxy signer. So the verifier must verify two signatures, which increases the amount of calculation. Later, Kim et al. [9] proposed a new kind of proxy signature: partial delegation with warrant. This kind of proxy signature combines the benefits of both the partial delegation and the delegation by warrant. So this delegation has fast processing speed and is appropriate for the restricting documents to be signed. Since then, most work on proxy signature schemes has focused on this type of proxy signature.

To restrict the power of the proxy signer, in 1997, Kim et al. [9] and Zhang [10] proposed the threshold proxy signature. A (t,n) threshold proxy signature scheme is a variant of the proxy signature scheme in which the proxy signature key is shared by a group of n proxy signers in such a way that any t or more proxy signers can cooperatively employ the proxy signature key to sign messages on behalf of an original signer, but t - 1 or fewer proxy signers cannot. In 2000, Hwang et al. [11] proposed the concept of multi-proxy signature, in such a scheme, an original signer can delegate his signing capability to a proxy group and only the cooperation of all members of the group can generate a proxy signature on behalf of the original signer. In the same year, Yi et al. [12] proposed the concept of proxy multi-signature, in such a scheme, a group of original signers can delegate their signing capability to a proxy signer. In 2001, Hwang et al. [13] proposed the concept of multi-proxy multi-signature, in such a scheme, a group of original signers can delegate their signing capabilities to a group of proxy signers. In 2003, Li et al. [14] proposed the concept of threshold proxy threshold signature, in which $t_1 - out - of - n_1$ original signers can cooperatively delegate the signing capabilities to a group of proxy signers, and $t_2 - out - of - n_2$ proxy signers can cooperatively produce the proxy signature on behalf of the original group.

In addition, proxy signature can combine other special signatures to obtain some new types of proxy signature, such as designated verifier proxy signature [15], proxy blind signature [16], forward security proxy signature [17], ID-based proxy signature [18] and certificateless proxy signature [19] et al.

The first work on proxy signature in the provable security direction was done by Boldyreva et al. [20] in 2003. They formalized the notion of security for proxy signature schemes in order to prove the security of proxy signature schemes under some well-established hard problems. Later, Herranz and Saez [21] extended Boldyreva et al.'s security model to analyze fully distributed proxy signatures. Malkin et al. [22] gave a security model for hierarchical proxy signatures. Schuldt et al. [23] further strengthened the proxy signatures security model by considering exposure arbitrary proxy signing keys. In addition, many other provably secure proxy signature schemes [24-26] have been proposed since then.

## 3. Preliminaries

### 3.1. Bilinear Pairing

Let $G_1$ be a cyclic additive group, whose order is a prime $q$ , and $G_2$ be a cyclic multiplicative group of the same order. Let $\hat{e}: G_1 \times G_1 \to G_2$ be a mapping with the following properties:

(1). Bilinearity: $\hat{e}(aP,bQ) = \hat{e}(P,Q)^{ab}$ for all $P,Q \in G_1$, $a,b \in Z_q$ .

(2). Non-degeneracy: There exists $P,Q \in G_1$ such that $\hat{e}(P,Q) \neq 1_{G2}$ .

(3). Computability: There exists an efficient algorithm to compute $\hat{e}(P,Q)$ , for all $P,Q \in G_1$ .

### 3.2. Computational Diffie-hellmen Problem (CDHP)

The CDH problem is, given $P,aP,bP \in G_1$ for unknown $a,b \in Z_q^*$, to compute $abP$ . The advantage of any probabilistic polynomial time (PPT) algorithm $G$ in solving CDH problem in $G_1$ is defined to be: $ADV_G^{CDH} = \Pr[G(P,aP,bP) = abP : a,b \in Z_q^*]$ .

CDH assumption: For every PPT algorithm $G$ , $ADV_G^{CDH}$ is negligible.

### 3.3. Syntax of ID-based Proxy Signature Scheme with Message Recovery

An ID-based proxy signature scheme with message recovery consists of the following eight polynomial time algorithms [8]: Setup, Extract, DelGen, DelVerify, PKgen, PSign, SignVerify/Message Recovery, ID.

(1) Setup: This algorithm takes as input a security parameter $\lambda$ and outputs the key generation center KGC's master key, global public key and system parameters params.

(2) Extract: An algorithm, which takes as input an identity $ID_A \in \{0,1\}^*$ of a user A and master key of KGC and then outputs the public key and private key pair $(q_A, d_A)$ .

(3) DelGen: In this algorithm, the original signer A computes the delegation $W_{A \to B}$ from his secret key $d_A$ and warrant $m_w$ and sends to the proxy signer in a secure way.

(4) DelVerify: The delegation verification algorithm, takes as input $ID_A \in \{0,1\}^*$ , $W_{A \to B}$ and verifies whether $W_{A \to B}$ is a valid delegation coming from A.

(5) PKGen: The proxy key generation algorithm, takes as input $W_{A \to B}$ and some secret information (for example the secret key of executer) and outputs a signing key $d_P$ for proxy signer.

(6) PSign: In this probabilistic algorithm, the proxy signer computes the proxy signature $\delta$ on a message $m \in \{0,1\}^l$ using the proxy signing key.

(7) SignVerify/Message Recovery: In this deterministic algorithm, the verifier receives the signature and takes the identity of original signer and the identity of

the proxy signer as input and then recovers the message and displays acceptance or rejection.

(8) ID: The proxy identification algorithm takes as input a valid proxy signature and outputs the identity of proxy signer.

### 3.4. Security Model of ID-based Proxy Signature Scheme with Message Recovery

We consider the security model described in Singh and Verma [8], in which an adversary A which is assumed to be a probabilistic Turing machine, takes as input the global scheme parameters and a random tape and performs an experiment, as described below.

