CSI-Otter: Isogeny-Based (Partially) Blind Signatures from the Class Group Action with a Twist

Shuichi Katsumata — PQSHIELD, AIST

Yi-Fu Lai — THE UNIVERSITY OF AUCKLAND NEW ZEALAND

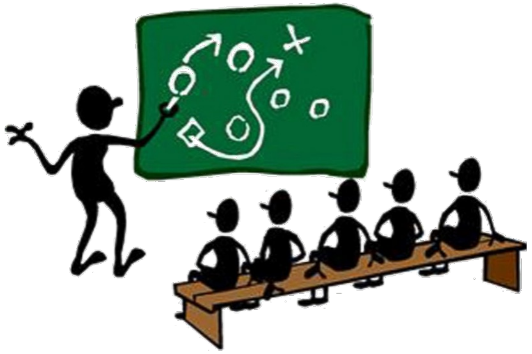
Jason T. LeGrow — VIRGINIA TECH, Commonwealth Cyber Initiative

Ling Qin — THE UNIVERSITY OF AUCKLAND NEW ZEALAND

Our Result in Short

A new Schnorr-type 3-round **blind signature** based on **isogenies** (CSIDH).

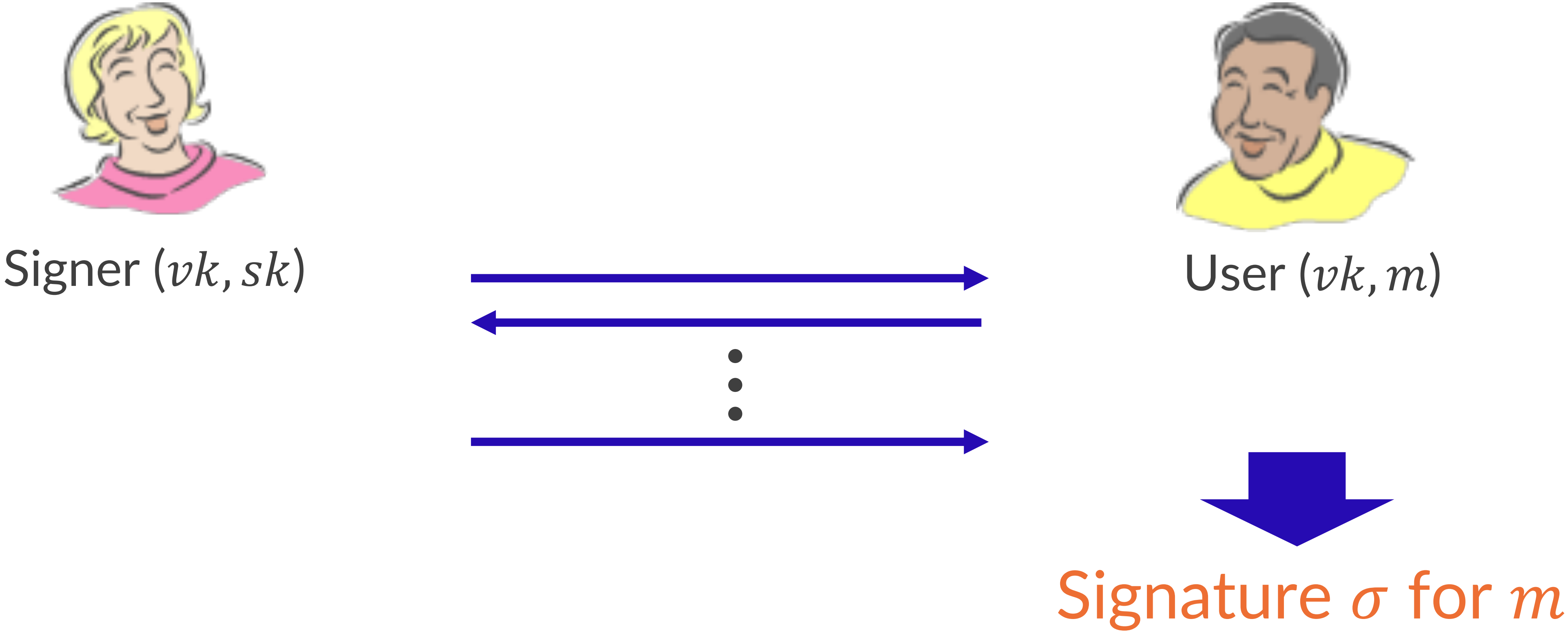
- The **first** (partially) blind signature from isogenies.
- Provable security for log-concurrent sessions.
- New hardness assumption for optimization.



1. Background

What are Blind Signatures?

⇒ An interactive signing protocol with **“privacy”**.



