28th Australasian Conference on Information Security and Privacy (ACISP 2023)

iC[2] Institute of Cybersecurity and Cryptology

UNIVERSITY OF WOLLONGONG AUSTRALIA

A Tightly Secure ID-Based Signature Scheme under DL Assumption in AGM

Jia-Chng Loh, Fuchun Guo, Willy Susilo

IC^2,SCIT, University of Wollongong, Australia

Guomin Yang

Singapore Management University, Singapore

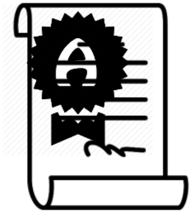
Outline

- Introduction
- Challenge and Our Contribution
- The proposed BLS-IBS Scheme
- Security Proof in AGM (High-level)
- Conclusion

Introduction

Identity-based Signatures

Digital Signatures



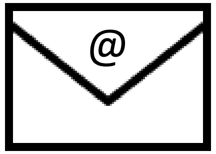
- Integrity, authentication, and non-repudiation
- Can be provably secure (forging is **computationally hard**)

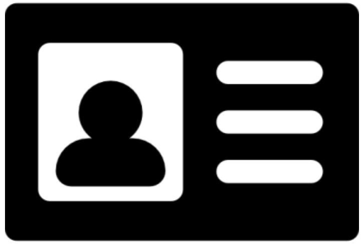
However, there is a need of public key infrastructure

- To certify users' public keys

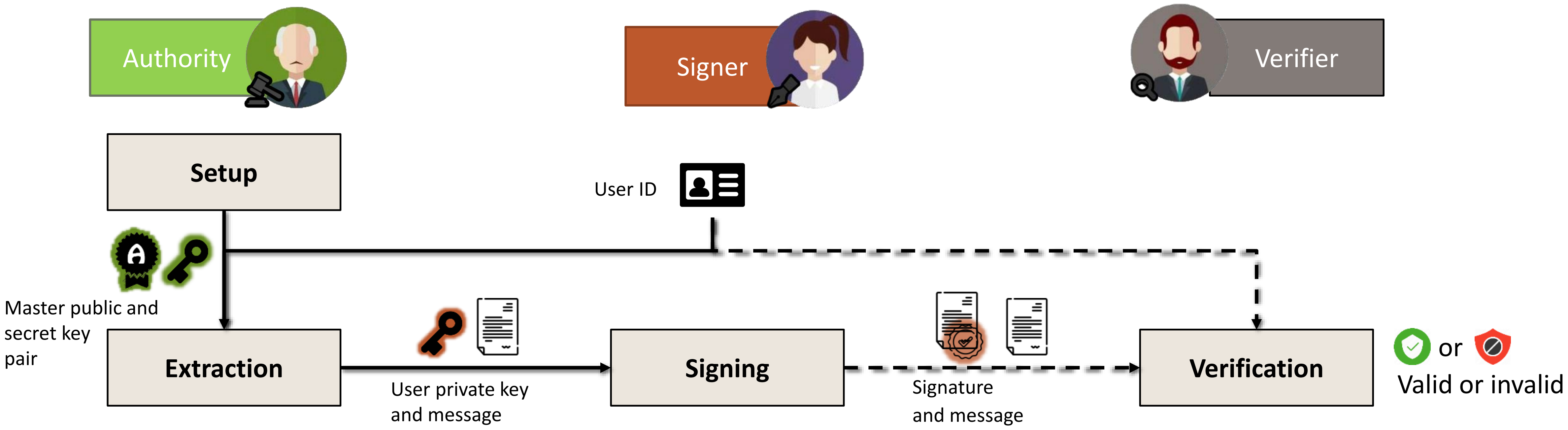
Identity (ID)-based Signatures (IBS)

- Users' identity *ID* serves as the public key
- E.g. email address and ID number