**Definition 1.** For an ID-based proxy signature scheme with message recovery (IDPSWM), we define an experiment $Exp_A^{IDPSWM}(\lambda)$ of adversary A and security parameter $\lambda$ as follows:

(1) A challenger C runs setup and gives the system parameters $Params$ to A.

(2) Set $C_{list} \leftarrow \phi$, $D_{list} \leftarrow \phi$, $G_{list} \leftarrow \phi$, $S_{list} \leftarrow \phi$.

(3) Adversary A can make the following requests or queries adaptively:

- Extract(.): This oracle takes as input a user's $ID_i$, and returns the corresponding private key $d_i$. If A gets $d_i \leftarrow Extract(ID_i)$, let $C_{list} \leftarrow C_{list} \cup \{(ID_i, d_i)\}$.
- Delegate(.): This oracle takes as input the designator's identity ID and a warrant $m_w$ and output a delegation W. If A gets $W \leftarrow Delegate(ID, m_w)$, let $D_{list} \leftarrow D_{list} \cup \{(ID, m_w, W)\}$.
- PKGen(.): This oracle takes as input the proxy signer's identity ID and a delegation W and outputs a proxy signing key $d_P$. If A gets $d_P \leftarrow PKgen(ID, W)$, let $G_{list} \leftarrow G_{list} \cup \{(ID, W, d_P)\}$.
- PSign(.): This oracle takes as input the delegation W and message $m \in \{0,1\}^l$, and outputs a proxy signature created by proxy signer. If A gets $(m, \delta) \leftarrow PSign(W, m)$, let $S_{list} \leftarrow S_{list} \cup \{(W, m, \delta)\}$.

(4) A outputs $(ID, m_w, W)$ or $(W, m, \delta)$.

(5) If A's output satisfies one of the following terms, A's attack is successful.

- The output is $(ID, m_w, W)$, and satisfies: $DelVerify(W, ID) = 1$, $(ID,.) \notin C_{list}$, $(ID,.,.) \notin G_{list}$ and $(ID, m_w,.) \notin D_{list}$. Then $Exp_A^{IDPSWM}(\lambda)$ returns 1 otherwise returns 0.
- The output is $(W, m, \delta)$, and satisfies $SignVerify/MessageRecovery((m, \delta), ID_i) = 1$, $(W, m,.) \notin S_{list}$, $(ID_j,.) \notin C_{list}$, $(ID_j, W,.) \notin G_{list}$ ,where $ID_i$ and $ID_j$ are the identities of the designator and the proxy signer defined by W, respectively. Then $Exp_A^{IDPSWM}(\lambda)$ returns 2 otherwise returns 0.

**Definition 2.** An ID-based proxy signature scheme with message recovery IDPSWM is said to be existentially delegation and signature unforgeable under adaptively chosen message and ID attack (DS-EUF-ACMIA), if for any polynomial time adversary A, any polynomial p(.) and big enough $\lambda$, $\Pr[Exp_A^{IDPSWM} = 1] < 1/p(\lambda)$ and $\Pr[Exp_A^{IDPSWM} = 2] < 1/p(\lambda)$.

## 4. Review of Singh and Verma's Scheme

### 4.1. Notations

$a \| b$ : a concatenation of two strings a and b.

$\oplus$ : X-OR computation in the binary system.

$[x]_{10}$ : the decimal representation of $x \in \{0,1\}^*$ .
$[y]_2$ : the binary representation of $y \in Z$ .
${}_l|\beta|$: the first $l$ bits of $\beta$ from the left side.
$|\beta|_l$ : the first $l$ bits of $\beta$ from the right side.

### 4.2. Singh and Verma's Scheme

(1). Setup

It takes as input a security parameter $\lambda$ , and returns a master key s and system parameters $Params=(G_1,G_2,H_0,H_1,H_2,F_1,F_2,e,P,P_{pub},q,l_1,l_2)$ , where $G_1$ is an additive cyclic group of order q, $G_2$ is a multiplicative cyclic group of same order q. $H_0:\{0,1\}^* \to G_1^*$, $H_1:\{0,1\}^* \times G_2 \to Z_q$, $H_2:G_2 \to Z_q^*$, $F_1:\{0,1\}^{l_2} \to \{0,1\}^{l_1}$ and $F_2:\{0,1\}^{l_1} \to \{0,1\}^{l_2}$ are hash functions. $e:G_1 \times G_1 \to G_2$ is a bilinear pairing. $l_1,l_2$ are positive integers such that $l_1+l_2=|q|$ . $P \in G_1$ . $P_{pub}=sP$ is the global public key of PKG and q is a prime.

(2). Extract

It takes as input identity $ID_U \in \{0,1\}^*$ of user U, and computes $d_U=sH_0(ID_U)$ as secret key and $q_U=H_0(ID_U)$ as corresponding public key.

(3). Delegate

It takes as input the secret key $d_A$ of original signer and a warrant $m_w$ . Then the original signer selects $k_A \in_R Z_q^*$, computes $r_A=e(P,P)^{k_A}$ , $h_A=H_1(m_w,r_A)$ and $S=h_A d_A+k_A P$ , and outputs the delegation $W_{A\to B}=(m_w,r_A,S)$ .

(4). DelVerify

Once B receives $W_{A\to B}=(m_w,r_A,S)$ , he computes $h_A=H_1(m_w,r_A)$ , $q_A=H_0(ID_A)$ and accepts the delegation if and only if $e(S,P)=e(q_A,P_{pub})^{h_A} r_A$ .

(5). PKGen

If B accepts the delegation $W_{A\to B}=(m_w,r_A,S)$ , he computes the proxy signing key $d_P=h_A d_B+S$ , where $h_A=H_1(m_w,r_A)$ .

