A Principle for Cryptographic Protocols Beyond Security, Less Parameters

Zhengjun Cao
Département d'informatique, Université Libre de Bruxelles, Belgium
zhencao@ulb.ac.be or caoamss@gmail.com

Abstract Almost cryptographic protocols are presented with security arguments. None of them, however, did explain why a protocol should like this, not like that. The reason is that there are short of any principles for designing and analyzing cryptographic protocols. In this paper, we put forth such a principle beyond security, called *Less Parameters*, which says that the involved parameters should be reduced as less as possible. Actually, the principle ensures a protocol better cost. In different scenarios, the principle is not easy to grasp. Intuitively, we advise to introduce public parameters as less as possible. In the light of the principle, we investigate some signatures. We believe the techniques developed in this paper will be helpful to better some cryptographic protocols.

Keywords Less Parameters, Schnorr signature, group signature

1 Introduction

What are the fundamental requirements for a cryptographic protocol? So far, almost cryptographic protocols are presented with security arguments. None of them, however, did explain why a protocol should like this, not like that. That is, the psychological activities relating to design a cryptographic protocol have always been unveiled. The reason of the phenomenon, we think, is that there are short of any principles for designing cryptographic protocols. In this paper, we put forth such a principle beyond security. It is called *Less Parameters*, which says that the involved parameters should be reduced as less as possible. Actually, the Less Parameters ensures a protocol better cost. In different scenarios, the principle is not easy to grasp. Intuitively, we advise to introduce public parameters as less as possible. In the light of the principle, we investigate some signatures [3, 5, 7, 8, 10]. We believe the techniques developed in this paper will be helpful to better some cryptographic protocols.

2 Preliminary

2.1 The Schnorr signature

The Schnorr signature [8] is popular with researchers. The scheme can be described as follows.

Setup Pick a large prime p and $g \in \mathbb{Z}_p^*$ with a large prime order q. Pick $x \stackrel{R}{\leftarrow} \mathbb{Z}_q^*$. Compute $y = g^x \bmod p$ (For convenience, we will omit the notations $\bmod p$ and $\bmod q$ later). Choose a hash function $\mathcal{H} : \{0,1\}^* \longrightarrow \mathbb{Z}_q$. Set the public key as $\{p, q, g, y, \mathcal{H}\}$, the private key as $\{x\}$.

Sign For a message m, pick $k \stackrel{R}{\leftarrow} \mathbb{Z}_q^*$. Compute $c = \mathcal{H}(m||g^k),\ a = xc + k$. Output the signature (a, c) for m.

Verification Check that $\mathcal{H}(m||g^a y^{-c}) = c$.

2.2 What are intractable in Schnorr signature

The security of Schnorr signature is based on two intractable problems:

(1) A one-way hash function is intractable;

(2) The Discrete Logarithm (DLog for short) is intractable.

In the scheme, the first intractable problem which has rarely been mentioned, is embodied by that the input of the hash function is *naturally independent* of its output (usually called *challenge value*). Concretely, suppose that $g^a y^{-c} = g^\delta$, $\mathcal{H}(m||g^\delta) = c$, it requires that δ can be freely assigned. To further explain the subtle property, we investigate the following example.

Example 1 In the example, the Setup is the same as that of the Schnorr signature. To sign a message m, randomly choose secret $\alpha, k \in \mathbb{Z}_q^*$ and compute

$$c = \mathcal{H}(m||g^k),\ A = g^\alpha,\ b = xc + k - \alpha$$

The resulting signature for m is (A, b, c). The verification is $\mathcal{H}(m||Ag^b y^{-c}) = c$.

Notice that the scheme is not secure. Given a random challenge value c, an adversary solves $\mathcal{H}(m||Ag^b y^{-c}) = c$ for (A, b). Without less of generality, the adversary can set

$$A = g^{\delta_1} y^{\delta_2}$$

where δ_1, δ_2 are undetermined. Hence, $Ag^b y^{-c} = g^{\delta_1+b} y^{\delta_2-c}$. To ensure the input of the hash function, $g^{\delta_1+b} y^{\delta_2-c}$, is naturally independent of the output, c, he can set

$$\delta_2 = c$$

This leads to $Ag^b y^{-c} = g^{\delta_1+b} y^{\delta_2-c} = g^{\delta_1+b}$, which is independent of c because of both δ_1 and b can be freely assigned by the adversary, although the other parameter δ_2 should be assigned as c. Therefore, to forge a signature (A, b, c) for m, the adversary chooses random $\alpha, k \in \mathbb{Z}_q^*$ and computes

$$c = \mathcal{H}(m||g^k),\ A = g^\alpha y^c,\ b = k - \alpha$$

It is easy to verify that $Ag^b y^{-c} = (g^\alpha y^c) g^{k-\alpha} y^{-c} = g^k$.

2.3 The Schnorr signature is of optimal cost

In the light of the *Less Parameters* principle, the Schnorr signature is of optimal cost because: (1) the user's secret key consists of only one parameter x; (2) to blind the secret key x, only one parameter k is used; (3) The resulting signature consists of only a pair (a, c), where c is the challenge value which is necessary in such a cryptographic scenario. In a DLog-based cryptographic protocol, one random secret parameter is at lest required to blind the encrypted message or the user's secret key. To highlight the merit of Schnorr signature, we investigate the following example.

Example 2 In the example, the Setup is the same as that of the Schnorr signature. To sign a message m, randomly choose secret $k \in \mathbb{Z}_q^*$ and compute

$$c = \mathcal{H}(m||g^k||y^k),\ a = k + xc,\ B = y^{k+c}$$

The resulting signature for m is (a, B, c). The verification is $\mathcal{H}(m||g^a y^{-c}||By^{-c}) = c$.

Notice that the scheme in the Example 2 is as secure as the Schnorr signature scheme. In fact, the additional term By^{-c} equals to y^k. An adversary can not derive the session key k from the additional term. However, the term By^{-c} is not necessary since the secret key x is not used to compute B. To reduce the involved parameters as less as possible, it is better to remove B. Therefore, the corresponding term, By^{-c}, can be reasonably discarded.

See the Table 1 for the differences between the Schnorr signature, the scheme in the Example 1 and the scheme in the Example 2.

Table 1

	The Schnorr signature	The Example 1	The Example 2
Setup	$PK : \{p, q, g, y, \mathcal{H}\}$ $SK : \{x\}$	$PK : \{p, q, g, y, \mathcal{H}\}$ $SK : \{x\}$	$PK : \{p, q, g, y, \mathcal{H}\}$ $SK : \{x\}$
Sign	$k \xleftarrow{R} \mathbb{Z}_q^*, c = \mathcal{H}(m\|\|g^k),$ $a = xc + k$ $\sigma : \{m, a, c\}$	$\alpha, k \xleftarrow{R} \mathbb{Z}_q^*, c = \mathcal{H}(m\|\|g^k),$ $A = g^\alpha,\ b = xc + k - \alpha$ $\sigma : \{m, A, b, c\}$	$k \xleftarrow{R} \mathbb{Z}_q^*, c = \mathcal{H}(m\|\|g^k\|\|y^k),$ $a = k + xc,\ B = y^{k+c}$ $\sigma : \{m, a, B, c\}$
Verification	$\mathcal{H}(m\|\|g^a y^{-c}) = c$	$\mathcal{H}(m\|\|Ag^b y^{-c}) = c$	$\mathcal{H}(m\|\|g^a y^{-c}\|\|By^{-c}) = c$
Security	Yes	No	Yes

2.4 The Schnorr signature VS the Okamoto signature

The Okamoto signature [7] is a variation of the Schnorr signature, which extends the single secret key x to a tuple (x_1, x_2). We now describe it as follows.