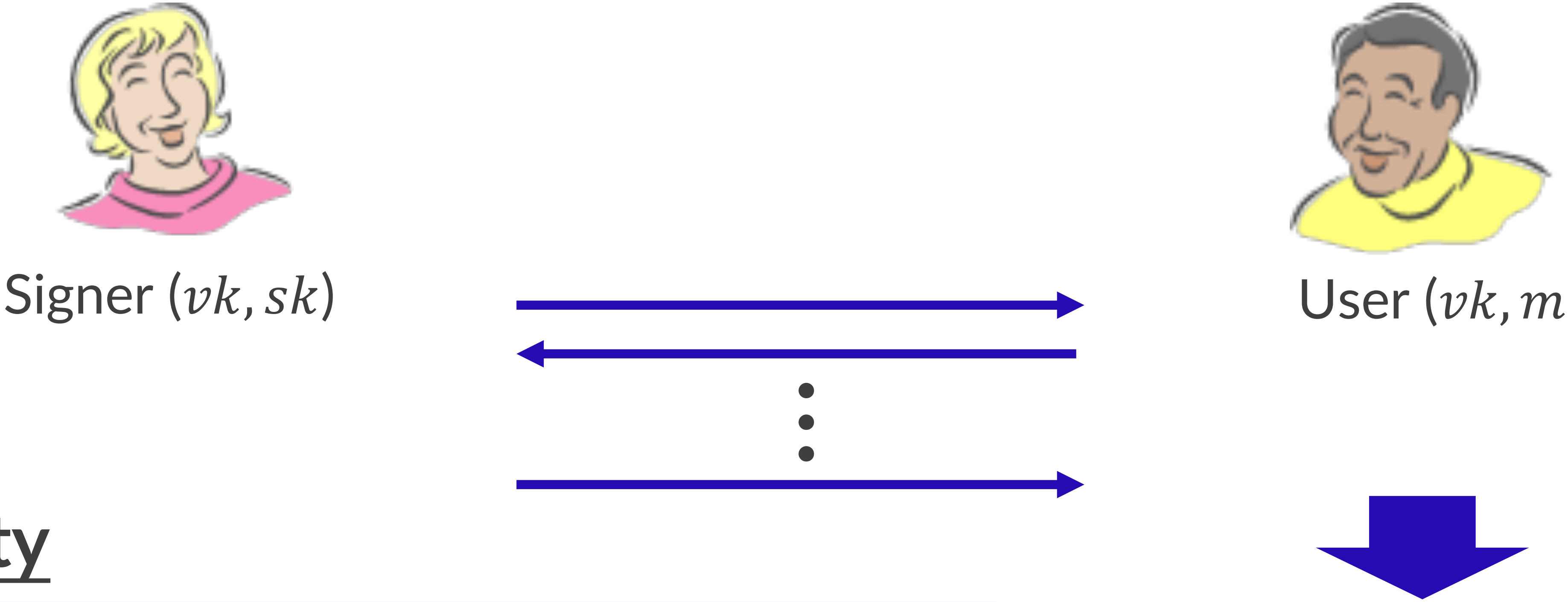
Signer (vk, sk)

User (vk, m)

Signature σ for m

What are Blind Signatures?

⇒ An interactive signing protocol with **“privacy”**.



Signer (vk, sk)

User (vk, m)

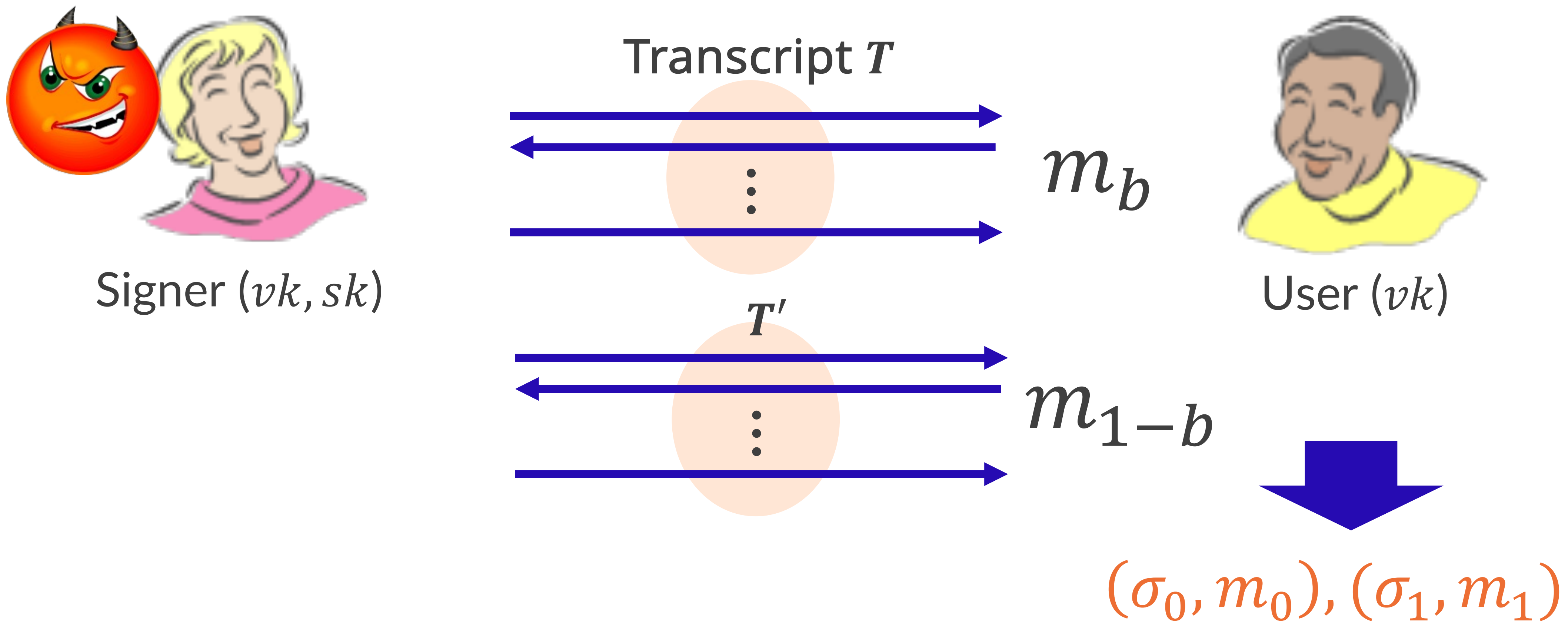
Signature σ for m

Security

Honest user ⇒ **Want m to be hidden**

Honest signer ⇒ Want unforgeability

Blindness



Transcript T

m_b

Signer (vk, sk)

User (vk)