(6). PSign

Proxy signer B chooses $k_B \in_R Z_q^*$ and message $m \in \{0,1\}^{l_2}$ and computes the proxy signature $\delta=(r_A,V_B,m_w,U)$ , where
$r_B=e(P,P)^{k_B}$ , $v=r_A r_B$ , $\beta=F_1(m)\,||\,(F_2(F_1(m))\oplus m)$ , $\alpha=[\beta]_{10}$ , $V_B=H_2(v)+\alpha$ , $U=k_B P+d_P$

(7). SignVerify/Message Recovery

For a proxy signature $\delta=(r_A,V_B,m_w,U)$ , a recipient first checks if the proxy signer conforms to warrant $m_w$ . Then he computes the following: $h_A=H_1(m_w,r_A)$ , $\alpha=V_B-H_2(e(U,P)e(q_A+q_B,P_{Pub})^{-h_A})$ , $\beta=[\alpha]_2$ , and recovers $m'=F_2({}_{l_1}|\beta|)\oplus|\beta|_{l_2}$ . Then he accepts the signature and message $m'$ as valid if and only if ${}_{l_1}|\beta|=F_1(m')$ .

(8). ID

The proxy signer's identity $ID_B$ can be revealed by $m_w$.

### 4.3. A Forgery Attack on Singh and Verma's Scheme

In this section, we show that the Singh and Verma's scheme is vulnerable to the forgery attack. An adversary can forge a valid proxy signature for any message with knowing a previous valid proxy signature. Assume that $\delta=(r_A,V_B,m_w,U)$ is a valid proxy signature for message m, the adversary can forge a valid proxy signature $\delta=(r_A,\overline{V_B},m_w,U)$ for any message $\overline{m}$ as follows:

(1) $\beta=F_1(m)\,||\,(F_2(F_1(m))\oplus m)$ , $\alpha=[\beta]_{10}$ , $\overline{\beta}=F_1(\overline{m})\,||\,(F_2(F_1(\overline{m}))\oplus\overline{m})$ , $\overline{\alpha}=[\overline{\beta}]_{10}$ ,

(2) $\overline{V_B}=V_B-\alpha+\overline{\alpha}$

The following equations show that the proxy signature $\delta=(r_A,\overline{V_B},m_w,U)$ is valid for message $\overline{m}$ .

Because $\delta=(r_A,V_B,m_w,U)$ is a valid proxy signature,

so $\alpha=V_B-H_2(e(U,P)e(q_A+q_B,P_{Pub})^{-h_A})=V_B-H_2(v)$ ,

so $\overline{\alpha}=\overline{V_B}-V_B+\alpha=\overline{V_B}-V_B+(V_B-H_2(v))=\overline{V_B}-H_2(v)$ ,

so $\overline{\beta}=[\overline{\alpha}]_2=F_1(\overline{m})\,||\,(F_2(F_1(\overline{m}))\oplus\overline{m})$ .

so ${}_{l_1}|\overline{\beta}|=F_1(\overline{m})$ , $F_2(F_1(\overline{m}))\oplus\overline{m}=|\overline{\beta}|_{l_2}$ ,

so $\overline{m}^{'}=F_2({}_{l_1}|\overline{\beta}|)\oplus|\overline{\beta}|_{l_2}=F_2(F_1(\overline{m}))\oplus F_2(F_1(\overline{m}))\oplus\overline{m}=\overline{m}$ ,

So, Singh and Verma's ID-based proxy signature scheme is not secure mainly due to the reason that the part of $U$ is independent of message m. Once the attacker gets a valid proxy signature $\delta=(r_A,V_B,m_w,U)$, he can choose another message $\overline{m}$ , computes $\overline{V_B}$ , and combines with $r_A,m_w,U$ to produce another valid proxy signature.

### 4.4. The Security Flaw in the Proof of Singh and Verma's Scheme

In the proof of Singh and Verma's ID-based proxy signature scheme, they referenced the proof of Gu and Zhu [27], and they defined a generic digital signature scheme with message recovery, called IDWM. They stated if an attacker A could forge a valid ID-based proxy signature with message recovery of their scheme, the challenger B could forge a valid signature of IDWM. Because IDWM scheme was a generic digital signature, based on the forking lemma [28], B could produce two valid signatures, which makes B compute $aQ$ on input of any given $P,aP,Q\in G_1^*$ . Thus, their scheme was proved. Here, we must point out that forking lemma requires a generic digital signature scheme. Namely, by given the input message $m$ , it produces triples $(\sigma_1,h,\sigma_2)$ , where $\sigma_1$ randomly takes its values in a large set, $h$ is the hash value of $(m,\sigma_1)$ and $\sigma_2$ only depends on $\sigma_1$ , the message $m,h,$ and the private key of the signer. But in the Singh and Verma's scheme, the $\sigma_2$ (the U) only depends on $\sigma_1$ (the $k_B$ ) and the private key of the proxy signer, not the $h$ (the $V_B$ ) and the message $m$ , so their IDWM scheme is not a generic digital signature scheme, and the forking lemma is not suitable for it, meaning that their proof is wrong.

## 5. An Improved Scheme

### 5.1. The Scheme

Singh and Verma's scheme referenced the scheme of Zhang et al. [29]. In the original paper of Zhang et al.'s scheme, the $U$ part is dependent of message $m$ , so Zhang et al.'s

scheme is secure, and Singh et al.'s scheme is not secure. To resist our attack, we improve the scheme as follows.
(1)-(5),(8) are the same as Singh and Verma's scheme.