Definition of IBS

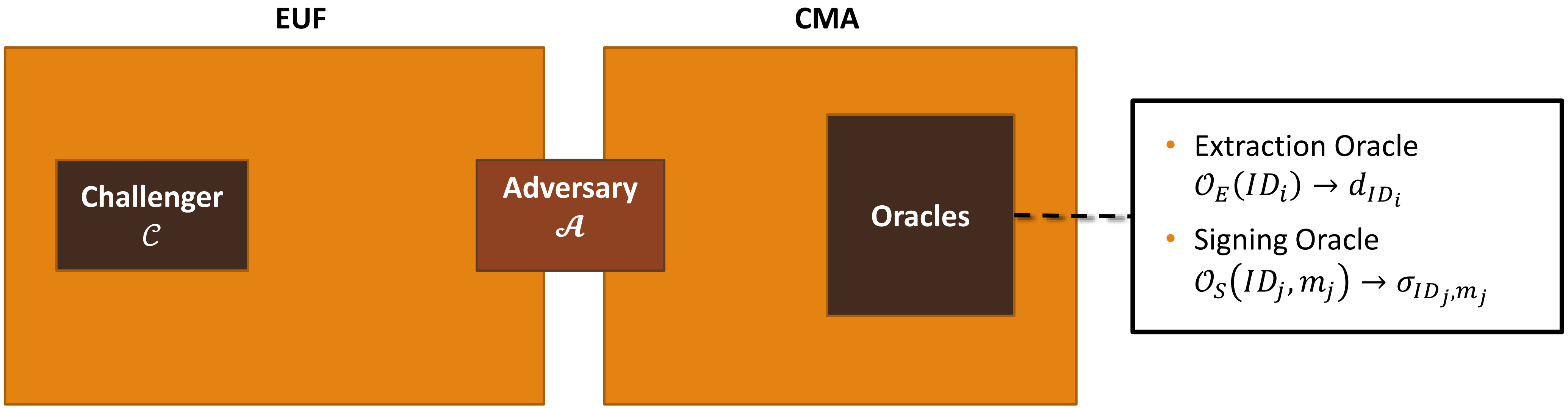


An IBS scheme is defined with four main algorithms:

- **Setup:** On input security parameters, it generates master public and secret key pair
- **Extraction**: On input master secret key and user ID, it generates a user private key
- **Signing**: On input user private key and message, it generates a signature
- **Verification**: On input ID, signature, and message, it returns its validity

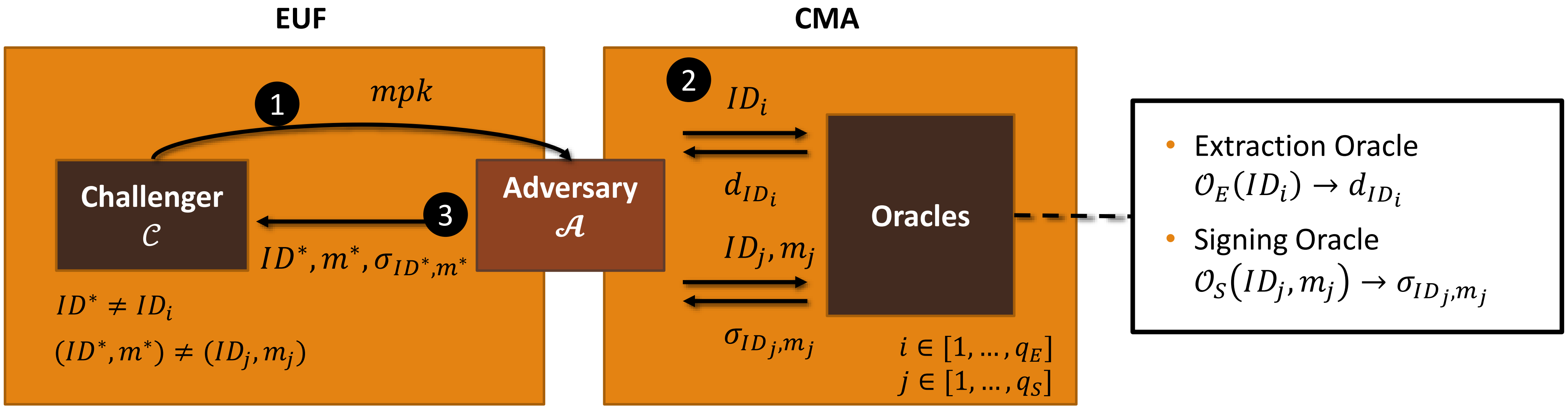
Security Model

Existential unforgeability against chosen identity-and-message attacks (EUF-CMA)