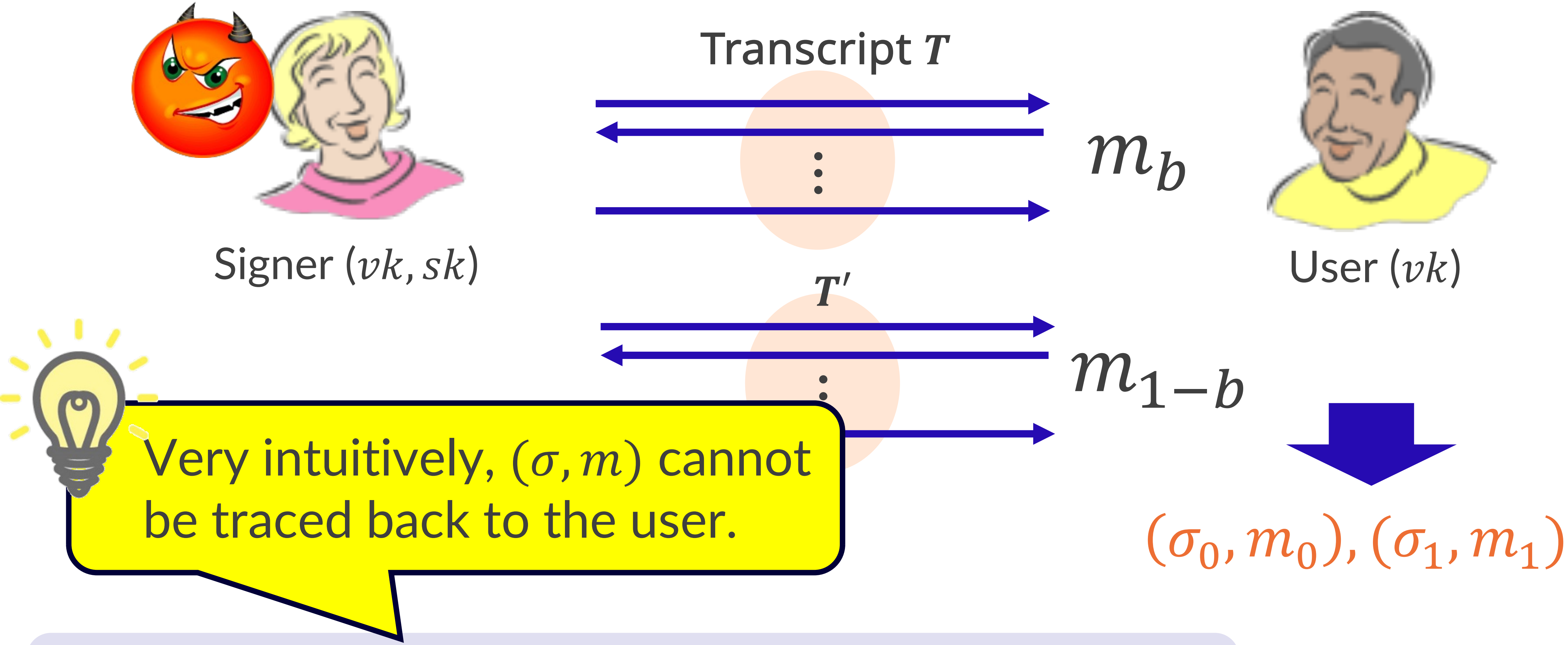
T'

m_{1-b}

$(\sigma_0, m_0), (\sigma_1, m_1)$

Given two transcripts (T, T') and $(\mathrm{m}_0, \sigma_0), (\mathrm{m}_1, \sigma_1)$, Adv cannot guess bit b.

Blindness



Very intuitively, (σ, m) cannot be traced back to the user.

Given two transcripts (T, T') and $(\mathrm{m}_0, \sigma_0), (\mathrm{m}_1, \sigma_1)$, Adv cannot guess bit b.

Applications of Blind Signatures

□ Traditional Applications

- E-cash, anonymous credentials, e-voting.

U-Prove

By Microsoft: Based on (the now "insecure") Brand's blind signature

Applications of Blind Signatures

□ Traditional Applications

- E-cash, anonymous credentials, e-voting.

U-Prove By Microsoft: Based on (the now "insecure") Brand's blind signature

□ Recent Applications

- Adding anonymity for cryptocurrency transactions [ASIACCS:YL19]
- Hiding metadata in secure messaging [CCS:KKP22]
- Privacy-preserving authentication tokens [Google22]

[ASIACCS:YL19] Yi, Xun, and Kwok-Yan Lam. "A new blind ECDSA scheme for bitcoin transaction anonymity." AsiaCCS.

[CCS:KKP22] Hashimoto, Katsumata, Prest"How to Hide MetaData in MLS-Like Secure Group Messaging: Simple, Modular, and Post-Quantum." CCS.

[Google22] "VPN by Google One, Explained" https://one.google.com/about/vpn/howitworks

Known Methods to Construct Blind Signatures

1 Blind Schnorr Type [AC:PS92]

2 Fischlin Type [C:Fis06]

Known Methods to Construct Blind Signatures

1 Blind Schnorr Type [AC:PS92]

- Very simple and efficient.
- 3-round protocol. (*Construction based on Sigma protocols.)
- Only secure up to logarithmically concurrent sessions.

2 Fischlin Type [C:Fis06]

Known Methods to Construct Blind Signatures

1 Blind Schnorr Type [AC:PS92]

- Very simple and efficient.
- 3-round protocol. (*Construction based on Sigma protocols.)
- Only secure up to logarithmically concurrent sessions.

2 Fischlin Type [C:Fis06]

- Generic construction from standard tools.
- Uses NIZK and (typically) less efficient.
- 2-round protocol.
- Secure for polynomial concurrent sessions.

What About Isogenies?