Setup Pick a large prime p and $g_1, g_2 \in \mathbb{Z}_p^*$ with a large prime order q. Pick $x_1, x_2 \xleftarrow{R} \mathbb{Z}_q^*$. Compute $y = g_1^{x_1} g_2^{x_2} \bmod p$. Choose a hash function $\mathcal{H} : \{0, 1\}^* \longrightarrow \mathbb{Z}_q$. Set the public key as $\{p, q, g_1, g_2, y, \mathcal{H}\}$, the private key as $\{x_1, x_2\}$.

Sign For a message m, pick $k_1, k_2 \xleftarrow{R} \mathbb{Z}_q^*$. Compute $c = \mathcal{H}(m||g_1^{k_1} g_2^{k_2}),\ a_1 = x_1 c + k_1,\ a_2 = x_2 c + k_2$. Output the signature (a_1, a_2, c) for m.

Verification Check that $\mathcal{H}(m||g_1^{a_1} g_2^{a_2} y^{-c}) = c$.

See the Table 2 for the differences between the Schnorr signature and the Okamoto signature.

Table 2

	The Schnorr signature	The Okomoto signature
Setup	$PK : \{p, q, g, y = g^x, \mathcal{H}\}$ $SK : \{x\}$	$PK : \{p, q, g_1, g_2, y = g_1^{x_1} g_2^{x_2}, \mathcal{H}\}$ $SK : \{x_1, x_2\}$
Sign	$k \xleftarrow{R} \mathbb{Z}_q^*, c = \mathcal{H}(m\|\|g^k)$, $a = xc + k$ $\sigma : \{m, a, c\}$	$k_1, k_2 \xleftarrow{R} \mathbb{Z}_q^*, c = \mathcal{H}(m\|\|g_1^{k_1} g_2^{k_2})$, $a_1 = x_1 c + k_1, a_2 = x_2 c + k_2$ $\sigma : \{m, a_1, a_2, c\}$
Verification	$\mathcal{H}(m\|\|g^a y^{-c}) = c$	$\mathcal{H}(m\|\|g_1^{a_1} g_2^{a_2} y^{-c}) = c$

Apparently, the Okamoto signature is inefficient than the Schnorr signature. We here stress that the claim that the security assumptions for the Okamoto signature are weaker than those for the Schnorr signature [8], is not sound. Actually, the security of the Okamoto signature is reduced to the following assumptions:

(1) The hash function $\mathcal{H}$ is intractable, which is the same as that for the Schnorr signature.

(2) Both $\log_y g_1, \log_y g_2$ are intractable. It is a bit different from the assumption for the Schnorr signature that $\log_y g$ is intractable.

Definitely, the assumption both $\log_y g_1, \log_y g_2$ are intractable is more stronger than the assumption $\log_y g$ is intractable.

By the comparisons of the Schnorr signature and the Okamoto signature, we know it is better to introduce parameters as less as possible. (We'd like to stress that the Okamoto signature is more apt for constructing subliminal channels.) But in different scenarios, the principle is not easy to grasp. Intuitively, we advise to introduce public parameters as less as possible. According to the instruction, we will investigate some signature schemes in the sections that followed.

3 The investigation of the BBS04 group signature

Group signatures, introduced by Chaum and Heyst [6], allow individual members to make signatures on behalf of the group. Formally, a group signature should satisfy [1, 4]: *Unforgeability* Only group members are able to sign messages on behalf of the group. *Anonymity* Given a valid signature of some message, identifying the actual signer is computationally hard for everyone but the group manager. *Unlinkability* Deciding whether two different valid signatures were produced by the same group member is computationally hard. *Exculpability* Neither a group member nor the group manager can sign on behalf of other group member. *Traceability* The group manager is always able to open a valid signature and identity of the actual signer.

3.1 Review of the BBS04 group signature

In Crypto'2004, Boneh, Boyen, and Shacham [3] proposed a group signature (BBS04 for short). The scheme can be described as follows.

Setup Choose groups G_1, G_2 of prime order p with a bilinear map $e(\cdot,\cdot)$, and a hash function $\mathcal{H}$ with respective range $\mathbb{Z}_p$. Randomly pick generators g_1, g_2 in G_1, G_2. Pick $h \xleftarrow{R} G_1^*$, $\xi_1, \xi_2 \xleftarrow{R} \mathbb{Z}_p^*$, and set $u, v \in G_1$ such that $u^{\xi_1} = v^{\xi_2} = h$. Pick $\gamma \xleftarrow{R} \mathbb{Z}_p^*$, and set $\omega = g_2^{\gamma}$. Generate for each user $i, 1 \le i \le n$, a tuple (A_i, x_i) where $x_i \xleftarrow{R} \mathbb{Z}_p^*$, $A_i = g_1^{1/(\gamma+x_i)}$. The group public key is $gpk = \{G_1, G_2, e(\cdot,\cdot), p, \mathcal{H}, g_1, g_2, u, v, h, \omega\}$. The private key of the group manager is $gmsk = \{\xi_1, \xi_2\}$. Each user's private key is her tuple $gsk[i] = (A_i, x_i)$. No party is allowed to possess γ; it is only known to the private-key issuer.

Sign Given gpk, $gsk[i]$ and a message $m \in \{0,1\}^*$, it proceeds as follows.

1. Pick $\alpha, \beta, r_\alpha, r_\beta, r_x, r_{\delta_1}, r_{\delta_2} \xleftarrow{R} \mathbb{Z}_p^*$, and compute

$$T_1 = u^\alpha,\ T_2 = v^\beta,\ T_3 = Ah^{\alpha+\beta},\ \delta_1 = x\alpha,\ \delta_2 = x\beta$$
$$R_1 = u^{r_\alpha},\ R_2 = v^{r_\beta},\ R_4 = T_1^{r_x} \cdot u^{-r_{\delta_1}},\ R_5 = T_2^{r_x} \cdot v^{-r_{\delta_2}}$$
$$R_3 = e(T_3, g_2)^{r_x} \cdot e(h, \omega)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\delta_1} - r_{\delta_2}}$$
$$c = \mathcal{H}(m, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$$
$$s_\alpha = r_\alpha + \alpha c,\ s_\beta = r_\beta + \beta c,\ s_x = r_x + xc,\ s_{\delta_1} = r_{\delta_1} + \delta_1 c,\ s_{\delta_2} = r_{\delta_2} + \delta_2 c$$

2. Output the signature $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$ for m.

Verify Given gpk, m and σ, verify it as follows:

1. Compute

$$\widetilde{R}_1 = u^{s_\alpha} \cdot T_1^{-c},\quad \widetilde{R}_2 = v^{s_\beta} \cdot T_2^{-c},\quad \widetilde{R}_4 = T_1^{s_x} \cdot u^{-s_{\delta_1}},\quad \widetilde{R}_5 = T_2^{s_x} \cdot v^{-s_{\delta_2}}$$
$$\widetilde{R}_3 = e(T_3, g_2)^{s_x} \cdot e(h, \omega)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} \cdot (e(T_3, \omega)/e(g_1, g_2))^c$$

2. Check $c = \mathcal{H}(m, T_1, T_2, T_3, \widetilde{R}_1, \widetilde{R}_2, \widetilde{R}_3, \widetilde{R}_4, \widetilde{R}_5)$.

