Journal of WoCrypt
www.uow.edu.au/~fuchun/jow

Volume: 2023 (52)

Practice Makes Perfect: Attack Signature Schemes *

Fuchun Guo, Willy Susilo

University of Wollongong, Australia
{fuchun, wsusilo}@uow.edu.au

Abstract

We collect insecure group-based signature schemes and simplify them for you to attack.

1 Introduction

There are at least four steps towards becoming a Master like WuGui who is believed to be good at constructing excellent cryptograpgic schemes. They are:

Step 1: Propose working but insecure schemes.

Step 2: Understand why they are insecure or know how to attack them.

Step 3: Have the ability of fixing insecurity.

Step 4: Can prove security for secure schemes after fixing them.

The step 2 is important. With this kind of knowledge, beginners can avoid similar mistakes in step 1 and know where to fix in the step 3. There have been many published schemes that are later showed to be insecure. The beginners can practice the step 2 with the help of those published schemes. Unfortunately, most of them are rather complicated and not friendly for beginners. We therefore decide to do something for beginners.

In this article, we have surveyed many insecure signature schemes, simplified their constructions while keeping mistakes, and presented them for beginners. We believe that this article will help beginners quickly master the skills of attacking schemes. A pity is that all collected schemes were constructed based on groups (with or without pairing). We don't know how to attack schemes from other foundations such as lattice, although the core uses similar tricks. We will collect insecure encryption schemes and publish them in the next article.

To make sure that the new section will start from the page, we decide to say something else. We observed that beginners do like simplified schemes and proofs for practice or understanding. For example, many students do like the simplified proofs (security reduction) for signature schemes and encryption schemes collected in the book "Introduction to Security Reduction". :)

*We decided to collect insecure schemes in 2022 but don't have time to finish this project until recently.

2 Preliminaries

We assume that you have known the essential foundations for scheme constructions including hash function, cyclic group, and bilinear pairing. A signature scheme is composed of three algorithms $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$. The standard security model is EUF-CMA. Many schemes in the literature are not secure in this security model. That is, an adversary is able to forge a valid signature in this security model. We also note that some schemes are even insecure in weaker security models. For example, given a public key, the adversary can forge a valid signature without signature queries.

If you cannot quickly understand the above preliminaries, it means that you are not yet ready for attacking schemes. Now, we understand the need to have a "preliminaries" section in cryptology papers. It is an entry and time-saving test!

3 Schemes

3.1 Scheme

Let $(\mathbb{G}, g, p)$ be the cyclic group and $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses a random number $\alpha \in \mathbb{Z}_p$, computes $g_1 = g^\alpha$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = g_1, sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk. It computes the signature σ_m on m as

$$\sigma_m = \alpha + H(m) \mod p.$$

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$g^{\sigma_m} = g_1 \cdot g^{H(m)}.$$

Question 1 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 2 *Is this signature scheme secure in the EUF-CMA security model?*

3.2 Scheme

Let $(\mathbb{G}, g, p)$ be the cyclic group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses a random number $\alpha \in \mathbb{Z}_p$, computes $g_1 = g^\alpha$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = g_1, sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose a random $r \in \mathbb{Z}_p$ and compute $\sigma_1 = g^r$.
- Compute $\sigma_2 = r + \alpha H(m) \mod p$.
- Return the signature $\sigma_m = (\sigma_1, \sigma_2)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$g^{\sigma_2} = \sigma_1 \cdot g_1^{H(m)}.$$

Question 3 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 4 *Is this signature scheme secure in the EUF-CMA security model?*

3.3 Scheme

Let $(\mathbb{G}, g, p)$ be the cyclic group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses random numbers $\alpha, \beta \in \mathbb{Z}_p$, computes $g_1 = g^\alpha, g_2 = g^\beta$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, g_2), sk = (\alpha, \beta).$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk. It computes the signature σ_m on m as

$$\sigma_m = \alpha + H(m) \cdot \beta \mod p.$$

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$g^{\sigma_m} = g_1 \cdot g_2^{H(m)}.$$

Question 5 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 6 *Is this signature scheme secure in the EUF-CMA security model?*