1 Blind Schnorr Type [AC:PS92]

Current construction relies on **modules/rings** but **isogenies are less expressive** ☹

2 Fischlin Type [C:Fis06]

No efficient NIZKs and compatible signatures ☹

What About Isogenies?

1 Blind Schnorr Type [AC:PS92]

Today's Talk

2 Fischlin Type [C:Fis06]

No efficient NIZKs and compatible signatures ☹

2. Reviewing Blind Schnorr

The Basics: Blind Schnorr

⇒ First Step: Interactive signing protocol **w/o blindness**.

Signer ($vk = h = g^a, sk = a$)

User ($vk = h, m$)

The Basics: Blind Schnorr

⇒ First Step: Interactive signing protocol **w/o blindness**.

Signer ($vk = h = g^a, sk = a$)

User ($vk = h, m$)

$$y \leftarrow \mathbb{Z}_p$$
$$Y = g^y \qquad \xrightarrow{\quad Y \quad}$$

The Basics: Blind Schnorr

⇒ First Step: Interactive signing protocol **w/o blindness**.

Signer $(vk = h = g^a, sk = a)$

User $(vk = h, m)$

$$y \leftarrow \mathbb{Z}_p$$

$$Y = g^y \qquad \xrightarrow{\quad Y \quad}$$

$$\xleftarrow{\quad c \quad} \qquad c \leftarrow H(Y, m)$$

The Basics: Blind Schnorr

⇒ First Step: Interactive signing protocol **w/o blindness**.

Signer $(vk = h = g^a, sk = a)$

User $(vk = h, m)$

$y \leftarrow \mathbb{Z}_p$

$Y = g^y$ $\xrightarrow{\qquad Y \qquad}$

$\xleftarrow{\qquad c \qquad}$ $c \leftarrow H(Y, m)$

$r = y - c \cdot a$ $\xrightarrow{\qquad r \qquad}$ $\sigma = (c, r)$

We have $g^r \cdot h^c = Y$.

The Basics: Blind Schnorr

⇒ First Step: Interactive signing protocol **w/o blindness.**

Signer $(vk = h = g^a, sk = a)$

User $(vk = h, m)$

$y \leftarrow \mathbb{Z}_p$

$Y = g^y$

$\xrightarrow{\quad Y \quad}$

$\xleftarrow{\quad c \quad}$ $\qquad c \leftarrow H(Y, m)$

$r = y - c \cdot a$

$\xrightarrow{\quad r \quad}$ $\qquad \sigma = (c, r)$

We have
$g^r \cdot h^c = Y.$

Not blind since σ contains the transcript.

Blinding the Schnorr Protocol

Idea: Randomize signature

$\sigma^* = (c + d, r + z)$ with $(d, z) \leftarrow \mathbb{Z}_p^2$

Signer $(vk = g^a, sk = a)$

User $(vk = h, m)$

$y \leftarrow \mathbb{Z}_p$
$Y = g^y$ $\xrightarrow{Y}$

$\xleftarrow{c}$

$r = y - c \cdot a$ $\xrightarrow{r}$

$$\boldsymbol{\sigma^* = (c^*, r^*)}$$
$$\boldsymbol{= (c + d, r + z)}$$

Blinding the Schnorr Protocol

Idea: Randomize signature

$\sigma^* = (c + d, r + z)$ with $(d, z) \leftarrow \mathbb{Z}_p^2$

Signer $(vk = g^a, sk = a)$

User $(vk = h, m)$

$y \leftarrow \mathbb{Z}_p$
$Y = g^y$

$\xrightarrow{Y}$

$(d, z) \leftarrow \mathbb{Z}_p^2$
$Y^* = \boxed{g^z \cdot} Y \boxed{\cdot h^d}$

$\xleftarrow{c}$

$r = y - c \cdot a$

$\xrightarrow{r}$

$$\boldsymbol{\sigma}^* = (\boldsymbol{c}^*, \boldsymbol{r}^*) = (\boldsymbol{c} + \boldsymbol{d}, \boldsymbol{r} + \boldsymbol{z})$$

Blinding the Schnorr Protocol

Idea: **Randomize signature**

$\sigma^* = (c + d, r + z)$ with $(d, z) \leftarrow \mathbb{Z}_p^2$

Signer $(vk = g^a, sk = a)$

User $(vk = h, m)$

$y \leftarrow \mathbb{Z}_p$
$Y = g^y$

$\xrightarrow{\quad Y \quad}$

$(d, z) \leftarrow \mathbb{Z}_p^2$
$Y^* = \boxed{g^z \cdot} Y \boxed{\cdot\, h^d}$
$c^* \leftarrow H(Y^*, m)$
$c = c^* - d$

$\xleftarrow{\quad c \quad}$

$r = y - c \cdot a$

$\xrightarrow{\quad r \quad}$

$$\boldsymbol{\sigma^* = (c^*, r^*)}$$
$$\boldsymbol{= (c + d, r + z)}$$

Blinding the Schnorr Protocol

Idea: Randomize signature

$\sigma^* = (c + d, r + z)$ with $(d, z) \leftarrow \mathbb{Z}_p^2$

Signer $(vk = g^a, sk = a)$

User $(vk = h, m)$

$y \leftarrow \mathbb{Z}_p$

$Y = g^y$

$\xrightarrow{\quad Y \quad}$

$(d, z) \leftarrow \mathbb{Z}_p^2$

$Y^* = \boxed{g^z \cdot} Y \boxed{\cdot\, h^d}$

$c^* \leftarrow H(Y^*, m)$

$c = c^* - d$

$\xleftarrow{\quad c \quad}$

$r = y - c \cdot a$

$\xrightarrow{\quad r \quad}$

$\boldsymbol{\sigma^* = (c^*, r^*)}$

$\boldsymbol{= (c + d, r + z)}$

Why correct?

original

$g^r \cdot h^c = Y$

$\Downarrow$

$g^{r+z} \cdot h^{c+d} = Y^*$

$\Downarrow$

randomized

$\boldsymbol{g^{r^*} \cdot h^{c^*} = Y^*}$

A Modular Construction from Modules

The core idea is to randomize the commitment Y **twice**.

$$\begin{array}{lcl} y \leftarrow \mathbb{Z}_p & \xrightarrow{\quad Y \quad} & (d, z) \leftarrow \mathbb{Z}_p^2 \\ Y = g^y & & Y^* = \underbrace{\boxed{g^z} \cdot Y \cdot \boxed{h^d}} \end{array}$$

Uses the fact that $\mathbb{G}$ is a $\mathbb{Z}_p$-module.