(6). PSign

Proxy signer B chooses $k_B \in_R Z_q^*$ and message $m \in \{0,1\}^{l_2}$ and computes the proxy signature $\delta = (r_A, V_B, m_w, U)$ , where $r_B = e(P,P)^{k_B}$ , $\beta = F_1(m) \| (F_2(F_1(m)) \oplus m)$ , $\alpha = [\beta]_{10}$ , $V_B = H_2(r_B) + \alpha$ , $U = k_B P + V_B d_P$

(7). SignVerify/Message Recovery

For a proxy signature $\delta = (r_A, V_B, m_w, U)$ , a recipient first checks if the proxy signer conforms to the warrant $m_w$ . Then he computes the following: $h_A = H_1(m_w, r_A)$ , $\alpha = V_B - H_2(e(U,P)e(q_A + q_B, P_{Pub})^{-V_B h_A} r_A^{-V_B})$ , $\beta = [\alpha]_2$ , and recovers $m' = F_2({}_{l_1}|\beta|) \oplus |\beta|_{l_2}$ . Then he accepts the signature and message $m'$ as valid if and only if ${}_{l_1}|\beta| = F_1(m')$
Correctness:

$$e(U,P)e(q_A + q_B, P_{Pub})^{-V_B h_A} r_A^{-V_B} = e(k_B P + V_B(h_A d_B + h_A d_A + k_A P), P)e(q_A + q_B, P_{Pub})^{-V_B h_A} r_A^{-V_B}$$
$$= e(P,P)^{k_B} e(h_A d_B + h_A d_A, P)^{V_B} e(k_A P, P)^{V_B} e(q_A + q_B, P_{Pub})^{-V_B h_A} r_A^{-V_B}$$
$$= e(P,P)^{k_B} e(q_A + q_B, P_{Pub})^{V_B h_A} r_A{}^{V_B} e(q_A + q_B, P_{Pub})^{-V_B h_A} r_A^{-V_B}$$
$$= e(P,P)^{k_B} = r_B ,$$

so, we obtain: $\alpha = V_B - H_2(e(U,P)e(q_A + q_B, P_{Pub})^{-V_B h_A} r_A^{-V_B}) = V_B - H_2(r_B)$ ,

so $\beta = [\alpha]_2 = F_1(m) \| (F_2(F_1(m)) \oplus m)$ . Now ${}_{l_1}|\beta| = F_1(m)$ , $F_2(F_1(m)) \oplus m = |\beta|_{l_2}$ ,

so $m' = F_2({}_{l_1}|\beta|) \oplus |\beta|_{l_2} = F_2(F_1(m)) \oplus F_2(F_1(m)) \oplus m = m$ , and ${}_{l_1}|\beta| = F_1(m')$ .

## 6. Security and Efficiency Analysis of the Improved Scheme

### 6.1. Security Analysis

**Theorem 1.** Let us denote our scheme by IDPSWM in the random oracle model, and let A be a polynomial time adversary, who manages an experiment $Exp_A^{IDPSWM}(\lambda)$ within a time bound $T$ , and gets return 1 by non-negligible probability $\varepsilon$ . Then there is an adversary B, who can succeed in existential forgery of Hess's scheme [30] with probability at least $\varepsilon$ . (Hess's identity-based signature scheme is proved to be existentially unforgeable under adaptively chosen message and ID attack in [30])

**Proof of the Theorem.** From A, we can construct an adversary B of Hess's scheme, who can succeed in existential forgery with probability at least $\varepsilon$ .
(1) A challenger C runs $Setup(1^\lambda)$ and gives the system parameters $Params$ to B.
(2) $C_{list} \leftarrow \phi$ , $D_{list} \leftarrow \phi$ , $G_{list} \leftarrow \phi$ , $S_{list} \leftarrow \phi$ .
(3)B gives A $Params$ and lets A manage $Exp_A^{IDPSWM}(\lambda)$ . During the execution, B emulates A's oracles as follows:

- $H_0(.)$ : For input $ID$ , B checks if $H_0(ID)$ defined, if not he defines $H_0(ID) = xP, x \in Z_q^*$ and returns $H_0(ID)$ to A
- $H_1(.)$ : If A makes a query $(m,r)$ to random oracle $H_1(.)$ , B checks if $H_1(m,r)$ is defined. If not, it picks a random $c_1 \in Z_q^*$ and sets $H_1(m,r) \leftarrow c_1$ . Then returns $H_1(m,r)$ to A.

- $H_2(.)$: If A makes a query $r$ to random oracle $H_2(.)$, B checks if $H_2(r)$ is defined. If not, it picks a random $c_2 \in Z_q^*$ and sets $H_2(r) \leftarrow c_2$. Then returns $H_2(r)$ to A.
- $Extract(.)$ : For input $ID$ , B lets $d_{ID} = xP_{Pub}$ be the reply to A and sets $C_{list} \leftarrow C_{list} \cup \{(ID, d_{ID})\}$
- $Delegate(.)$: For input $ID$, and warrant $m_w$, B uses $d_{ID} = xP_{Pub}$ as his private key to sign $m_w$, with Hess's signature scheme [30] and gets $(r_{0,}S_0)$. Let $W = (m_w, r_{0,}S_0)$ be the reply and sets $D_{list} \leftarrow D_{list} \cup \{(ID, m_w, W)\}$.
- $PKGen(.)$ : For input proxy signer's $ID_j$ and delegation $W = (m_w, r_0, S_0)$ , B computes $d_P = H_1(m_w, r_0)x_j P_{Pub} + S_0$ as the reply to A and sets $G_{list} \leftarrow G_{list} \cup \{(W, ID_j, d_P)\}$.
- $PSign(.)$: Let the input be $W = (m_w, r_0, S_0)$ and message $m$, designator's identity be $ID_i$ and proxy signer's identity be $ID_j$. B computes the proxy signature $(r_{P,} V_{P,} U_P)$ on $m$ with secret signing key $d_P = H_1(m_w, r_0)x_j P_{Pub} + S_0$ and returns $\delta = (r_0, V_P, m_w, U_P)$ as the reply to A. Let $S_{list} \leftarrow S_{list} \cup \{(W, m, \delta)\}$

(4)Let $S_{list}^{'}$ and $E_{list}$ be the query/answer lists coming from B's $Sign(.)$ oracle and $Extract(.)$ oracle respectively during the attack. If A's output is $(ID, m_w, W)$ and $Exp_A^{IDPSWM}(\lambda)$ returns 1. Let $W = (m_w, r_{0,}S_0)$ , B can output $(ID, m_w, r_0, S_0)$ satisfying $Verify((m_w, r_0, S_0), ID) = 1$ and $(ID, m_w, r_0, S_0) \notin S_{list}^{'}$, $(ID,.) \notin E_{list}$.