Security Model

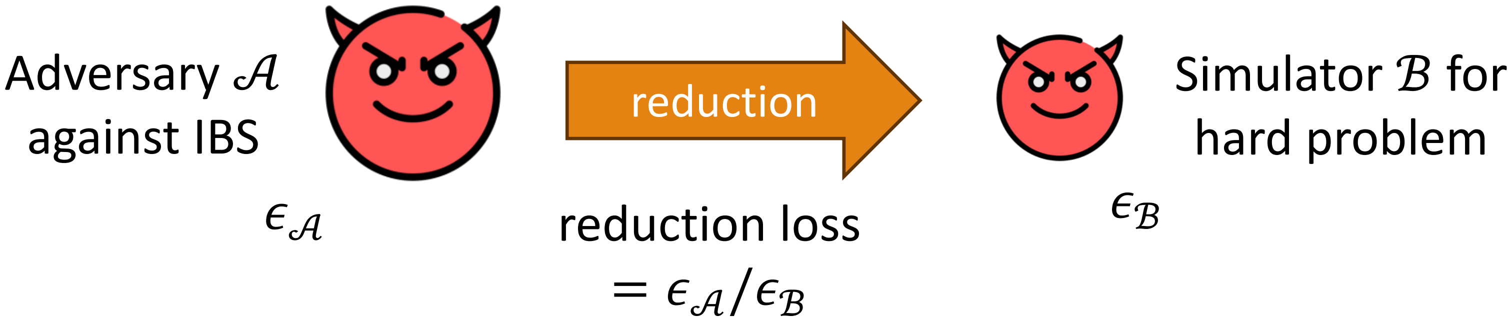
Existential unforgeability against chosen identity-and-message attacks (EUF-CMA)



1. Setup: $\mathcal{C}$ prepares parameters and returns mpk to $\mathcal{A}$
2. Query: $\mathcal{A}$ requests user private key $d_{ID_i} \leftarrow \mathcal{O}_E(ID_i)$ and signatures $\sigma_{ID_j,m_j} \leftarrow \mathcal{O}_S(ID_j, m_j)$
3. Forgery: $\mathcal{A}$ returns valid forgery $(ID^*, m^*, \sigma_{ID^*,m^*})$ and wins if ID^* and (ID^*, m^*) have **not** been queried to $\mathcal{O}_E(\cdot), \mathcal{O}_S(\cdot,\cdot)$ respectively

Security Reduction

To prove the security of the scheme



Security Reduction

To prove the security of the scheme

Adversary $\mathcal{A}$ against IBS

$\epsilon_{\mathcal{A}}$

reduction

reduction loss $= \epsilon_{\mathcal{A}} / \epsilon_{\mathcal{B}}$

Simulator $\mathcal{B}$ for hard problem

$\epsilon_{\mathcal{B}}$

"Ideal security" in cyclic group setting

- Hardest problem: Discrete logarithm (DL)
- Standard EUF-CMA security model

Tight reduction: Loss factor is $O(1)$

- Better theoretical result
- More efficient implementation

It is hard to achieve ideal security!

We may resort to the "idealized models"

Algebraic Group Model (AGM)

Idealizing the adversary's computations as algebraic [FKL18] at Crypto'18

Suppose $g \in \mathbb{G}$ is a cyclic group generator and $x_1, x_2 \in \mathbb{Z}_p^*$ are any prime number. Let $X_1 = g^{x_1}, X_2 = g^{x_2}$

Simulator		Algebraic Adversary
	❶ X_1, X_2 $\longrightarrow$	

Algebraic Group Model (AGM)

Idealizing the adversary's computations as algebraic [FKL18] at Crypto'18

Suppose $g \in \mathbb{G}$ is a cyclic group generator and $x_1, x_2 \in \mathbb{Z}_p^*$ are any prime number. Let $X_1 = g^{x_1}, X_2 = g^{x_2}$

Simulator

Algebraic Adversary

1 X_1, X_2

2 $\vec{\pi} = [\pi_1, \pi_2] \in \mathbb{Z}_p$

$Z = X_1^{\pi_1} \cdot X_2^{\pi_2}$

3 $Z, \vec{\pi}$

4 $Z = g^z$, such that

$z = x_1\pi_1 + x_2\pi_2$

Signature schemes with ideal security in AGM: BLS and Schnorr signatures

Algebraic Group Model (AGM)

Idealizing the adversary's computations as algebraic [FKL18] at Crypto'18

Suppose $g \in \mathbb{G}$ is a cyclic group generator and $x_1, x_2 \in \mathbb{Z}_p^*$ are any prime number. Let $X_1 = g^{x_1}, X_2 = g^{x_2}$