*Layman's term: Y can be multiplied with h^d.

- [EC:HKL19,C:HKLN20] abstract this and shows a **generic construction** of blind signatures based on "linear identification protocol".
- Can be instantiated by **classical groups** and **lattices.**

3. CSI-Otter
Isogeny-based Blind Signature

Review: Group Actions

$*: \mathbb{G} \times S \rightarrow S$

$$\underbrace{[\mathfrak{g}^a]}_{\text{Group element}} * \underbrace{E} = \underbrace{H}$$

Set element

Review: Group Actions

$$*\colon \mathbb{G} \times S \to S$$

$$\underbrace{[\mathfrak{g}^{a}]}_{\text{Group element}} * \underbrace{E} = \underbrace{H}$$

Set element

In Isogenies: $\mathbb{G}$ = “class group $\cong \mathbb{Z}_N$”, S = “set of elliptic curves”

*CSIDH parameters

Review: Group Actions

$$*: \mathbb{G} \times S \to S$$

$$\underbrace{[\mathfrak{g}^a]}_{\text{Group element}} * \underbrace{E}_{} = \underbrace{H}_{}$$

Set element

Example operation:

$[\mathfrak{g}^b] * H$

Review: Group Actions

$$*\colon \mathbb{G} \times S \rightarrow S$$

$$\underbrace{[\mathfrak{g}^{a}]}_{\text{Group element}} * \underbrace{E}_{} = \underbrace{H}_{}$$

Set element

Example operation:

$$[\mathfrak{g}^{b}] * H = [\mathfrak{g}^{b}] * ([\mathfrak{g}^{a}] * E)$$

Review: Group Actions

$$*\colon \mathbb{G} \times S \to S$$

$$\underbrace{[\mathfrak{g}^a]}_{\text{Group element}} * \underbrace{E} = \underbrace{H}$$

Set element

Example operation:

$$[\mathfrak{g}^b] * H = [\mathfrak{g}^b] * ([\mathfrak{g}^a] * E) = \left([\mathfrak{g}^b] \cdot [\mathfrak{g}^a]\right) * E$$

*compatibility

Review: Group Actions

$$*\colon \mathbb{G} \times S \to S$$

$$\underbrace{[\mathfrak{g}^a]}_{\text{Group element}} * \underbrace{E}_{\text{Set element}} = \underbrace{H}_{}$$

Example operation:

$$[\mathfrak{g}^b] * H = [\mathfrak{g}^b] * ([\mathfrak{g}^a] * E) = \left([\mathfrak{g}^b] \cdot [\mathfrak{g}^a]\right) * E = [\mathfrak{g}^{a+b}] * E$$

Review: Group Actions

$$*: \mathbb{G} \times S \to S$$

$$\underbrace{[\mathfrak{g}^{a}]}_{\text{Group element}} * \underbrace{E}_{} = \underbrace{H}_{}$$

Set element

Example operation:

$$[\mathfrak{g}^{b}] * H = [\mathfrak{g}^{b}] * ([\mathfrak{g}^{a}] * E) = \left([\mathfrak{g}^{b}] \cdot [\mathfrak{g}^{a}]\right) * E = [\mathfrak{g}^{a+b}] * E$$

BUT no operations over set elements! No $E \times H$!

Review: Group Actions

$$*: \mathbb{G} \times S \to S$$

$$\underbrace{[\mathfrak{g}^{a}]}_{\text{Group element}} * \underbrace{\boxed{E}}_{} = \underbrace{H}_{}$$

Set ele t

“Base” elliptic curve $E \in S$ is the generator $g \in \mathbb{G}$ in classical groups.

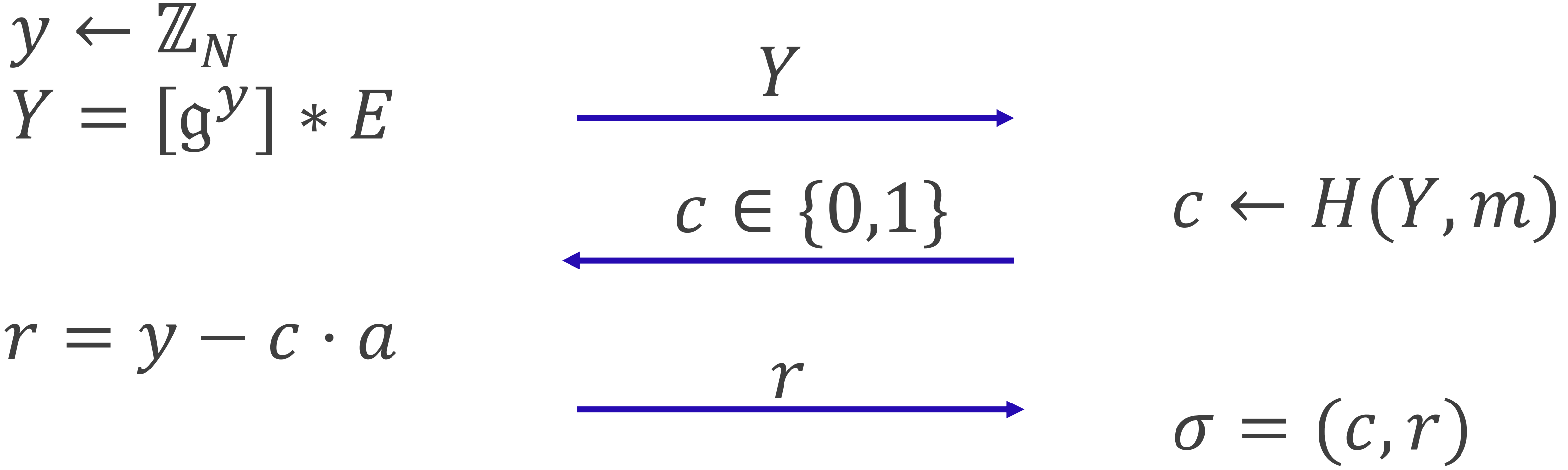
$$[\mathfrak{g}^{a}] * E \Longleftrightarrow g^{a}$$