So we can see, if A manages $Exp_A^{IDPSWM}(\lambda)$ and gets 1 return by an non-negligible probability $\varepsilon$, B will succeed in his attack against Hess's scheme with probability no less than $\varepsilon$.

**Theorem 2.** Let us denote our scheme by IDPSWM in the random oracle model, and let A be a polynomial time adversary, who manages an experiment $Exp_A^{IDPSWM}(\lambda)$ within a time bound $T$, and gets return 2 by non-negligible probability $\varepsilon$. We denote respectively by $n_{h_0}, n_{h_1}, n_{h_2}$ , and $n_s$ the number of queries that A can ask to the random oracle $H_0(.), H_1(.), H_2(.)$ and the proxy signing oracle $PSign(.)$ . Assume that $\varepsilon \geq 10(n_s + 1)(n_s + n_{h_2})(n_{h_0} + n_{h_1})/q$, then there is an adversary B, who can solve CDHP within expected time less than $120686 n_s n_{h_0} n_{h_1} n_{h_2} T/\varepsilon$.

To prove the theorem, we can do the same as Gu and Zhu [27], and Singh and Verma [8]. That is, we can define a generic digital signature scheme with message recovery, called IDWM-NEW as follows:

- KeyGen. Given a security parameter $\lambda \in N$, generates the key pair as follows:
  1. $(s, params) \leftarrow (Setup(1^\lambda))$ ,
     where $params = (G_1, G_2, H_0, H_1, H_2, F_1, F_2, e, P, P_{pub}, q, l_1, l_2)$ , $P_{pub} = sP$ .Picks randomly $Q$, $q_A \in G_1^*$, and set $d_A = sq_A$, $d = sQ$.
  2. Picks a random $m_w \in \{0,1\}^*$ and use Hess's [30] signature scheme to compute the signature ( $m_w, r_A, U_A$ ) on $m_w$ with secret key $d_A$.
  3. Computes $h_A = H_1(m_w, r_A)$ and $d_P = h_A d + U_A$.
  4. The public key is $(G_1, G_2, H_0, H_1, H_2, F_1, F_2, e, q, P, P_{pub}, Q, q_A, m_w, h_A, r_A)$ and private key is $d_P$.
- Sign. To sign a message $m \in \{0,1\}^{|l_2|}$ , chooses $k_1 \in_R Z_q^*$ , and computes $r_p = e(P, P)^{k_1}$ , $\beta = F_1(m) \| (F_2(F_1(m)) \oplus m)$, $\alpha = [\beta]_{10}$, $V_B = H_2(r_p) + \alpha$, $U = k_1 P + V_B d_P$. Let $\delta = (r_A, V_B, m_w, U)$ be the signature of message m.

- Verify. For a signature with message recovery $\delta = (r_A, V_B, m_w, U)$, the recipient computes $\alpha = V_B - H_2(e(U,P)e(q_A + Q, P_{Pub})^{-V_B h_A} r_A^{-V_B})$, where $h_A = H_1(m_w, r_A)$ and $\beta = [\alpha]_2$ and verify the signature by checking $_{l_1}|\beta| = F_1(F_2(_{l_1}|\beta|) \oplus |\beta|_{l_2})$ and accepts $F_2(_{l_1}|\beta|) \oplus |\beta|_{l_2}$ as valid message if the equation holds.

This time, we use $U = k_1P + V_B d_P$ instead of $U = k_1P + d_P$, so in the triples $(\sigma_1, h, \sigma_2)$, $\sigma_2$ (the $U$ part) depends on $\sigma_1$ (the $k_1$), the message m, h(the $V_B$), and the private key of the signer, and so the IDWM-NEW is a generic digital signature scheme and forking lemma is suitable for it. We can use the same method to prove our improved scheme as Gu and Zhu [27], and Singh and Verma [8].

**Proof of the Theorem.** Without loss of generality, we may assume that for any $ID$, A queries $H_0(.)$ with $ID$ before $ID$ is used as (part of) an input of any query to Extract(.), Delegate(.), PKGen(.) and PSign(.) using a simple wrapper of A.

From the adversary A, we can construct a probabilistic algorithm B such that B computes $aQ$ on input $P, aP, Q \in G_1^*$ as follows:

(1) A challenger C runs $Setup(1^\lambda)$ to generate $params = (G_1, G_2, H_0, H_1, H_2, F_1, F_2, e, P, P_{pub}, q, l_1, l_2)$ and gives $Params$ to B.

(2) B sets $P_{Pub} \leftarrow aP$ and $i \leftarrow 1$.

(3) $C_{list} \leftarrow \phi$, $D_{list} \leftarrow \phi$, $G_{list} \leftarrow \phi$, $S_{list} \leftarrow \phi$.

(4) B picks randomly t, $1 \le t \le n_{h_0}$ and $x_i \in Z_q, i = 1, 2, ..., n_{h_0}$.