Open Verify that σ is a valid signature for m and recover $A = T_3/(T_1^{\xi_1} \cdot T_2^{\xi_2})$.

3.2 The BBS04 scheme is not a standard group signature

The BBS04 scheme is not a standard group signature because it requires an additional participant, the private-key issuer. By the equation $A_i = g_1^{1/(\gamma+x_i)}$, where (A_i, x_i) is the secret key for the group member i and the claim that γ is only known to the private-key issuer, we know in the scheme the private-key issuer who also knows (A_i, x_i) should be an *absolutely* trustworthy third party. That is, the BBS04 scheme is not of perfect Exculpability since the private-key issuer can sign on behalf of any group member. In the presence of an *absolutely* trustworthy third party, almost cryptographic protocols become easy to achieve. As for the role of a trustworthy third party in cryptographic protocols, we refer to [2]:

> A trustworthy third party is a disinterested third party trusted to complete a protocol. Trusted means that all people involved in the protocol accept what he says as true, what he does as correct, and that he will complete his part of the protocol.

Notice that a protocol with the presence of a trustworthy third party does not entail that the third party knows all private keys of the involved users.

In the later sections, we put aside the discussion about the reasonability of the model and focus on how to better its cost according to the Less Parameters principle.

3.3 The BBS04 scheme revisited

In the BBS04 scheme, the involved parameters are

$$gpk = \{G_1, G_2, e(\cdot,\cdot), p, \mathcal{H}, g_1, g_2, u, v, h, \omega\},\ gmsk = \{\xi_1, \xi_2\},\ gsk[i] = (A_i, x_i)$$

By $u^{\xi_1} = v^{\xi_2} = h, A = T_3/(T_1^{\xi_1} \cdot T_2^{\xi_2})$ and the singing procedure, we know u and v are used in parallel. Likewise, ξ_1 and ξ_2 are used in parallel, too. Intuitively, by the analysis of the Schnorr signature and the Okamoto signature we can discard $\{u, \xi_1\}$ or $\{v, \xi_2\}$. We now relate the case without $\{v, \xi_2\}$ as follows.

Setup Choose groups G_1, G_2 of prime order p with a bilinear map $e(\cdot,\cdot)$, and a hash function $\mathcal{H}$ with respective range $\mathbb{Z}_p$. Pick $h \xleftarrow{R} G_1^*$, $\xi_1 \xleftarrow{R} \mathbb{Z}_p^*$, and set $u \in G_1$ such that $u^{\xi_1} = h$. Pick $\gamma \xleftarrow{R} \mathbb{Z}_p^*$, and set $\omega = g_2^{\gamma}$. Generate for each user $i, 1 \le i \le n$, an SDH tuple (A_i, x_i) where $x_i \xleftarrow{R} \mathbb{Z}_p^*$, $A_i = g_1^{1/(\gamma+x_i)}$. The group public key is $gpk = \{G_1, G_2, e(\cdot,\cdot), p, \mathcal{H}, g_1, g_2, u, h, \omega\}$. The private key of the group manager is $gmsk = \{\xi_1\}$. Each user's private key is her tuple $gsk[i] = (A_i, x_i)$. No party is allowed to possess γ; it is only known to the private-key issuer.

Sign Given gpk, $gsk[i]$ and a message $m \in \{0,1\}^*$, it proceeds as follows.

1. Pick $\alpha, r_\alpha, r_x, r_{\delta_1} \xleftarrow{R} \mathbb{Z}_p^*$, and compute

$$\begin{aligned}
&T_1 = u^{\alpha},\ T_3 = Ah^{\alpha},\ \delta_1 = x\alpha,\\
&R_1 = u^{r_\alpha},\ R_4 = T_1^{r_x} \cdot u^{-r_{\delta_1}},\ R_3 = e(T_3, g_2)^{r_x} \cdot e(h,\omega)^{-r_\alpha} \cdot e(h, g_2)^{-r_{\delta_1}},\\
&c = \mathcal{H}(m, T_1, T_3, R_1, R_3, R_4),\\
&s_\alpha = r_\alpha + \alpha c,\ s_x = r_x + xc,\ s_{\delta_1} = r_{\delta_1} + \delta_1 c
\end{aligned}$$

2. Output the signature $\sigma = (T_1, T_3, c, s_\alpha, s_x, s_{\delta_1})$ for m.

Verify Given gpk, m and σ, verify it as follows:

1. Compute

$$\begin{aligned}
&\widetilde{R}_1 = u^{s_\alpha} \cdot T_1^{-c}, \quad \widetilde{R}_4 = T_1^{s_x} \cdot u^{-s_{\delta_1}}\\
&\widetilde{R}_3 = e(T_3, g_2)^{s_x} \cdot e(h,\omega)^{-s_\alpha} \cdot e(h, g_2)^{-s_{\delta_1}} \cdot (e(T_3,\omega)/e(g_1, g_2))^c
\end{aligned}$$

2. Check $c = \mathcal{H}(m, T_1, T_3, \widetilde{R}_1, \widetilde{R}_3, \widetilde{R}_4)$.

Open Verify that σ is a valid signature and recover $A = T_3/T_1^{\xi_1}$.