Simulator

Algebraic Adversary

1 X_1, X_2

2 $\vec{\pi} = [\pi_1, \pi_2] \in \mathbb{Z}_p$

$Z = X_1^{\pi_1} \cdot X_2^{\pi_2}$

3 $Z, \vec{\pi}$

4 $Z = g^z$, such that

$z = x_1\pi_1 + x_2\pi_2$

Signature schemes with ideal security in AGM: BLS and Schnorr signatures

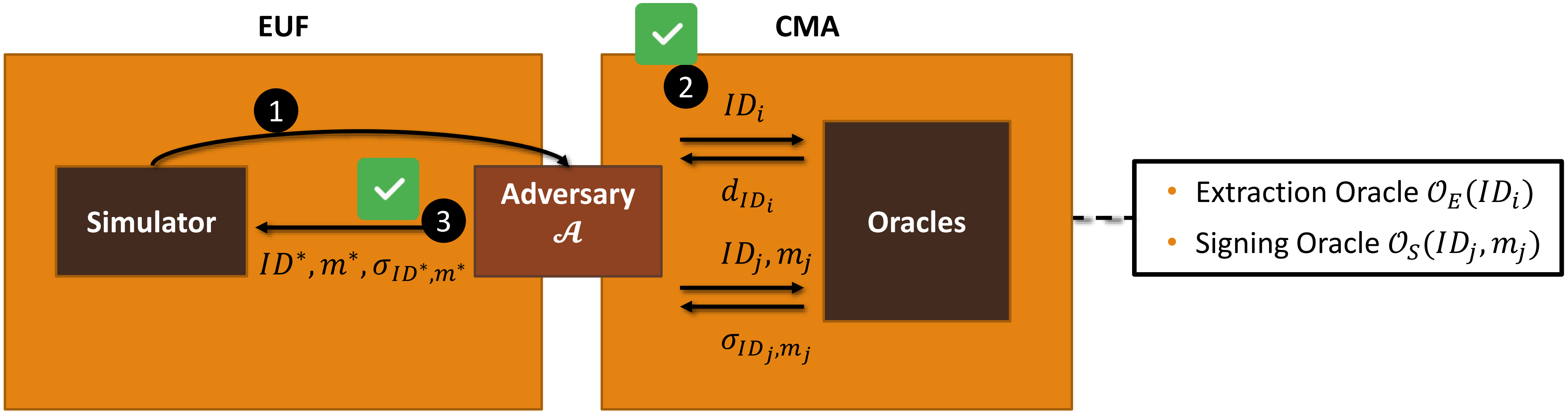
However, IBS scheme with ideal security in AGM has not been discovered

Challenge and Our Contribution

Difficulty of Tight Reduction in IBS

To achieve tightly EUF-CMA secure IBS scheme, a reduction must capture the following two points

- Respond any user private key query
- Reduce problem solutions based on any forgery