(5) B gives A $Params$ and lets A manage $Exp_A^{IDPSWM}(\lambda)$. During the execution, B emulates A's oracle as follows:

- $H_0(.)$: For input $ID$, B checks if $H_0(ID)$ defined, if not he defines $H_0(ID) = \begin{cases} Q, & i = t \\ x_i P, & i \ne t \end{cases}$ and sets $ID_i \leftarrow ID, i \leftarrow i+1$. B returns $H_0(ID)$ to A.
- $H_1(.)$: If A makes a query $(m, r)$ to random oracle $H_1(.)$, B checks if $H_1(m, r)$ is defined. If not, it picks a random $c_1 \in Z_q^*$ and sets $H_1(m, r) \leftarrow c_1$. Then returns $H_1(m, r)$ to A.
- $H_2(.)$: If A makes a query $r$ to random oracle $H_2(.)$, B checks if $H_2(r)$ is defined. If not, it picks a random $c_2 \in Z_q^*$ and sets $H_2(r) \leftarrow c_2$. Then returns $H_2(r)$ to A.
- $Extract(.)$: For input $ID_i$, if $i = t$, then B abort, otherwise, B lets $d_i = x_i P_{Pub}$ be the reply to A and sets $C_{list} \leftarrow C_{list} \cup \{(ID_i, d_i)\}$
- $Delegate(.)$: For input $ID_i$, and warrant $m_w$, if $i \ne t$, B uses $d_i = x_i P_{Pub}$ as his private key to sign $m_w$, with Hess's signature scheme [30] and gets $(r_{0,} S_0)$. Otherwise, B simulates $ID_i$'s proxy designation as follows:
    - ✧ Picks randomly $S_0 \in G_1$, $h_0 \in Z_q$
    - ✧ Computes $r_0 = e(S_0, P)e(Q, P_{Pub})^{-h_0}$
    - ✧ If A has made the query $(m_w, r_0)$ to $H_1(.)$, then B aborts (a collision appears). Otherwise B sets $H_1(m_w, r_0) = h_0$. Let $W = (m_w, r_{0,} S_0)$ be the reply and sets $D_{list} \leftarrow D_{list} \cup \{(ID_i, m_w, W)\}$.
- $PKGen(.)$: For input proxy signer's $ID_j$ and delegation $W = (m_w, r_0, S_0)$, if $j = t$, then B aborts. Otherwise, B computes $d_P = H_1(m_w, r_0) x_j P_{Pub} + S_0$ as the reply to A and sets $G_{list} \leftarrow G_{list} \cup \{(W, ID_j, d_P)\}$.

- $PSign(.)$: Let the input be $W=(m_w,r_0,S_0)$ and message $m$, designator's identity be $ID_i$ and proxy signer's identity be $ID_j$. If $j \neq t$ B computes the proxy signature $(r_{P,}V_{P,}U_P)$ on $m$ with secret signing key $d_P=H_1(m_w,r_0)x_jP_{Pub}+S_0$ and returns $\delta=(r_0,V_P,m_w,U_P)$ as the reply to A. Otherwise B simulates $ID_j$ 's proxy signature on behalf of $ID_i$ as follows:

✧ Picks randomly $U^{'} \in G_1, V \in Z$ such that $|V| \leq |q|$

✧ Checks whether $H_1(m_w,r_0)$ is defined. If not, request oracle $H_1(.)$ with $(m_w,r_0)$. Let $h=H_1(m_w,r_0)$.

✧ Computes $r_P=e(U^{'},P)(e(x_iP+Q,P_{Pub})^h r_0)^{-V}$ and $U_P=U^{'}$.

✧ If A has made the query $r_P$ to $H_2(.)$, he aborts (a collision appears). Otherwise he sets $H_2(r_P)=V$ , $V_P=H_2(r_P)+[\beta]_{10}$ ,where $\beta=F_1(m)\|(F_2(F_1(m))\oplus m)$.

✧ Let $\delta=(r_0,U_P,m_w,V_P)$ be the reply of $PSign(.)$. Let $S_{list} \leftarrow S_{list} \cup \{(W,m,\delta)\}$.

(6) If A's output $(W,m,\delta)=((m_w,r_0,S_0),m,(r_0,U_P,m_w,V_P))$ with designator's identity $ID_i$ and proxy signer's identity $ID_j$ , satisfying: $PVerify((m,\delta),ID_i)=1$ , $(W,m,.) \notin S_{list}$ , $(ID_j,.) \notin C_{list}$ , $(ID_j,W,.) \notin G_{list}$ and $j=t$ . B can get a forgery $(r_0,U_P,m_w,V_P)$ of IDWM-NEW corresponding to private key $d_P=haQ+S_0$ , where $h=H_1(m_w,r_0)$.

(7) If B has got two IDWM-NEW signatures corresponding to private key $d_P=haQ+S_0$: $(m,r_P,U_P,V)$ and $(m,r_P,U_P^{'},V^{'})$, B can computes and outputs $aQ$ as follows:

✧ $d_P \leftarrow (U-U^{'})(V-V^{'})^{-1}$

✧ $aQ \leftarrow h^{-1}(d_P-S_0)$

Otherwise, B sets $H_1(m_w,r_0)=h, i=1$ and go to step 5. During B's execution, if A manages an $Exp_A^{IDPSWM}(\lambda)$ and gets return 2, collision appears with negligible probability, as mentioned in [28]. So, B's simulations are indistinguishable from A's oracles. Because $t$ is chosen randomly, B can output a forgery of IDWM-NEW scheme corresponding to private key $d_P=haQ+S_0$ within expected time $T$ with probability $\varepsilon/n_{h_1}$ . IDWM-NEW scheme is a generic digital signature scheme based on Forking Lemma [28], B can produce two valid signatures $(m,r_P,U_P,V)$ and $(m,r_P,U_P^{'},V^{'})$, such that $V \neq V^{'}$ within expected time less than $120686 n_s n_{h_0} n_{h_1} n_{h_2} T/\varepsilon$ . So, B can output $aQ$. Thus the theorem is proved.