Correctness

$$\begin{aligned}
&\widetilde{R}_1 = u^{s_\alpha} \cdot T_1^{-c} = u^{s_\alpha - \alpha c} = u^{r_\alpha} = R_1\\
&\widetilde{R}_4 = T_1^{s_x} \cdot u^{-s_{\delta_1}} = T_1^{r_x + xc} u^{-r_{\delta_1} - \delta_1 c} = T_1^{r_x} (u^{\alpha})^{xc} u^{-r_{\delta_1} - x\alpha c} = T_1^{r_x} \cdot u^{-r_{\delta_1}} = R_4\\
&\widetilde{R}_3 = e(T_3, g_2)^{s_x} \cdot e(h,\omega)^{-s_\alpha} \cdot e(h, g_2)^{-s_{\delta_1}} \cdot (e(T_3,\omega)/e(g_1, g_2))^c\\
&= e(T_3, g_2)^{r_x + xc} \cdot e(h,\omega)^{-r_\alpha - \alpha c} \cdot e(h, g_2)^{-r_{\delta_1} - \delta_1 c} \cdot (e(T_3,\omega)/e(g_1, g_2))^c\\
&= R_3 \cdot \left[e(T_3, g_2)^{xc} \cdot e(h,\omega)^{-\alpha c} \cdot e(h, g_2)^{-\delta_1 c} \cdot (e(T_3,\omega)/e(g_1, g_2))^c \right]\\
&= R_3 \cdot \left[e(T_3, g_2)^{x} \cdot e(h,\omega)^{-\alpha} \cdot e(h, g_2)^{-\delta_1} \cdot e(T_3,\omega)/e(g_1, g_2) \right]^c\\
&= R_3 \cdot \left[e(T_3, g_2^x \omega) \cdot e(h,\omega)^{-\alpha} \cdot e(h, g_2)^{-x\alpha}/e(g_1, g_2) \right]^c
\end{aligned}$$

$$= R_3 \cdot \left[e(T_3, g_2^x\omega) \cdot e(h^{-\alpha}, g_2^x\omega)/e(g_1, g_2)\right]^c$$
$$= R_3 \cdot \left[e(T_3 h^{-\alpha}, g_2^x\omega)/e(g_1, g_2)\right]^c = R_3 \cdot \left[e(A, g_2^{x+\gamma})/e(g_1, g_2)\right]^c$$
$$= R_3 \cdot \left[e(g_1^{1/(\gamma+x)}, g_2^{x+\gamma})/e(g_1, g_2)\right]^c = R_3$$

Security The argument for the security of the revisited BBS04 scheme can be directly reduced to the other group signature proposed by Boneh and Shacham [5] (BS04 for short). For details, see the sections that followed.

3.4 Review of BS04 group signature

Setup Choose groups G_1, G_2 of prime order p with isomorphism ψ, a bilinear map $e(\cdot,\cdot)$ and hash functions $\mathcal{H}_0$ and $\mathcal{H}$, with respective ranges G_2^2 and $\mathbb{Z}_p$. Randomly pick a generator $g_2 \in G_2$, and set $g_1 \leftarrow \psi(g_2)$. Pick $\gamma \xleftarrow{R} \mathbb{Z}_p^*$ and set $\omega = g_2^\gamma$. Using γ, generate for each user an SDH tuple (A_i, x_i) by selecting $x_i \xleftarrow{R} \mathbb{Z}_p^*$ such that $\gamma + x_i \neq 0$, and set $A_i \leftarrow g_1^{1/(\gamma+x_i)}$. The group public key is $gpk = \{G_1, G_2, p, \psi, g_1, g_2, \omega, e(\cdot,\cdot), \mathcal{H}_0, \mathcal{H}\}$. Each user's private key is her tuple $gsk[i] = (A_i, x_i)$. The revocation token corresponding to a user's key (A_i, x_i) is $grt[i] = A_i$. No party is allowed to possess γ; it is only known to the private-key issuer.

Sign Given a message $m \in \{0,1\}^*$, it proceeds as follows.

S1. Pick a nonce $r \xleftarrow{R} \mathbb{Z}_p^*$. Obtain generators $(\hat{u}, \hat{v})$ in G_2^2 from $\mathcal{H}_0$ as $(\hat{u}, \hat{v}) \leftarrow \mathcal{H}_0(gpk, m, r)$, and compute their images in G_1: $u \leftarrow \psi(\hat{u}), v \leftarrow \psi(\hat{v})$.

1. Pick $\alpha, r_\alpha, r_x, r_\delta \xleftarrow{R} \mathbb{Z}_p^*$ and compute

$$T_1 = u^\alpha, T_2 = A_i v^\alpha, \delta = x_i\alpha,$$
$$R_1 = u^{r_\alpha}, R_3 = T_1^{r_x} u^{-r_\delta}, R_2 = e(T_2, g_2)^{r_x} \cdot e(v, \omega)^{-r_\alpha} \cdot e(v, g_2)^{-r_\delta},$$
$$c = \mathcal{H}(gpk, m, r, T_1, T_2, R_1, R_2, R_3),$$
$$s_\alpha = r_\alpha + \alpha c, s_x = r_x + x_i c, s_\delta = r_\delta + \delta c$$

2. Output the signature $\sigma = (r, T_1, T_2, c, s_\alpha, s_x, s_\delta)$ for m.

Verify Given gpk, m, σ and a set RL of revocation tokens, verify it as follows.

V1. Compute $(\hat{u}, \hat{v})$ in G_2^2 from $\mathcal{H}_0$ as $(\hat{u}, \hat{v}) \leftarrow \mathcal{H}_0(gpk, m, r)$, and compute their images in G_1: $u \leftarrow \psi(\hat{u}), v \leftarrow \psi(\hat{v})$.

1. Compute

$$\widetilde{R}_1 = u^{s_\alpha}/T_1^c, \widetilde{R}_3 = T_1^{s_x} u^{-s_\delta}, \widetilde{R}_2 = e(T_2, g_2)^{s_x} e(v, \omega)^{-s_\alpha} e(v, g_2)^{-s_\delta} (e(T_2, \omega)/e(g_1, g_2))^c$$

Check that $c = H(gpk, m, r, T_1, T_2, \widetilde{R}_1, \widetilde{R}_2, \widetilde{R}_3)$. If it is, accept. Otherwise, reject.

2. For each element $A \in RL$, check wether A is encoded in (T_1, T_2) by checking if: $e(T_2/A, \hat{u}) \stackrel{?}{=} e(T_1, \hat{v})$. If no element of RL is encoded in (T_1, T_2), the signer of σ has not been revoked. The algorithm outputs valid if both phases accepts, invalid otherwise.

Remark 1 Notice that the nonce r is equally used in the phase S1 and V1. That is, the security of scheme is independent of the nonce r. For simplification, it is better to set the corresponding u, v in the Setup.

3.5 The revisited BBS04 scheme VS the simplified BS04 scheme

To investigate the similarities between the revisited BBS04 scheme and the simplified BS04 scheme, we will rewrite some subscripts and notations. For details, see the Table 3.