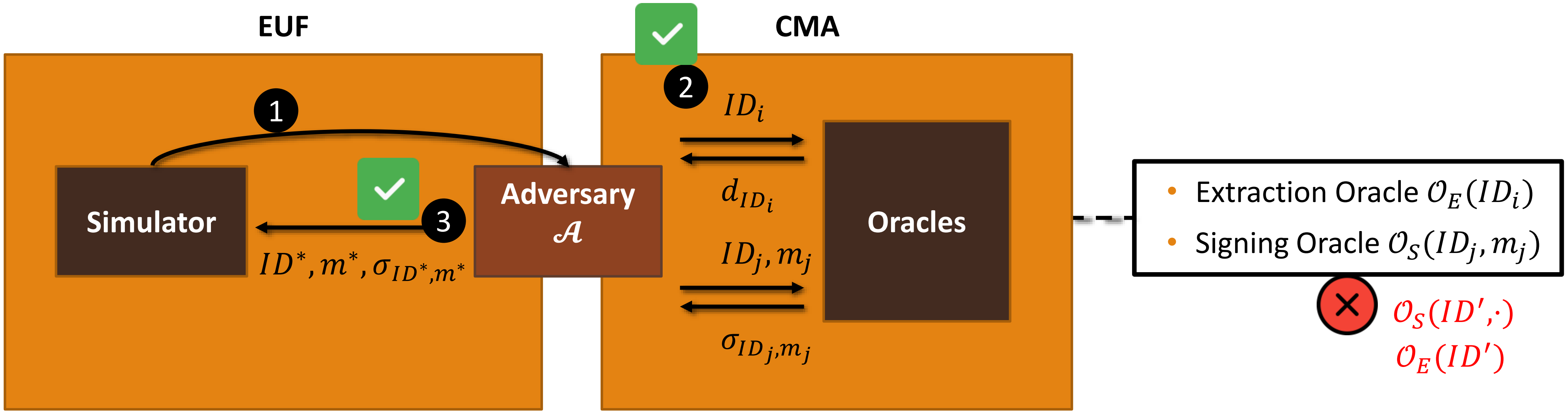


Difficulty of Tight Reduction in IBS

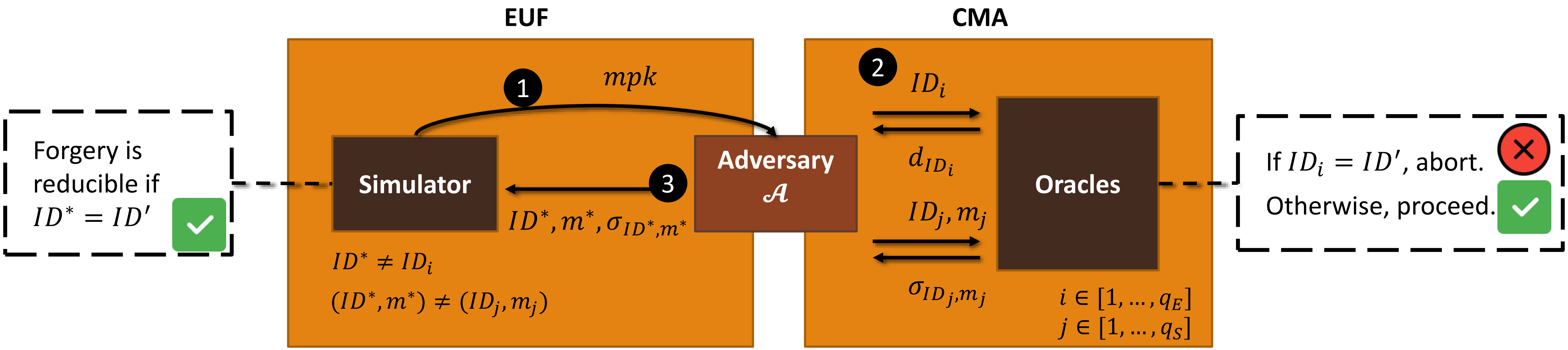
To achieve tightly EUF-CMA secure IBS scheme, a reduction must capture the following two points

- Respond any user private key query
- Reduce problem solutions based on any forgery

Conflicting: In most reductions, if user private key is simulated, the corresponding forgery may not be reducible



Existing Techniques



Existing techniques

- Choose a target $ID' \in [ID_1, \ldots, ID_q]$ with a non-simulatable key
- If forgery matches target one, i.e. $ID^* = ID'$, it can be reducible

However, this results loose reduction due to random target ID'

Our Contribution

We **present a new IBS scheme**, namely BLS-IBS, which is extended based on BLS signatures [BLS04]

The security of BLS-IBS **achieve ideal security in AGM:**

Approach 1: The reduction can simulate any user private key

Approach 2: The reduction can reduce any forgery

- In AGM, the adversary's forgery and representations can be classified into several cases
- The reduction contains two simulations
- The reduction solves for DL solution in either simulations

Our Contribution

We **present a new IBS scheme**, namely BLS-IBS, which is extended based on BLS signatures [BLS04]

The security of BLS-IBS **achieve ideal security in AGM:**

Approach 1: The reduction can simulate any user private key

Approach 2: The reduction can reduce any forgery

DL problem instance is always embedded

- In AGM, the adversary's forgery and representations can be classified into several cases
- The reduction contains two simulations
- The reduction solves for DL solution in either simulations

The BLS-IBS Scheme

The Scheme

Extended based on BLS signatures: public key $pk = g^x$ and signature $\sigma_m = H(m)^x$

Security parameters $\xrightarrow{1^k}$ **Setup** $\rightarrow$ Master public and secret key pair

$$mpk = (p, g, e, \mathbb{G}, \mathbb{G}_T, H_1(\cdot), H_2(\cdot), X = g^x)$$

$$msk = x$$

(msk, ID) $\rightarrow$ **Extraction** $\rightarrow$ User private key

$$d_{ID} = \left(H_1(ID)\right)^x$$

(mpk, d_{ID}, m) $\rightarrow$ **Signing** $\rightarrow$ Signature

$$\sigma_{ID,m} \begin{cases} \sigma_{ID,m}^{(1)} = d_{ID} \cdot \left(H_2(m)\right)^r \\ \sigma_{ID,m}^{(2)} = g^r \end{cases}$$

$(mpk, ID, m, \sigma_{ID,m})$ $\rightarrow$ **Verification** $\rightarrow$ 1 or 0

$$e\left(\sigma_{ID,m}^{(1)}, g\right) = e(H_1(ID), X) \cdot e\left(H_2(m), \sigma_{ID,m}^{(2)}\right)$$