Base Non-Blind Protocol Based on Isogeny

Due to limited structure, challenge space is now binary.

Signer
$(vk = H = [\mathfrak{g}^a] * E, sk = a)$

User $(vk = H, m)$



$y \leftarrow \mathbb{Z}_N$
$Y = [\mathfrak{g}^y] * E$

Y

$c \in \{0,1\}$

$c \leftarrow H(Y, m)$

$r = y - c \cdot a$

r

$\sigma = (c, r)$

If $c = 0$: $[\mathfrak{g}^r] * E = Y$.
If $c = 1$: $[\mathfrak{g}^r] * H = Y$.

Why Blind Schnorr Fails with Group Actions

- □ Module Setting

$$h = g^a, \qquad \begin{array}{c} y \leftarrow \mathbb{Z}_p \\ Y = g^y \end{array} \qquad \xrightarrow{\qquad Y \qquad} \qquad \begin{array}{c} (d, z) \leftarrow \mathbb{Z}_p^2 \\ Y^* = \boxed{g^z} \cdot Y \cdot \boxed{h^d} \end{array}$$

Why Blind Schnorr Fails with Group Actions

□ Module Setting

$$h = g^a, \qquad \begin{array}{l} y \leftarrow \mathbb{Z}_p \\ Y = g^y \end{array} \qquad \xrightarrow{\quad Y \quad} \qquad \begin{array}{l} (d, z) \leftarrow \mathbb{Z}_p^2 \\ Y^* = \boxed{g^z} \cdot Y \cdot \boxed{h^d} \end{array}$$

□ Group Action Setting

$$H = [\mathfrak{g}^a] * E, \qquad \begin{array}{l} y \leftarrow \mathbb{Z}_N \\ Y = [\mathfrak{g}^y] * E \end{array} \qquad \xrightarrow{\quad Y \quad} \qquad (d, z) \leftarrow \mathbb{Z}_N^2$$

Can only do

$\boxed{[\mathfrak{g}^z]} * Y$ or $\boxed{[\mathfrak{g}^d]} * \boldsymbol{H}$!!

☠ Can only randomize once!!
Not enough for blindness ☹

Here Comes the Twist ☺

💎 Isogeny has *slightly* more structure than a group action.

Given $H = [\mathfrak{g}^a] * E$,

Can compute the **quadratic twist** $H^{-1} \overset{\text{def}}{=} [\mathfrak{g}^{-a}] * E$

* “Inverse” in the classical setting: $h = g^a \Rightarrow h^{-1} = g^{-a}$

Non-Blind Protocol using Twist

First Fix: The challenge space is now $\{1, -1\}$

Signer
$(vk = H = [\mathfrak{g}^a] * E, sk = a)$

User $(vk = H, m)$

$y \leftarrow \mathbb{Z}_N$
$Y = [\mathfrak{g}^y] * E$

$\xrightarrow{Y}$

$\xleftarrow{c \in \{1, -1\}}$

$c \leftarrow H(Y, m)$

$r = y - c \cdot a$

$\xrightarrow{r}$

$\sigma = (c, r)$

$\boldsymbol{[\mathfrak{g}^r] * H^c = Y}$.

CSI-Otter: Making it Blind

Idea: Randomize signature

$\sigma^* = (\boldsymbol{c \cdot d}, \boldsymbol{r \cdot d + z})$ with $(d, z) \leftarrow \{1, -1\} \times \mathbb{Z}_N$

Signer $(vk = H, sk = a)$

User $(vk = H, m)$

$y \leftarrow \mathbb{Z}_N$

$Y = [\mathfrak{g}^y] * E$ $\xrightarrow{\quad Y \quad}$

$\xleftarrow{\quad c \quad}$

$r = y - c \cdot a$ $\xrightarrow{\quad r \quad}$

$$\begin{aligned} \boldsymbol{\sigma}^* &= (\boldsymbol{c}^*, \boldsymbol{r}^*) \\ &= (\boldsymbol{c \cdot d}, \boldsymbol{r \cdot d + z}) \end{aligned}$$

CSI-Otter: Making it Blind

Idea: Randomize signature

$\sigma^* = (\boldsymbol{c \cdot d}, \boldsymbol{r \cdot d + z})$ with $(d, z) \leftarrow \{1, -1\} \times \mathbb{Z}_N$

Signer $(vk = H, sk = a)$

User $(vk = H, m)$

$y \leftarrow \mathbb{Z}_N$

$Y = [g^y] * E \quad \xrightarrow{Y}$

$(d, z) \leftarrow \{1, -1\} \times \mathbb{Z}_N$

$Y^* = \boxed{[g^z]} * \boxed{\boldsymbol{Y^d}}$

$c^* \leftarrow H(Y^*, m)$

$\xleftarrow{c} \quad c = c^* \cdot d^{-1}$

$r = y - c \cdot a \quad \xrightarrow{r}$

$\boldsymbol{\sigma^*} = (\boldsymbol{c^*}, \boldsymbol{r^*})$

$\quad = (\boldsymbol{c \cdot d}, \boldsymbol{r \cdot d + z})$

Randomize with Quadratic Twist!!