Table 3

	Revisited BBS04 scheme	Simplified BS04 scheme
Setup	$gpk = \{G_1, G_2, e(\cdot,\cdot), p, \mathcal{H}, g_1, g_2, u, v, \omega\}$ $\omega = g_2^\gamma, u^{\xi_1} = v,\ gmsk = \{\xi_1\}$, $x_i \xleftarrow{R} \mathbb{Z}_p^*, A_i \leftarrow g_1^{1/(\gamma+x_i)}$ $gsk[i] = (A_i, x_i)$ γ is only known to the private-key issuer	$gpk = \{G_1, G_2, e(\cdot,\cdot), p, \mathcal{H}, \psi, g_1, g_2,$ $u, v, \hat{u}, \hat{v}, \omega\}$ $\omega = g_2^\gamma, g_1 = \psi(g_2), u = \psi(\hat{u}), v = \psi(\hat{v})$ $x_i \xleftarrow{R} \mathbb{Z}_p^*, A_i \leftarrow g_1^{1/(\gamma+x_i)}$ $gsk[i] = (A_i, x_i)$ γ is only known to the private-key issuer
Sign	For m, pick $\alpha, r_\alpha, r_x, r_\delta \xleftarrow{R} \mathbb{Z}_p^*$, compute $T_1 = u^\alpha, T_2 = A_i v^\alpha, \delta = x_i\alpha$, $R_1 = u^{r_\alpha}, R_3 = T_1^{r_x} u^{-r_\delta}$, $R_2 = e(T_2, g_2)^{r_x} \cdot e(v,\omega)^{-r_\alpha} \cdot e(v, g_2)^{-r_\delta}$, $c = \mathcal{H}(m, T_1, T_2, R_1, R_2, R_3)$, $s_\alpha = r_\alpha + \alpha c, s_x = r_x + x_i c, s_\delta = r_\delta + \delta c$, Output $\sigma = (T_1, T_2, c, s_\alpha, s_x, s_\delta)$	For m, pick $\alpha, r_\alpha, r_x, r_\delta \xleftarrow{R} \mathbb{Z}_p^*$, compute $T_1 = u^\alpha, T_2 = A_i v^\alpha, \delta = x_i\alpha$, $R_1 = u^{r_\alpha}, R_3 = T_1^{r_x} u^{-r_\delta}$, $R_2 = e(T_2, g_2)^{r_x} \cdot e(v,\omega)^{-r_\alpha} \cdot e(v, g_2)^{-r_\delta}$, $c = \mathcal{H}(gpk, m, T_1, T_2, R_1, R_2, R_3)$, $s_\alpha = r_\alpha + \alpha c, s_x = r_x + x_i c, s_\delta = r_\delta + \delta c$, Output $\sigma = (T_1, T_2, c, s_\alpha, s_x, s_\delta)$
Verify	Compute $\widetilde{R}_1 = u^{s_\alpha}/T_1^c, \widetilde{R}_3 = T_1^{s_x} u^{-s_\delta}$, $\widetilde{R}_2 = e(T_2, g_2)^{s_x} e(v,\omega)^{-s_\alpha} e(v, g_2)^{-s_\delta}$ $\cdot(e(T_2,\omega)/e(g_1, g_2))^c$ Check $c = H(m, T_1, T_2, \widetilde{R}_1, \widetilde{R}_2, \widetilde{R}_3)$	Compute $\widetilde{R}_1 = u^{s_\alpha}/T_1^c, \widetilde{R}_3 = T_1^{s_x} u^{-s_\delta}$, $\widetilde{R}_2 = e(T_2, g_2)^{s_x} e(v,\omega)^{-s_\alpha} e(v, g_2)^{-s_\delta}$ $\cdot(e(T_2,\omega)/e(g_1, g_2))^c$ Check $c = H(gpk, m, T_1, T_2, \widetilde{R}_1, \widetilde{R}_2, \widetilde{R}_3)$ Revocation: For each element $A \in RL$, check $e(T_2/A, \hat{u}) \stackrel{?}{=} e(T_1, \hat{v})$
Open	Recover $A = T_2/T_1^{\xi_1}$	

In the revisited BBS04 scheme, it sets a group manager secret key ξ_1 such that $u^{\xi_1} = v$, which is used to recover the signer for a signature by $A = T_2/T_1^{\xi_1}$. In the simplified BS04 signature, it revokes the signer of a signature by searching for the token $A \in RL$ such that $e(T_2/A, \hat{u})$=$e(T_1, \hat{v})$. Apparently, the revisited BBS04 scheme can be directly derived from the simplified BS04 scheme. That is, the security of the revisited BBS04 scheme is reduced to that of the BS04 scheme.

4 The investigation of the YSM09 scheme

The identity-based cryptography is due to Shamir [9]. It aims to simplify the authentication of a public key by merely using an identity string as a certain user's public key. In the common identity-based cryptosystem, there is a trusted party, called the private key generator (PKG), who generates the secret key for each user's identity. As the PKG generates and holds the secret key for all users, a complete trust must be placed on the PKG. Clearly, this may not be desirable in a real world scenario because a malicious PKG can impersonate users. This is known as the key escrow problem.

In EuroPKI'2009, Yuen et al [10] proposed an escrow-free identity-based signature scheme (YSM09 for short). In this model, each signer has his own public key and secret key. The PKG generates the identity-based secret key for the signer with respect to the user public key. Then, the signer uses both secret keys to sign a message. Therefore, the signer is protected against a malicious PKG. To verify the signature, it only requires the signer's identity, the system public key and the message. It is very impressive that the YSM09 scheme has to introduce six generators in the underlying group. We now review the scheme as follows.

4.1 Review of the YSM09 scheme

Setup Let G, G_T be groups of order prime p. $e : G \times G \rightarrow G_T$ is a bilinear mapping. Pick generators $g, u, v, g_0, g_1, g_2 \overset{R}{\leftarrow} G$. Choose hash functions $\mathcal{H}_1 : \{0,1\}^* \rightarrow Z_p^*$ and $\mathcal{H}_2 : \{0,1\}^* \rightarrow G$. The authority (PKG) who is responsible for keeping system secret parameters selects $\alpha \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and computes $g_a = g^\alpha$. Set the system public keys mpk as $\{e(\cdot,\cdot), G, G_T, p, g, u, v, g_0, g_1, g_2, g_a, \mathcal{H}_1, \mathcal{H}_2\}$. The system secret key msk is α (only known to the PKG).

Extract The user picks $x \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and sets $y = g^x, v' = v^x$. He also computes a non-interactive zero-knowledge (NIZK) proof $\sum$ of x with respect to v' and v. He sends v', ID, y, a joining proof Pf and the NIZK proof $\sum$ to the PKG. The PKG checks the validity of $Pf, \sum$. If so, the PKG computes

$$A = (uv'^{-1})^{\frac{1}{\alpha+i}}$$

where $i = \mathcal{H}_1(ID)$ and returns A to the user. The PKG stores the transcript $(v', \sum, ID, y, Pf)$. A, y, v' are viewed as the user's tokens (only known to the PKG and the user). The user's secret key usk is x (only known to the user).

Sign For a message m, the user with the identity ID picks $s, r, r_2 \overset{R}{\leftarrow} \mathbb{Z}_p^*, R_1 \overset{R}{\leftarrow} G$ and computes

$$\begin{aligned}
&t_0 = g_0^s, t_1 = Ag_1^s, t_2 = v^x g_2^s, \tau_0 = g_0^r, \tau_1 = R_1 g_1^r, i = \mathcal{H}_1(ID) \\
&\tau_2 = v^{r_2} g_2^r, \tau_3 = [e(g_1, g_a g^i) e(g_2, g)]^r, \tau_4 = e(g_2, \mathcal{H}_2(m))^r \\
&c = \mathcal{H}_3(t_0, t_1, t_2, \tau_0, \cdots, \tau_4, m, mpk, ID) \\
&z_0 = r - cs, Z_1 = R_1 A^{-c}, z_2 = r_2 - cx, S = e(v, \mathcal{H}_2(m))^x
\end{aligned}$$

Output the signature $\sigma = (t_0, t_1, t_2, c, z_0, Z_1, z_2, S)$.