The Security Proof

AGM (High-level)

Approach 1: How to simulate

User private key: $d_{ID} = \left(H_1(ID)\right)^x$ **Signature $\boldsymbol{\sigma_{ID,m}}$:** $\sigma_{ID,m}^{(1)} = d_{ID} \cdot \left(H_2(m)\right)^r$, $\sigma_{ID,m}^{(2)} = g^r$

Given a DL problem instance tuple (g, g^a), we design two simulations as follows:

$\mathcal{R}_1$: Embeds g^a into master public key X and signature randomness $\sigma_{ID,m}^{(2)}$

$\mathcal{R}_2$: Embeds g^a into hash values $H_1(\cdot) \to H_{ID},\ H_2(\cdot) \to H_m$

Approach 1: How to simulate

User private key: $d_{ID} = \left(H_1(ID)\right)^x$ **Signature $\boldsymbol{\sigma_{ID,m}}$:** $\sigma_{ID,m}^{(1)} = d_{ID} \cdot \left(H_2(m)\right)^r$, $\sigma_{ID,m}^{(2)} = g^r$

Given a DL problem instance tuple (g, g^a), we design two simulations as follows:

- $\mathcal{R}_1$: Embeds g^a into master public key X and signature randomness $\sigma_{ID,m}^{(2)}$
- $\mathcal{R}_2$: Embeds g^a into hash values $H_1(\cdot) \rightarrow H_{ID}$, $H_2(\cdot) \rightarrow H_m$

Approach 1: How to simulate

User private key: $d_{ID} = \left(H_1(ID)\right)^x$ **Signature $\boldsymbol{\sigma_{ID,m}}$:** $\sigma^{(1)}_{ID,m} = d_{ID} \cdot \left(H_2(m)\right)^r$, $\sigma^{(2)}_{ID,m} = g^r$

Given a DL problem instance tuple (g, g^a), we design two simulations as follows:

$\mathcal{R}_1$: Embeds g^a into master public key X and signature randomness $\sigma^{(2)}_{ID,m}$

$\mathcal{R}_2$: Embeds g^a into hash values $H_1(\cdot) \rightarrow H_{ID}$, $H_2(\cdot) \rightarrow H_m$

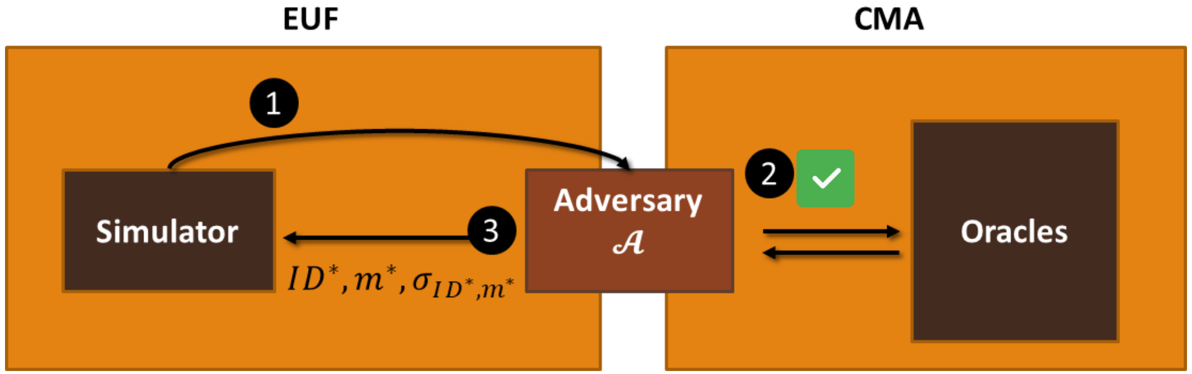
Elements	Simulation $\mathcal{R}_1$	Simulation $\mathcal{R}_2$
X	g^a	g^x
H_{ID_i}	$g^{h_{ID_i}}$	$g^{au_{1,i}+v_{1,i}}$
H_{m_i}	$g^{h_{m_i}}$	$g^{au_{2,i}+v_{2,i}}$
d_{ID_i}	$g^{ah_{ID_i}}$	$g^{au_{1,i}x+v_{1,i}x}$
$\sigma^{(1)}_{ID_i,m_i}$	$g^{a\left(h_{ID_i}+h_{m_i}s_i\right)+h_{m_i}t_i}$	$g^{a\left(u_{1,i}x+u_{2,i}r_i\right)+v_{1,i}x+v_{2,i}r_i}$
$\sigma^{(2)}_{ID_i,m_i}$	$g^{as_i+t_i}$	g^{r_i}

During the query phase,

Every d_{ID} is simulatable.