3.4 Scheme

Let $(\mathbb{G}, g, p)$ be the cyclic group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses random numbers $\alpha, \beta \in \mathbb{Z}_p$, computes $g_1 = g^\alpha, g_2 = g^\beta$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, g_2), sk = (\alpha, \beta).$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose a random $r \in \mathbb{Z}_p$ and compute $\sigma_1 = g^r$.
- Compute $\sigma_2 = \frac{\alpha+r}{\beta+H(m)} \mod p$.
- Return the signature $\sigma_m = (\sigma_1, \sigma_2)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$\left(g_2 g^{H(m)}\right)^{\sigma_2} = g_1 \cdot \sigma_1.$$

Question 7 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 8 *Is this signature scheme secure in the EUF-CMA security model?*

3.5 Scheme

Let $(\mathbb{G}, g, p)$ be the cyclic group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses a random number $\alpha \in \mathbb{Z}_p$, computes $g_1 = g^\alpha$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = g_1, sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose random $r_1, r_2, t \in \mathbb{Z}_p$ and compute $\sigma_1 = g^{r_1}, \sigma_2 = g^{r_2}, \sigma_3 = g^t$.
- Compute $\sigma_4 = \frac{r_1 H(m,pk,1) + r_2 H(m,pk,2)}{\alpha + tH(\sigma_3)} \mod p$.
- Return the signature $\sigma_m = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$\sigma_1^{H(m,pk,1)} \cdot \sigma_2^{H(m,pk,2)} = (g_1 \sigma_3^{H(\sigma_3)})^{\sigma_4}.$$

Question 9 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 10 *Is this signature scheme secure in the EUF-CMA security model?*

3.6 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses random numbers $\alpha, \beta \in \mathbb{Z}_p$, computes $g_1 = g^\alpha, g_2 = g^\beta$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, g_2), sk = (\alpha, \beta).$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose a random $r \in \mathbb{Z}_p$ and compute $\sigma_2 = g^r$.
- Compute $\sigma_1 = g^{\alpha\beta + H(m)\cdot r}$.
- Return the signature $\sigma_m = (\sigma_1, \sigma_2)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_1, g) = e(g_1, g_2)e(g^{H(m)}, \sigma_2).$$

Question 11 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 12 *Is this signature scheme secure in the EUF-CMA security model?*

3.7 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses random numbers $\alpha, \beta \in \mathbb{Z}_p$, computes $g_1 = g^\alpha, g_2 = g^\beta$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, g_2), sk = (\alpha, \beta).$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose a random $r \in \mathbb{Z}_p$ and compute $\sigma_2 = g^r$.
- Compute $\sigma_1 = g^{\alpha\beta + H(m) \cdot r \cdot \beta}$.
- Return the signature $\sigma_m = (\sigma_1, \sigma_2)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_1, g) = e(g_1, g_2) e(g_2^{H(m)}, \sigma_2).$$

Question 13 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 14 *Is this signature scheme secure in the EUF-CMA security model?*

3.8 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses random numbers $\alpha, \beta \in \mathbb{Z}_p$, computes $g_1 = g^\alpha, g_2 = g^\beta$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, g_2), sk = (\alpha, \beta).$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose a random $r \in \mathbb{Z}_p$ and compute $\sigma_2 = g^r$.
- Compute $\sigma_1 = g^{\frac{\alpha \cdot r}{\beta + H(m)}}$.
- Return the signature $\sigma_m = (\sigma_1, \sigma_2)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_1, g_2 g^{H(m)}) = e(g_1, \sigma_2).$$

Question 15 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 16 *Is this signature scheme secure in the EUF-CMA security model?*

3.9 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses random numbers $\alpha, \beta \in \mathbb{Z}_p$, computes $g_1 = g^\alpha, g_2 = g^\beta$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, g_2), sk = (\alpha, \beta).$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose a random $r \in \mathbb{Z}_p$ and set $\sigma_2 = r$.
- Compute $\sigma_1 = g^{\frac{\alpha}{\beta + H(m,1) + r \cdot H(m,2)}}$.
- Return the signature $\sigma_m = (\sigma_1, \sigma_2)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_1, g_2 g^{H(m,1)} \cdot g^{r \cdot H(m,2)}) = e(g_1, g).$$

Question 17 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 18 *Is this signature scheme secure in the EUF-CMA security model?*