## 6.2. Efficiency Analysis

Since computation time and ciphertext size are two important factors affecting the efficiency, we present the comparison with respect to them. Table 1 shows the comparison. We denote by M a scalar multiplication in $G_1$, by E an exponentiation in $G_2$, by e the pairing computation.

From Table 1, it is clear that the full length of our message signature pair is the same as the original one, and less than other schemes considered, i.e., it is providing the benefit of being a message recovery signature scheme. In delegation phase, Xu *et al.* [31] is the most efficient one and the other schemes are almost the same. In delegation verification phase, Gu and Zhu [32] is the most efficient one because it only needs one pairing computation, and Xu et al. is the most inefficient one because it needs three pairing computations. In proxy key generation phase all schemes are the same. In proxy signing phase, our improved scheme and Zhang and Kim [33] need one more operation than the other three schemes. In signature verification phase, Gu and Zhu [32] is the most efficient one because it only needs one pairing computation, and Xu *et al.* [31] is the most inefficient one because it needs five pairing computations.

The improved scheme adds one scalar multiplication in Proxy Signing phase and one exponentiation in Signature Verification phase, comparing with the original one. To sum up, our improved scheme is of high efficiency.

**Table 1. Comparison with Other Proxy Signatures**

| Scheme | Total length | Delegate | DelVerify | PKGen | PSign | SignVerify |
|---|---|---|---|---|---|---|
| [33] | $\|m\|+\|m_w\|+1Z_q+1G_1+1G_2$ | $2M+1E$ | $2e+1E$ | $1M$ | $2M+1E$ | $2e+2E$ |
| [32] | $\|m\|+\|m_w\|+1G_1+2G_2$ | $1M+2E$ | $1e+2M+2E$ | $1M$ | $1M+1E$ | $1e+2M+2E$ |
| [31] | $\|m\|+\|m_w\|+3G_1$ | $2M$ | $3e$ | $1M$ | $2M$ | $5e+1E$ |
| [8] | $\|m_w\|+1Z_q+1G_1+1G_2$ | $2M+1E$ | $2e+1E$ | $1M$ | $1M+1E$ | $2e+1E$ |
| Ours | $\|m_w\|+1Z_q+1G_1+1G_2$ | $2M+1E$ | $2e+1E$ | $1M$ | $2M+1E$ | $2e+2E$ |

## 7. Conclusion and Future Work

In this paper, we show that Singh and Verma's ID-based proxy signature scheme with message recovery is insecure against the forgery attack. An adversary can forge a valid proxy signature for any message with knowing a previous valid proxy signature, and there is a security flaw in their proof. Our improved scheme can remedy the weakness of their scheme. Our improved scheme is a generic digital signature scheme, so forking lemma is suitable for it, and the improved scheme can be proved using the same method of Gu and Zhu [27], and Singh and Verma [8]. Efficiency analysis shows our improved scheme has high efficiency. The future work is to design proxy signature schemes with message recovery feature that are proven secure in the standard model.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 61462048) and the key program of Jiujiang University under Grant No. 2013ZD02.

## References

[1] K. Nyberg and R. A. Rueppel, "A new signature scheme based on the dsa giving message recovery", Proceedings of the 1st ACM conference on Computer and Communications Security, **(1993)** November 3-5; Fairfax, USA, pp. 58– 61.

[2] A. Shamir, "Identity based cryptosystems and signature", Proceedings of Crypto'1984, LNCS 196, **(1984)** August 19-22; Santa Barbara, USA, pp. 47–53.

[3] M. Mambo, K. Usuda and E. Okamoto, "Proxy signatures for deligating signing operation, Proceedings of the 3rd ACM Conference on Computer and Communication Security (CCS)", **(1996)** May 14-16; New Dehli, India, pp. 48–57.

[4] J. Leiwo, C. Hanle, P. Homburg and A. S. Tanenbaum, "Disallowing unauthorized state changes of distributed shared objects", Proceedings of Information Security for Global Information Infrastructures (SEC'2000), **(2000)** August 22-24; Beijing, China, pp. 381–390.

[5] I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, "A security architecture for computational grids", Proceedings of the Fifth ACM Conference on Computers and Communications Security, **(1998)** November 3-5; San Francisco, USA, pp. 83–92.

[6] H. Kim, J. Baek, B. Lee and K. Kim, "Secret computation with secrets for mobile agent using one-time proxy signature", Proceedings of Symposium on Cryptography and Information Security (SCIS'2001), **(2001)** January 23-26; Oiso, Japan, pp. 845–850.

[7] A. Bakker, M. Steen and A. S. Tanenbaum, "A law-abiding peer-to-peer network for free-software distribution", In: IEEE International Symposium on Network Computing and Applications (NCA 2001), **(2001)** October 8-10; Cambridge, USA, pp. 60–67.

[8] H. Singh and G. K. Verma, "ID-based proxy signature scheme with message recovery", The Journal of systems and software, vol. 85, **(2012)**, pp. 209-214.

[9] S. Kim, S. Park and D. Won, "Proxy signature, revisited, Proceedings of the First International Conference of Information and Communication Security (ICICS'1997)", LNCS 1334, **(1997)** November 11-14; Beijing, China, pp. 223-232.

[10] K. Zhang, “Threshold proxy signature schemes, Proceedings of the First International Workshop of Information Security (ISW 1997)”, **(1997)** September 17-19; Tatsunokuchi, Japan, pp. 191-197.

[11] S. J. Hwang and C. H. Shi, A simple multi-proxy signature scheme, Proceedings of the Tenth National Conference on Information Security, **(2000)**, Hualien, Taiwan, pp. 134-138.