Both simulations will not abort!

Approach 2: How to reduce

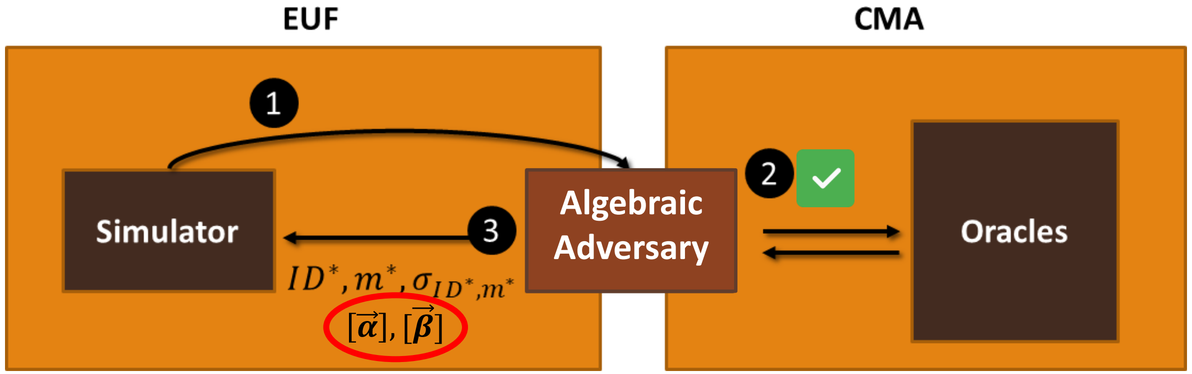


The simulator obtains forgery σ_{ID^*,m^*}

$$\sigma^{(1)}_{ID^*,m^*} = d_{ID^*} \cdot \left(H_2(m^*)\right)^{r^*} = g^{x h_{ID^*} + r^* h_{m^*}}$$

$$\sigma^{(2)}_{ID^*,m^*} = g^{r^*}$$

Approach 2: How to reduce



The simulator obtains forgery σ_{ID^*,m^*}

$$\sigma^{(1)}_{ID^*,m^*} = d_{ID^*} \cdot \left(H_2(m^*)\right)^{r^*} = g^{x h_{ID^*} + r^* h_{m^*}}$$

$$\sigma^{(2)}_{ID^*,m^*} = g^{r^*}$$

And representations (in AGM), such that

$$\sigma^{(1)}_{ID^*,m^*} = g^{x\theta + \hat{\theta} + \sum_i^{q_s} r_i \omega_{\alpha_i}}$$

$$\sigma^{(2)}_{ID^*,m^*} = g^{x\delta + \hat{\delta} + \sum r_i \omega_{\beta_i}}$$

1

2

Therefore, the simulator can derive a general modular equation

1 $$x h_{ID^*} + r^* h_{m^*} = x\theta + \hat{\theta} + \sum_i^{q_s} r_i \omega_{\alpha_i}$$

2 $$r^* = x\delta + \hat{\delta} + \sum_i^{q_s} r_i \omega_{\beta_i}$$

General equation:

$$\boxed{x(h_{ID^*} + h_{m^*}\delta - \theta) + \Sigma_i^{q_s} r_i \left(h_{m^*}\omega_{\beta_i} - \omega_{\alpha_i}\right) = \hat{\theta} - h_{m^*}\hat{\delta}}$$

Next, we classify this into several cases (in full proof).
For high-level, we compress them into two.

Classification: Compact and high-level

In this presentation, we compress into two (at high-level)

The simulator obtains:

$$\boxed{x(h_{ID^*} + h_{m^*}\delta - \theta) + \Sigma_i^{q_s} r_i\left(h_{m^*}\omega_{\beta_i} - \omega_{\alpha_i}\right) = \hat{\theta} - h_{m^*}\hat{\delta}}$$

Non-zero coefficients

The simulator solves for either x or r_i.

Simulation $\mathcal{R}_1$

Zero coefficients

The simulator solves for either h_{ID^*}, h_{m^*}.