3.10 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \{0,1\}^n$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses a random number $\alpha \in \mathbb{Z}_p$, choose random group elements $(u_0, u_1, u_2, \cdots, u_n)$, computes $g_1 = g^\alpha$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, u_0, u_1, \cdots, u_n), sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk. It returns the signature σ_m on m as

$$\sigma_m = (u_0 \prod_{i=1}^{n} u_i^{m[i]})^\alpha,$$

where $m[i]$ is the i-th bit of $H(m)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_m, g) = e(u_0 \prod_{i=1}^{n} u_i^{m[i]}, g_1).$$

Question 19 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 20 *Is this signature scheme secure in the EUF-CMA security model?*

3.11 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses random numbers $\alpha, \beta \in \mathbb{Z}_p$, computes $g_1 = g^\alpha, g_2 = g^\beta$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, g_2), sk = (\alpha, \beta).$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk. It returns the signature σ_m on m as

$$\sigma_m = (g_1 g^{H(m)})^\beta.$$

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_m, g) = e(g_1 g^{H(m)}, g_2).$$

Question 21 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 22 *Is this signature scheme secure in the EUF-CMA security model?*

3.12 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses a random number $\alpha \in \mathbb{Z}_p$, chooses random group elements (u_1, u_2, v_1, v_2), computes $g_1 = g^\alpha$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, u_1, u_2, v_1, v_2), sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose random $r_1, r_2, s \in \mathbb{Z}_p$ and sets $\sigma_1 = r_1, \sigma_2 = r_2$.
- Compute $\sigma_3 = g^s$ and σ_4 as

$$\sigma_4 = (u_1^{r_1} u_2^{H(m)})^\alpha (v_1 v_2^{r_2})^s$$

- Return the signature $\sigma_m = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_4, g) = e(u_1^{\sigma_1} u_2^{H(m)}, g_1) e(v_1 v_2^{\sigma_2}, \sigma_3).$$

Question 23 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 24 *Is this signature scheme secure in the EUF-CMA security model?*

3.13 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \mathbb{G}$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses random numbers $\alpha, \beta \in \mathbb{Z}_p$, computes $g_1 = g^\alpha, g_2 = g^\beta$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, g_2), sk = (\alpha, \beta).$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose random $r, s \in \mathbb{Z}_p$ and compute $\sigma_3 = g^s$.
- Compute $\sigma_2 = g^r H(m)^s$
- Compute $\sigma_1 = g^{\frac{\alpha+r}{\beta}}$
- Return the signature $\sigma_m = (\sigma_1, \sigma_2, \sigma_3)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$\frac{e(\sigma_1, g_2)}{e(g_1, g)} = \frac{e(\sigma_2, g)}{e(H(m), \sigma_3)}.$$

Question 25 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 26 *Is this signature scheme secure in the EUF-CMA security model?*

3.14 Scheme

Let $(\mathbb{G}, g, p)$ be the cyclic group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses a random number $\alpha \in \mathbb{Z}_p$, computes $g_1 = g^\alpha$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = g_1, sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose random $r, s \in \mathbb{Z}_p$ and compute $\sigma_1 = g^r, \sigma_2 = g^s$.
- Compute $\sigma_3 = r + \alpha H(g^r, m, 1) + sH(g^r, m, 2) \mod p$.
- Return the signature $\sigma_m = (\sigma_1, \sigma_2, \sigma_3)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$g^{\sigma_3} = \sigma_1 \cdot g_1^{H(\sigma_1, m, 1)} \cdot \sigma_2^{H(\sigma_1, m, 2)}.$$

Question 27 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 28 *Is this signature scheme secure in the EUF-CMA security model?*

3.15 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group that will be shared by all users.