[12] [12] L. J. Yi, G. Q. Bai and G. Z. Xiao, “Proxy multi-signature scheme: a new type of proxy signature scheme”, Electronics Letters, vol. 36, no. 6, **(2000)**, pp. 527–528.

[13] S. J. Hwang and C. H. Chen, “A new multi-proxy multi-signature scheme”, 2001 National Computer Symposium: Information Security, **(2001),** pp. 19-26.

[14] L. H. Li, S. F. Tzeng and M. S. Hwang, “Generalization of proxy signature based on discrete logarithms”, Computers and Security, vol. 22, no. 3, **(2003)**, pp. 245-255.

[15] C. L. Hsu and H. Y. Lin, “Pairing-based strong designated verifier proxy signature scheme with low cost”, Security and Communication Networks, vol. 5, no. 5, **(2012)**, pp. 517-522.

[16] D. M. Alghazzawi, T. M. Salim and S. H. Hasan, “A secure proxy blind signature scheme using ECC”, Proceedings of Communications in Computer and Information Science, **(2011)** July 11-13; Macau, China, pp. 47-52.

[17] J. He, X. M. Li, L. J. Li and C. M. Tang, “A new forward-secure proxy signature scheme”, Proceedings of 2010 International Forum on Information Technology and Applications (IFITA 2010), **(2010)** July 16-18; Kunming, China, pp. 30-33.

[18] S. C. Xie, “A special id-based proxy signature scheme from bilinear pairings”, International Conference on Communication Systems and Network Technologies, **(2012)** May 11-13; Rajkot, India, pp. 481-484.

[19] S. H. Seo, K. Y. Choi, J. Y. Hwang and S. Kim, « Efficient certificateless proxy signature scheme with provable security”, Information Science, vol. 188, **(2012)**, pp. 322-337.

[20] A. Boldyreva, A. Palacio and B. Warinschi, “Secure proxy signature schemes for delegation of signing rights”, http://eprint.iacr.org/2003/096.pdf, **(2003)**.

[21] J. Herranz and G. Saez, “Revisiting fully distributed proxy signature schemes”, http://eprint.iacr.org /2003/197.pdf, **(2003)**.

[22] T. Malkin, S. Obana and M. Yung, “The hierarchy of key evolving signatures and a characterization of proxy signatures”, Proceedings of Eurocrypt ’2004, LNCS 3027, **(2004)** May 2-6; Interlaken, Switzerland, pp. 306-322.

[23] J. C. Schuldt, K. Matsuura and K. G. Paterson, “Proxy signatures secure against proxy key exposure”, Proceedings of PKC ’2008, LNCS 4939, **(2008)** March 9-12; Barcelona, Spain, pp. 141-161.

[24] K. Gu, W. Jia and C. Jiang, “Efficient identity based proxy signature in the standard model”, Comput. J., **(2013)**, doi:10.1093/comjnl/bxt132.

[25] J. H. Zhang and Y. Yu, “Short computational diffie-hellman-based proxy signature scheme in the standard model’, International Journal of Communication Systems, vol. 27, no. 10, **(2014)**, pp. 1894–1907.

[26] Y. Yu, Y. Mu, W. Susilo, Y. Sun and Y. Ji, « Provably secure proxy signature scheme from factorization”, Mathematical and Computer Modelling, vol. 55, nos. 3-4, **(2012)**, pp. 1160-1168.

[27] C. X. Gu and Y. F. Zhu, “Probable security of id-based proxy signature schemes”, Proceedings of the Third International Conference of Networking and Mobile Computing (ICCNMC’2005), LNCS 3619, **(2005)** August 2-4; Zhangjiajie, China, pp. 1277–1286.

[28] D. Pointcheval and J. Stern, “Security arguments for digital signatures and blind signatures”, Journal of Cryptology, vol. 13, no. 3, **(2000)**, pp. 361–396.

[29] F. G. Zhang, W. Susilo and Y. Mu, “Identity based partial message recovery signature (or how to shorten id-based signature)”, Proceedings of 9th International Conference of Financial Cryptography and Data Security (FC’2005), LNCS 3570, **(2005)** February 28-March 3; Roseau, The Commonwealth of Dominica, pp. 45–56.

[30] F. Hess, “Efficient identity based signature schemes based on pairings”, Proceedings of Selected Areas in Cryptography 9th Annual International Workshop (SAC 2002), LNCS 2595, **(2003)** August 15-16; Newfoundland, Canada, pp. 310-324.

[31] J. Xu, Z. F. Zhang and D. G. Feng, “ID-based proxy signature using bilinear pairing”, Proceedings of Third International Symposium of Parallel and Distributed Processing and Applications (ISPA 2005). LNCS 3759, **(2005)** November 2-5; Nanjing, China, pp. 359–367.

[32] C. X. Gu and Y. F. Zhu, “An efficient id-based proxy signature scheme from pairing”, Proceedings of Information Security Cryptology. LNCS 4990, **(2008)** August 31-September 5; Xining, China, pp. 40-50.

[33] F. G. Zhang and K. Kim, “Efficient id-based blind signature and proxy signature from bilinear pairings”, Proceedings of 8th Australasian Conference of Information Security and Privacy (ACISP’2003), LNCS 2727, **(2003)** July 9-11; Wollongong, Australia, pp. 312-323**.**

# Author

**Caixue Zhou,** he received his B.A. degree in Computer Science Department from Fudan University, Shanghai, China in 1988, and his M.S. degree in Space College of Beijing University of Aeronautics and Astronautics, Beijing, China in 1991. He is an Association Professor in the School of Information Science and Technology, University of Jiujiang, China since 2007. He is a member of the CCF (China Computer Federation), and a member of CACR (Chinese Association for Cryptologic Research). His research interests include applied cryptography and network security.