CSI-Otter: Making it Blind

Idea: **Randomize signature**

$\sigma^* = (\boldsymbol{c \cdot d}, \boldsymbol{r \cdot d + z})$ with $(d, z) \leftarrow \{1, -1\} \times \mathbb{Z}_N$

Signer $(vk = H, sk = a)$

User $(vk = H, m)$

$y \leftarrow \mathbb{Z}_N$
$Y = [\mathfrak{g}^y] * E$

$\xrightarrow{Y}$

$(d, z) \leftarrow \{1, -1\} \times \mathbb{Z}_N$
$Y^* = \boxed{[\mathfrak{g}^z]} * \boxed{Y^d}$
$c^* \leftarrow H(Y^*, m)$
$c = c^* \cdot d^{-1}$

$\xleftarrow{c}$

$r = y - c \cdot a$

$\xrightarrow{r}$

$\boldsymbol{\sigma^* = (c^*, r^*)}$
$\boldsymbol{= (c \cdot d, r \cdot d + z)}$

Why correct?

original

$$[\mathfrak{g}^r] * H^c = Y$$
$$\Downarrow$$
$$[\mathfrak{g}^{r \cdot d}] * H^{c \cdot d} = Y^d$$
$$\Downarrow$$
$$[\mathfrak{g}^{r \cdot d + z}] * H^{c \cdot d} = [\mathfrak{g}^z] * Y^d$$
$$\Downarrow$$
$$[\mathfrak{g}^{r^*}] * H^{c^*} = Y^*$$

randomized

In Other Words, Just Another Way to Blind

□ Blind Schnorr

Randomizing signature:
$\sigma^* = (c + d, r + z)$ with $(d, z) \leftarrow \mathbb{Z}_p^2$

$$\begin{array}{ccc} y \leftarrow \mathbb{Z}_p & \xrightarrow{\quad Y \quad} & (d, z) \leftarrow \mathbb{Z}_p^2 \\ Y = g^y & & Y^* = \boxed{g^z} \cdot Y \cdot \boxed{h^d} \end{array}$$

□ CSI-Otter-like Blind Schnorr

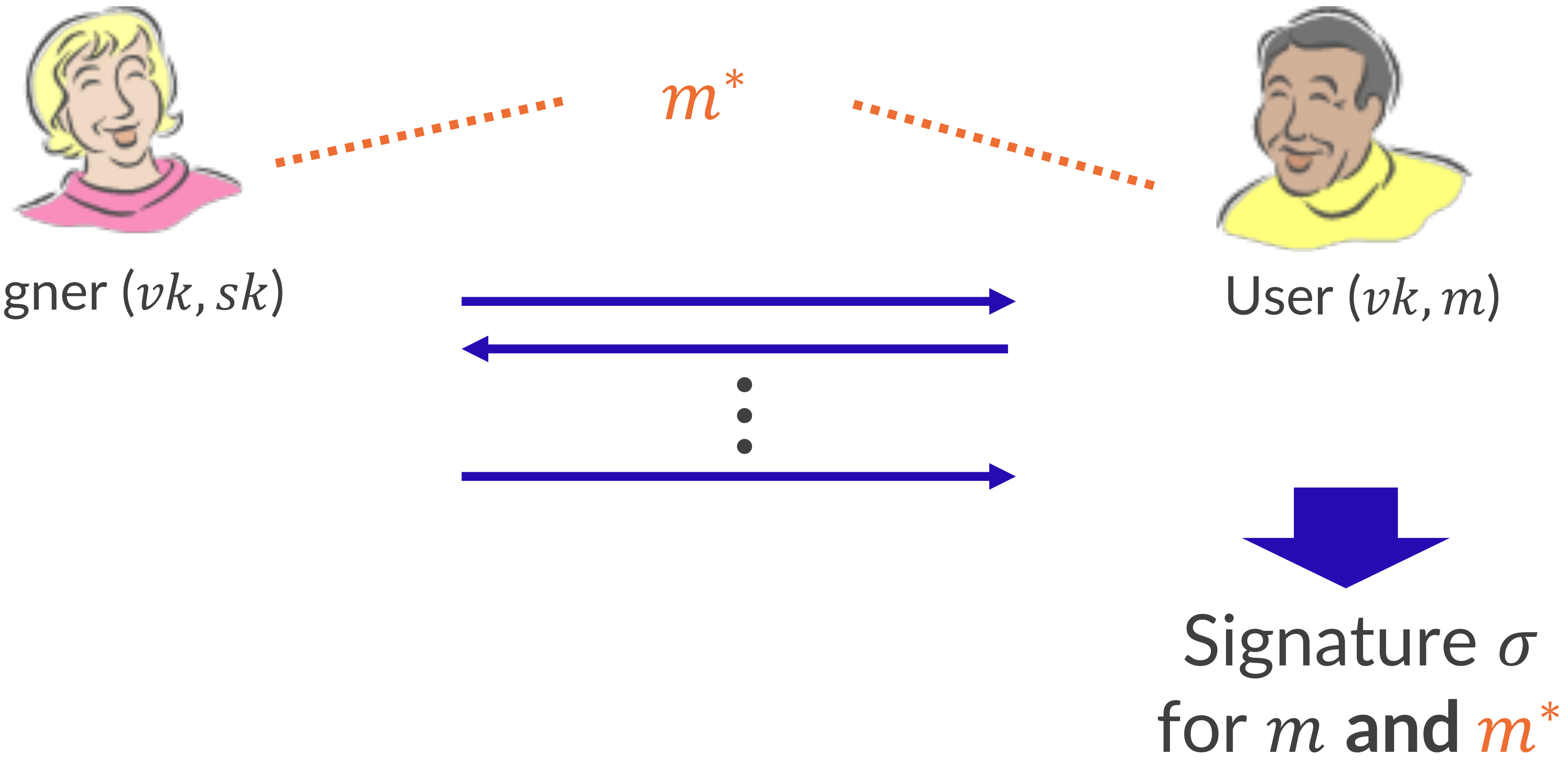
Randomizing signature:
$\sigma^* = (c \cdot d, r \cdot d + z)$ with $(d, z) \leftarrow \mathbb{Z}_p^2$