Verify Given σ for m and the identity ID, compute

$$\begin{aligned}
&i = \mathcal{H}_1(ID), \widetilde{\tau}_0 = g_0^{z_0} t_0^c,\ \widetilde{\tau}_1 = Z_1 g_1^{z_0} t_1^c,\ \widetilde{\tau}_2 = v^{z_2} g_2^{z_0} t_2^c, \\
&\widetilde{\tau}_3 = [e(g_1, g_a g^i) e(g_2, g)]^{z_0} [e(t_1, g_a g^i) e(t_2, g) e(u, g)^{-1}]^c \\
&\widetilde{\tau}_4 = e(g_2, \mathcal{H}_2(m))^{z_0} [e(t_2, \mathcal{H}_2(m)) S^{-1}]^c
\end{aligned}$$

and check that

$$c = \mathcal{H}_3(t_0, t_1, t_2, \widetilde{\tau}_0, \cdots, \widetilde{\tau}_4, m, mpk, ID)$$

Blame Omitted (see the original description).

4.2 A simple analysis of the YSM09 scheme

First, the authors specified the hash function $\mathcal{H}_1, \mathcal{H}_2$, but forgot to specify the hash function $\mathcal{H}_3$. As we know, the intractability of $\mathcal{H}_3$ is very important to the security argument.

Second, it is very impressive that the YSM09 scheme has to introduce six generators $g, u, v, g_0, g_1, g_2 \in G$. Tracing the usage of the generator g_0,

$$\text{(in the Sign) } t_0 = g_0^s, \tau_0 = g_0^r, z_0 = r - cs, \text{ where } s, r \stackrel{R}{\leftarrow} \mathbb{Z}_p^*, c \text{ is a challenge value}$$
$$\text{(in the Verify) } \widetilde{\tau}_0 = g_0^{z_0} t_0^c = g_0^{r-cs} g_0^{cs} = g_0^r$$

we know the user's secret key x and the token A are definitely not involved. That means the generator g_0 is not necessarily introduced.

Third, tracing the usage of the picked random element $R_1 \in G$,

$$\text{(in the Sign) } \tau_1 = R_1 g_1^r, Z_1 = R_1 A^{-c},$$
$$\text{(in the Verify) } \widetilde{\tau}_1 = Z_1 g_1^{z_0} t_1^c = (R_1 A^{-c}) g_1^{z_0} (A g_1^s)^c = R_1 g^{z_0 + sc} = R_1 g_1^r = \tau_1$$

we know it is used to only blind A^{-c} instead of the token A. In view of the challenge value c is assumed to be random, the blinding element R_1 can be reasonably removed.

Finally, in view of that $z_0 = r - cs, z_2 = r_2 - cx$, where $s, r, r_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, x is the user's secret key and c is the challenge value, we can replace r_2 with r without any loss of security. That is, the quantity of the involved random numbers can be reduced as well.

4.3 An explicit analysis of the YSM09 scheme

In this section, we further investigate a concrete forging attempt. It will be helpful to understand why the scheme should like this, not like that.

Given the system public keys $e(\cdot,\cdot), G, G_T, p, g, u, v, g_0, g_1, g_2, g_a, \mathcal{H}_1, \mathcal{H}_2$, a message m and a random challenge value c, the adversary has to solve $c = \mathcal{H}_3(t_0, t_1, t_2, \widetilde{\tau}_0, \cdots, \widetilde{\tau}_4, m, mpk, ID)$ for $\sigma = (t_0, t_1, t_2, z_0, Z_1, z_2, S)$, where

$$\widetilde{\tau}_0 = g_0^{z_0} t_0^c, \ \widetilde{\tau}_1 = Z_1 g_1^{z_0} t_1^c, \ \widetilde{\tau}_2 = v^{z_2} g_2^{z_0} t_2^c,$$
$$\widetilde{\tau}_3 = [e(g_1, g_a g^i) e(g_2, g)]^{z_0} [e(t_1, g_a g^i) e(t_2, g) e(u, g)^{-1}]^c$$
$$\widetilde{\tau}_4 = e(g_2, \mathcal{H}_2(m))^{z_0} [e(t_2, \mathcal{H}_2(m)) S^{-1}]^c$$

(1) $\Rightarrow t_0, z_0$ (On generating t_0, z_0). By $\widetilde{\tau}_0 = g_0^{z_0} t_0^c$, the adversary can pick $\beta_1 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and set $t_0 = g_0^{\beta_1}$. He then has $\widetilde{\tau}_0 = g_0^{z_0} g_0^{c\beta_1} = g_0^{z_0 + c\beta_1}$. Taking $z_0 = \alpha_1 - c\beta_1$ where α_1 is freely assigned, $\widetilde{\tau}_0 = g_0^{\alpha_1}$ is independent of the challenge value c.

(2) $\Rightarrow Z_1$ (On generating Z_1). By $\widetilde{\tau}_1 = Z_1 g_1^{z_0} t_1^c$ and $z_0 = \alpha_1 - c\beta_1$, the adversary takes $Z_1 = (g_1^{\beta_1} t_1^{-1})^c$. Thus, $\widetilde{\tau}_1 = g_1^{\alpha_1}$ is independent of the challenge value c.

(3) $\Rightarrow z_2, t_2$ (On generating z_2, t_2). By $\widetilde{\tau}_2 = v^{z_2} g_2^{z_0} t_2^c$ and $z_0 = \alpha_1 - c\beta_1$, the adversary can pick $\beta_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and set $t_2 = g_2^{\beta_1} v^{\beta_2}$. Thus, $\widetilde{\tau}_2 = v^{z_2} g_2^{z_0} (g_2^{\beta_1} v^{\beta_2})^c = v^{z_2 + c\beta_2} g_2^{\alpha_1}$. Taking $z_2 = \alpha_1 - c\beta_2$, $\widetilde{\tau}_2 = g_2^{\alpha_1} v^{\alpha_1}$ is independent of the challenge value c.

(4) $\Rightarrow S$ (On generating S). By $\widetilde{\tau}_4 = e(g_2, \mathcal{H}_2(m))^{z_0}[e(t_2, \mathcal{H}_2(m))S^{-1}]^c$, $z_0 = \alpha_1 - c\beta_1$ and $t_2 = g_2^{\beta_1}v^{\beta_2}$, the adversary has

$$\widetilde{\tau}_4 = e(g_2, \mathcal{H}_2(m))^{\alpha_1}[e(g_2^{-\beta}t_2, \mathcal{H}_2(m))S^{-1}]^c = e(g_2, \mathcal{H}_2(m))^{\alpha_1}[e(v^{\beta_2}, \mathcal{H}_2(m))S^{-1}]^c$$

Taking $S = e(v^{\beta_2}, \mathcal{H}_2(m))$, $\widetilde{\tau}_4 = e(g_2, \mathcal{H}_2(m))^{\alpha_1}$ is also independent of the challenge value c.