KeyGen: The key generation algorithm chooses random numbers $\alpha, \beta \in \mathbb{Z}_p$, chooses a random group element h, computes $U = e(g,g)^{\alpha}, V = e(g,g)^{\beta}$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (U, V, h),\ sk = (\alpha, \beta).$$

Sign: The signing algorithm takes as input a message $m \in \mathbb{Z}_p$ and the secret key sk. It chooses a random number r and returns the signature σ_m on m as

$$\sigma_m = (\sigma_1, \sigma_2) = (g^{\alpha m + \beta} \cdot h^r, g^r)$$

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_1, g) = U^m \cdot V \cdot e(h, \sigma_2).$$

Question 29 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 30 *Is this signature scheme secure in the EUF-CMA security model?*

3.16 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses a random number $\alpha \in \mathbb{Z}_p$, computes $g_1 = g^{\frac{1}{\alpha}}$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = g_1, sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose a random $r \in \mathbb{Z}_p$ and compute $\sigma_2 = g^r$.
- Compute $\sigma_1 = g^{\alpha(H(m)+r)}$
- Return the signature $\sigma_m = (\sigma_1, \sigma_2)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_1, g_1) = e(g^{H(m)}\sigma_2, g).$$

Question 31 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 32 *Is this signature scheme secure in the EUF-CMA security model?*

3.17 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \mathbb{Z}_p$ that will be shared by all users.

KeyGen: The key generation algorithm chooses a random number $\alpha \in \mathbb{Z}_p$, choose random group elements $(u_0, u_1, u_2, \cdots, u_n)$, computes $g_1 = g^\alpha$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, u_0, u_1, \cdots, u_n), sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk. It chooses a random number $r \in \mathbb{Z}_p$ and returns the signature σ_m on m as

$$\sigma_m = (\sigma_1, \sigma_2) = \Big((u_0^{H(m,r)} \prod_{i=1}^{n} u_i^{M[i]})^\alpha, r\Big),$$

where $M[i]$ is the i-th bit of $H(m, r)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_1, g) = e\Big(u_0^{H(m,\sigma_2)} \prod_{i=1}^{n} u_i^{M[i]}, g_1\Big).$$

Question 33 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 34 *Is this signature scheme secure in the EUF-CMA security model?*

3.18 Scheme

Let $(\mathbb{G}, g, p)$ be the cyclic group that will be shared by all users.

KeyGen: The key generation algorithm chooses a random number $\alpha \in \mathbb{Z}_p$, computes $g_1 = g^\alpha$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = g_1, sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \mathbb{G}$ and the secret key sk.

- Choose a random $r \in \mathbb{Z}_p$ and compute $\sigma_1 = m \cdot g^r$.
- Compute $\sigma_2 = r + \alpha H(\sigma_1) \mod p$.
- Return the signature $\sigma_m = (\sigma_1, \sigma_2)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$m \cdot \frac{g^{\sigma_2}}{g_1^{H(\sigma_1)}} = \sigma_1.$$

Question 35 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 36 *Is this signature scheme secure in the EUF-CMA security model?*

3.19 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \mathbb{Z}_p$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses random numbers $\alpha, \beta \in \mathbb{Z}_p$, computes $g_1 = g^\alpha, g_2 = g^\beta$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = (g_1, g_2), sk = (\alpha, \beta).$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose a random $R \in \mathbb{G}$ and set $\sigma_1 = R$.
- Compute $\sigma_2 = R^{\alpha + H(m)\beta}$
- Return the signature $\sigma_m = (\sigma_1, \sigma_2)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_2, g) = e(\sigma_1, g_1 g_2^{H(m)}).$$

Question 37 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 38 *Is this signature scheme secure in the EUF-CMA security model?*

3.20 Scheme

Let $(\mathbb{G}, \mathbb{G}_T, g, e, p)$ be the pairing group and $H : \{0,1\}^* \to \mathbb{G}$ be the cryptographic hash function that will be shared by all users.

KeyGen: The key generation algorithm chooses a random number $\alpha \in \mathbb{Z}_p$, computes $g_1 = g^\alpha$, and returns a public/secret key pair (pk, sk) as follows:

$$pk = g_1, sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$ and the secret key sk.

- Choose a random $r \in \mathbb{Z}_p$ and compute $\sigma_2 = g^r$.
- Compute $\sigma_1 = H(m, g_1, \sigma_2)^{\alpha+r}$
- Return the signature $\sigma_m = (\sigma_1, \sigma_2)$.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m) and the public key pk. It accepts the signature if

$$e(\sigma_1, g) = e(g_1 \cdot \sigma_2, H(m, g_1, \sigma_2)).$$

Question 39 *Is this signature scheme secure against forgeability in the key-only attack?*

Question 40 *Is this signature scheme secure in the EUF-CMA security model?*

==========We are tooooooo lazy to provide answers.==========