Simulation $\mathcal{R}_2$

Classification: Compact and high-level

In this presentation, we compress into two (at high-level)

The simulator obtains:

$$x(h_{ID^*} + h_{m^*}\delta - \theta) + \Sigma_i^{q_s} r_i\left(h_{m^*}\omega_{\beta_i} - \omega_{\alpha_i}\right) = \hat{\theta} - h_{m^*}\hat{\delta}$$

Non-zero coefficients → The simulator solves for either x or r_i. **Simulation $\mathcal{R}_1$**

Zero coefficients → The simulator solves for either h_{ID^*}, h_{m^*}. **Simulation $\mathcal{R}_2$**

Recall that by our simulations:

$\mathcal{R}_1$: Embeds g^a into master public key $X = g^x$ and signature randomness $\sigma_{ID,m_i}^{(2)} = g^{r_i}$

$\mathcal{R}_2$: Embeds g^a into hash values $H_{ID} = g^{h_{ID}},\ H_m = g^{h_m}$

Classification: Compact and high-level

In this presentation, we compress into two (at high-level)

The simulator obtains:

$$x(h_{ID^*} + h_{m^*}\delta - \theta) + \Sigma_i^{q_s} r_i\left(h_{m^*}\omega_{\beta_i} - \omega_{\alpha_i}\right) = \hat{\theta} - h_{m^*}\hat{\delta}$$

Non-zero coefficients

The simulator solves for either x or r_i.

Simulation $\mathcal{R}_1$

Zero coefficients

The simulator solves for either h_{ID^*}, h_{m^*}.

Simulation $\mathcal{R}_2$

Recall that by our simulations:

$\mathcal{R}_1$: Embeds g^a into master public key $X = g^x$ and signature randomness $\sigma_{ID,m_i}^{(2)} = g^{r_i}$

$\mathcal{R}_2$: Embeds g^a into hash values $H_{ID} = g^{h_{ID}}$, $H_m = g^{h_m}$

Classification: Compact and high-level

In this presentation, we compress into two (at high-level)

The simulator obtains:

$$x(h_{ID^*} + h_{m^*}\delta - \theta) + \Sigma_i^{q_s} r_i\left(h_{m^*}\omega_{\beta_i} - \omega_{\alpha_i}\right) = \hat{\theta} - h_{m^*}\hat{\delta}$$

Non-zero coefficients

The simulator solves for either x or r_i.

Simulation $\mathcal{R}_1$

Zero coefficients

The simulator solves for either h_{ID^*}, h_{m^*}.

Simulation $\mathcal{R}_2$

Recall that by our simulations:

$\mathcal{R}_1$: Embeds g^a into master public key $X = g^x$ and signature randomness $\sigma_{ID,m_i}^{(2)} = g^{r_i}$

$\mathcal{R}_2$: Embeds g^a into hash values $H_{ID} = g^{h_{ID}}$, $H_m = g^{h_m}$

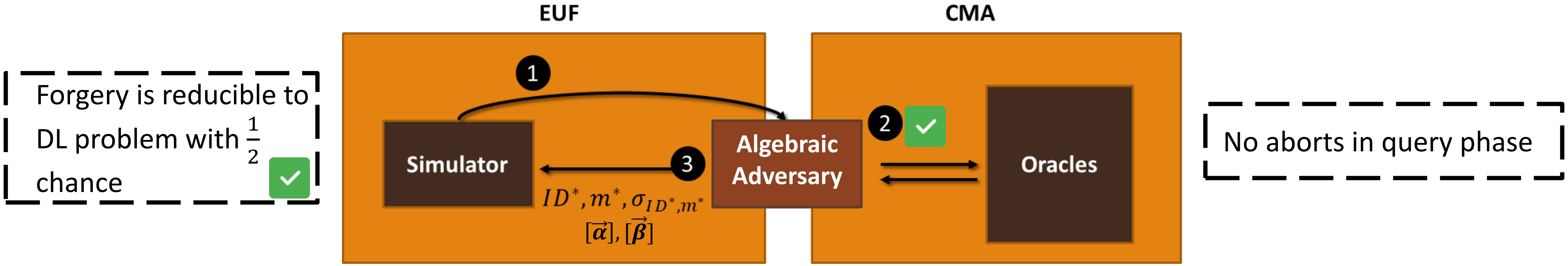
Therefore, the simulator solves for DL solution a in either simulation depending on adversary's behavior.

$$\Pr[Success] = \frac{1}{2} \cdot \Pr[\mathcal{R}_1] + \frac{1}{2} \cdot \Pr[\mathcal{R}_2] = \frac{1}{2}$$

Conclusion

Conclusion

The proposed BLS-IBS achieves ideal security in AGM



A valuable insight on how to achieve ideal security for IBS in AGM

Future work: Pairing-free IBS with ideal security

- Schnorr-like IBS [GG09]: A similar approach cannot work here as the simulator cannot simulate any user private key

Thank you