(5) Can the adversary ensure $\widetilde{\tau}_3$ is independent of the challenge value c ? By

$$\widetilde{\tau}_3 = [e(g_1, g_a g^i)e(g_2, g)]^{z_0}[e(t_1, g_a g^i)e(t_2, g)e(u, g)^{-1}]^c, z_0 = \alpha_1 - c\beta_1, t_2 = g_2^{\beta_1}v^{\beta_2}$$

the adversary has

$$\begin{aligned}
\widetilde{\tau}_3 &= [e(g_1, g_a g^i)e(g_2, g)]^{z_0}[e(t_1, g_a g^i)e(t_2, g)e(u, g)^{-1}]^c \\
&= [e(g_1, g_a g^i)e(g_2, g)]^{\alpha_1}[e(g_1^{-\beta_1}t_1, g_a g^i)e(g_2^{-\beta_1}t_2, g)e(u, g)^{-1}]^c \\
&= [e(g_1, g_a g^i)e(g_2, g)]^{\alpha_1}[e(g_1^{-\beta_1}t_1, g_a g^i)e(v^{\beta_2}, g)e(u, g)^{-1}]^c \\
&= [e(g_1, g_a g^i)e(g_2, g)]^{\alpha_1}[e(g_1^{-\beta_1}t_1, g_a g^i)e(v^{\beta_2}u^{-1}, g)]^c
\end{aligned}$$

Now the adversary is confronting the following problem: giving $\{e(\cdot,\cdot), g_a, g, u, i, v, \beta_1, \beta_2\}$, solve

$$e(g_1^{-\beta_1}t_1, g_a g^i)e(v^{\beta_2}u^{-1}, g) = 1 \tag{1}$$

for t_1 (notice that β_1, β_2 can be freely assigned by the adversary). Hence, he has to solve

$$(g_1^{-\beta_1}t_1)^{\alpha+i} = v^{-\beta_2}u \text{ or } \log_{g_a g^i}(v^{\beta_2}u^{-1}) \tag{2}$$

for t_1. Without loss of generality, he can set $t_1 = g_1^{\beta_1}\theta$ where θ is undetermined. Thus he has to solve

$$\theta^{\alpha+i} = v^{-\beta_2}u \tag{3}$$

for θ, where α is only known to the PKG. We now consider the following cases:

1) The user who knows (A, x) such that $A^{\alpha+i} = v^{-x}u$ can simply set $\theta = A, \beta_2 = x$. Therefore, t_1 is of the form $g_1^{\beta_1}A$ and t_2 is of the form $g_2^{\beta_1}v^x$.

2) The PKG who knows α can simply set $\theta = (v^{-\beta_2}u)^{\frac{1}{\alpha+i}}$. Taking into account $S = e(v^{\beta_2}, \mathcal{H}_2(m))$, the forgery can be constrained by checking $S = e(v', \mathcal{H}_2(m))$. That is, the token y is not necessary for the Blame phase in the original scheme. Therefore, y can be discarded.

3) Removing the generator u and corresponding terms in the original scheme, the Eq.(3) becomes

$$\theta^{\alpha+i} = v^{-\beta_2} \tag{4}$$

Given the fixed i, v and α (only known to the PKG), the adversary can not generate proper θ and $\beta_2\,(\neq 0)$ satisfying Eq.(4). That is, the generator u can be reasonably discarded.

4.4 The YSM09 scheme revisited

Setup Let G, G_T be groups of order prime p. $e : G \times G \rightarrow G_T$ is a bilinear mapping. Pick generators $g, v, g_1 \stackrel{R}{\leftarrow} G$. Choose hash functions $\mathcal{H}_1 : \{0,1\}^* \rightarrow Z_p^*$, $\mathcal{H}_2 : \{0,1\}^* \rightarrow G$ and $\mathcal{H}_1$ respective $\mathbb{Z}_p^*$. The authority (PKG) who is responsible for keeping system secret parameters picks $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and computes $g_a = g^\alpha$. The system public keys mpk consist of $e(\cdot,\cdot), G, G_T, p, g, v, g_1, g_a, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$. The system secret key msk is α.

Extract The user picks $x \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and sets $v' = v^x$. He also computes a non-interactive zero-knowledge (NIZK) proof $\sum$ of x with respect to v' and v. He sends v', ID a joining proof Pf and the NIZK proof $\sum$ to the PKG. The PKG checks the validity of $Pf, \sum$. If so, the PKG computes

$$A = (v'^{-1})^{\frac{1}{\alpha+i}}$$

where $i = \mathcal{H}_1(ID)$ and returns A to the user. The PKG stores the transcript $(v', \sum, ID, Pf)$. A, v' are viewed as the user's tokens. The user's secret key usk is x.

Sign For a message m, the user with the identity ID picks $s, r \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and computes

$$\begin{aligned}
&i = \mathcal{H}_1(ID), t_1 = Ag_1^s, t_2 = v'g_1^s, \tau_1 = g_1^r \\
&\tau_2 = v^r g_1^r, \tau_3 = [e(g_1, g_a g^i)e(g_1, g)]^r, \tau_4 = e(g_1, \mathcal{H}_2(m))^r \\
&c = \mathcal{H}_3(t_1, t_2, \tau_1, \cdots, \tau_4, m, mpk, ID) \\
&z_0 = r - cs, Z_1 = A^{-c}, z_2 = r - cx, S = e(v', \mathcal{H}_2(m))
\end{aligned}$$

Output the signature $\sigma = (t_1, t_2, c, z_0, Z_1, z_2, S)$.

Verify Given σ for m and the identity ID, compute

$$\begin{aligned}
&i = \mathcal{H}_1(ID), \widetilde{\tau}_1 = Z_1 g_1^{z_0} t_1^c,\ \widetilde{\tau}_2 = v^{z_2} g_1^{z_0} t_2^c, \\
&\widetilde{\tau}_3 = [e(g_1, g_a g^i)e(g_1, g)]^{z_0}[e(t_1, g_a g^i)e(t_2, g)]^c \\
&\widetilde{\tau}_4 = e(g_1, \mathcal{H}_2(m))^{z_0}[e(t_2, \mathcal{H}_2(m))S^{-1}]^c
\end{aligned}$$

and check that

$$c = \mathcal{H}_3(t_1, t_2, \widetilde{\tau}_1, \cdots, \widetilde{\tau}_4, m, mpk, ID)$$

Blame Given σ for m and the identity ID, The judge checks its validity, then asks the PKG to provide the transcript $(v', \sum, ID, Pf)$ and checks that $S = e(v', \mathcal{H}_2(m))$. If it holds, the judge can affirm that the signature is produced by the user. If $S \neq e(v', \mathcal{H}_2(m))$, the judge can affirm that it is produced by the PKG.

Correctness

$$\begin{aligned}
\widetilde{\tau}_1 &= Z_1 g_1^{z_0} t_1^c = A^{-c} g_1^{r-cs}(Ag_1^s)^c = g_1^r = \tau_1 \\
\widetilde{\tau}_2 &= v^{z_2} g_1^{z_0} t_2^c = v^{r-cx} g_1^{r-cs}(v'g_1^s)^c = v^r g_1^r = \tau_2 \\
\widetilde{\tau}_3 &= [e(g_1, g_a g^i)e(g_1, g)]^{z_0}[e(t_1, g_a g^i)e(t_2, g)]^c = [e(g_1, g_a g^i)e(g_1, g)]^r[e(g_1^{-s}t_1, g_a g^i)e(g_1^{-s}t_2, g)]^c \\
&= [e(g_1, g_a g^i)e(g_1, g)]^r[e(A, g_a g^i)e(v', g)]^c = [e(g_1, g_a g^i)e(g_1, g)]^r = \tau_3 \\
\widetilde{\tau}_4 &= e(g_1, \mathcal{H}_2(m))^{z_0}[e(t_2, \mathcal{H}_2(m))S^{-1}]^c = e(g_1, \mathcal{H}_2(m))^r[e(g_1^{-s}t_2, \mathcal{H}_2(m))S^{-1}]^c \\
&= e(g_1, \mathcal{H}_2(m))^r[e(v', \mathcal{H}_2(m))S^{-1}]^c = e(g_1, \mathcal{H}_2(m))^r = \tau_4
\end{aligned}$$

Remark 2 In the revisited scheme, the generator g_2 is replaced with g_1. The change does not endanger its security. This can be directly derived from the following security argument.

Security To differ from the general arguments, we here present a short security argument for it. The presentation is more apt for unveiling the psychological activities during the investigation.

Without loss of generality, suppose that $z_0 = \xi_1 - c\rho_1, z_2 = \xi_2 - c\rho_2$, where $\xi_1, \xi_2, \rho_1, \rho_2$ are undetermined. By $\widetilde{\tau}_2 = v^{z_2} g_1^{z_0} t_2^c$, we have $\widetilde{\tau}_2 = v^{\xi_2} g_1^{\xi_1} (v^{-\rho_2} g_1^{-\rho_1} t_2)^c$. To ensure that $v^{\xi_2} g_1^{\xi_1} (v^{-\rho_2} g_1^{-\rho_1} t_2)^c$ is independent of c, ξ_1 and ξ_2 must be freely assigned, and t_2 must be of the form $v^{\rho_2} g_1^{\rho_1}$.

By $\widetilde{\tau}_3 = [e(g_1, g_a g^i) e(g_1, g)]^{z_0} [e(t_1, g_a g^i) e(t_2, g)]^c$, we have

$$\widetilde{\tau}_3 = [e(g_1, g_a g^i) e(g_1, g)]^{\xi_1} [e(t_1 g_1^{-\rho_1}, g_a g^i) e(t_2 g_1^{-\rho_1}, g)]^c$$

To ensure that $[e(g_1, g_a g^i) e(g_1, g)]^{\xi_1} [e(t_1 g_1^{-\rho_1}, g_a g^i) e(t_2 g_1^{-\rho_1}, g)]^c$ is independent of c, where ξ_1 is freely assigned, $e(t_1 g_1^{-\rho_1}, g_a g^i) e(t_2 g_1^{-\rho_1}, g)$ is constrained to 1. Hence,

$$e(t_1 g_1^{-\rho_1}, g_a g^i) e(v^{\rho_2}, g) = 1 \tag{5}$$

Since $\log_{g_a g^i}(v)$ is not known to anybody, the Eq.(5) becomes

$$(t_1 g_1^{-\rho_1})^{\alpha+i} = v^{-\rho_2} \tag{6}$$

Suppose that $t_1 = \lambda g_1^{\rho_1}$, where λ is undetermined. Thus

$$\lambda^{\alpha+i} = v^{-\rho_2} \tag{7}$$

By $\widetilde{\tau}_4 = e(g_1, \mathcal{H}_2(m))^{z_0} [e(t_2, \mathcal{H}_2(m)) S^{-1}]^c$, we have

$$\widetilde{\tau}_4 = e(g_1, \mathcal{H}_2(m))^{\xi_1} [e(t_2 g_1^{-\rho_1}, \mathcal{H}_2(m)) S^{-1}]^c = e(g_1, \mathcal{H}_2(m))^{\xi_1} [e(v^{\rho_2}, \mathcal{H}_2(m)) S^{-1}]^c$$

where ξ_1 is freely assigned. To ensure that the above equation is independent of c, one has to set

$$S = e(v^{\rho_2}, \mathcal{H}_2(m)) \tag{8}$$

Combining Eq.(7), Eq.(8), and a Blame phase, ρ_2 can be directly constrained to x. Consequently, λ is constrained to the token A.

5 Conclusion

In the past, the psychological activities relating to design a cryptographic protocol have always been unveiled. As a result, it becomes difficult to explain why a protocol should like this, not like that. Likewise, it is difficult to check whether a protocol is of better cost. The principle Less Parameters and some investigations presented in this paper will promote the techniques for designing and analyzing cryptographic protocols.

Acknowledgements We acknowledge the Cryptasc Project (Institute for the Encouragement of Scientific Research and Innovation of Brussels), the National Natural Science Foundation of China (Project 60873227), the Shanghai Leading Academic Discipline Project (S30104) and the Innovation Program of the Shanghai Municipal Education Commission.

References

[1] G. Ateniese, J.Camenisch, M. Joye, and G.Tsudik. A practical and provably secure coalition-resistant group signature scheme. In proceedings of Crypto'2000, LNCS 1880, pp. 255-270, Springer, 2000

[2] B. Schneier. Applied Cryptography Second Edition: Protocols, Algorthms, and Source Code in C, Wiley, 1996

[3] D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In proceedings of CRYPTO'2004, LNCS 3152, pp. 41-55, Springer, 2004

[4] M. Bellare, D. Micciancio, and Bogdan Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In proceedings of EUROCRYPT'2003, LNCS 2656, pp.614-629, Springer, 2003

[5] D. Boneh and H. Shacham. Group Signatures with Verifier-Local Revocation. In proceedings of the 11'th ACM conference on Computer and Communications Security (CCS), pp. 168-177, 2004

[6] D. Chaum and E. van Heyst. Group signatures, In proceedings of EUROCRYPT'1991, LNCS 950, pp. 257-265, Springer, 1992

[7] T. Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In proceedings of CRYPTO'1992, pp. 31-53, Springer, 1992

[8] C. Schnorr. Efficient signature generation for smart cards. In proceedings of CRYPTO'1989, LNCS 435, pp. 239-252, Springer, 1989

[9] A. Shamir. Identity-based cryptosystems and signature schemes. In proceedings of CRYPTO'1984, LNCS 196, pp. 47-53, Springer, 1984

[10] T. Yuen, W. Susilo, and Y. Mu. How to Construct Identity-Based Signatures without the Key Escrow Problem. In proceedings of EuroPKI'2009. (http://eprint.iacr.org/2009/421)