$$\begin{array}{ccc} y \leftarrow \mathbb{Z}_p & \xrightarrow{\quad Y \quad} & (d, z) \leftarrow \mathbb{Z}_p^2 \\ Y = g^y & & Y^* = \boxed{g^z} \cdot \boxed{Y^d} \end{array}$$

4. Partially Blind Signature

Partially Blind Signatures (PBS)

⇒ Allows to embed a common message m^*.



m^*

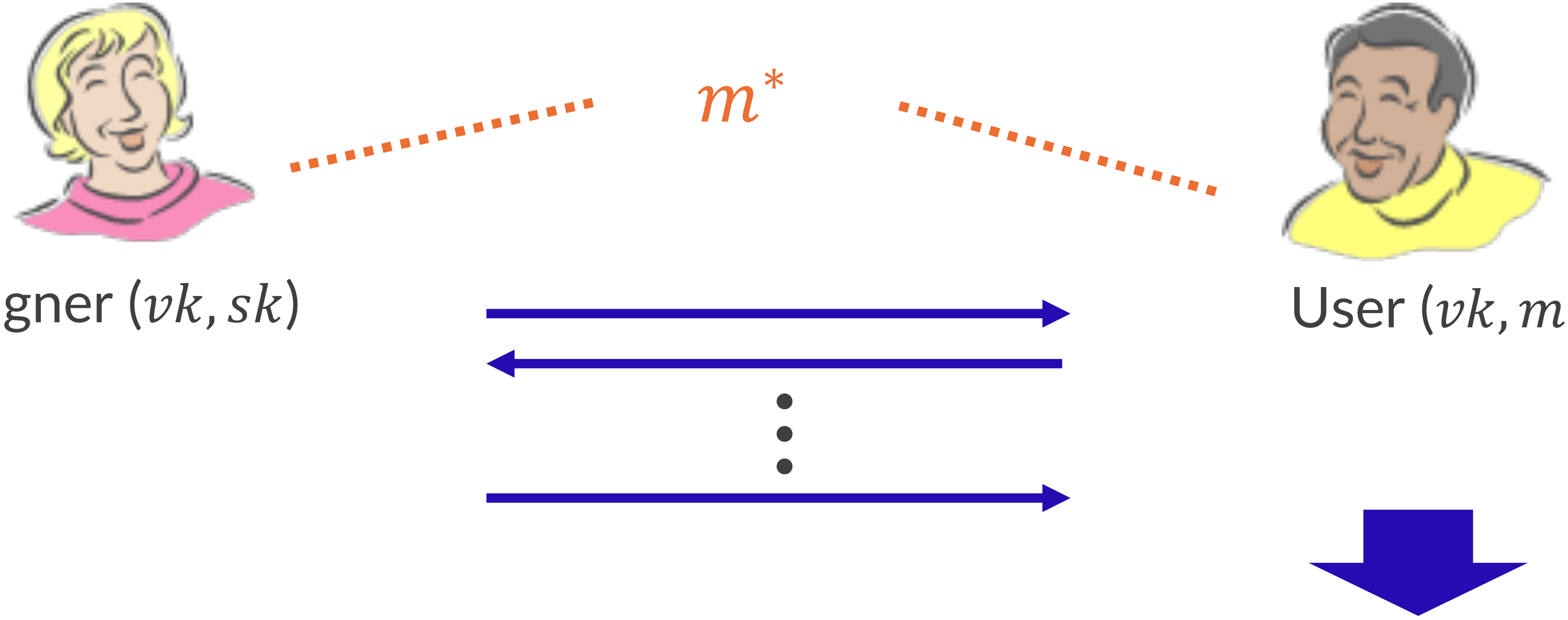
Signer (vk, sk)

User (vk, m)

Signature σ
for m **and** m^*

Partially Blind Signatures (PBS)

⇒ Allows to embed a common message m^*.



m^*

Signer (vk, sk)

User (vk, m)

Motivation: The signer can enforce rules, e.g., expiration date of signature.

Signature σ
for m **and** m^*

Strawman Idea that Doesn’t Work

⇒ Put m^* into the hash to bind it to the transcript…?

Signer $(vk = h = g^a, sk = a)$

User $(vk = h, m)$

$y \leftarrow \mathbb{Z}_p$

$Y = g^y$

$$\xrightarrow{\quad Y \quad}$$

$$\xleftarrow{\quad c \quad}$$

$$\boxed{c \leftarrow H(Y, m, \boldsymbol{m^*})}$$

$r = y - c \cdot a$

$$\xrightarrow{\quad r \quad}$$

$\sigma = (c, r)$

We have

$g^r \cdot h^c = Y.$

Strawman Idea that Doesn't Work

⇒ Put m^* into the hash to bind it to the transcript…?

Signer ($vk = h = g^a, sk = a$)

User ($vk = h, m$)

$y \leftarrow \mathbb{Z}_p$

$Y = g^y$

$\xrightarrow{Y}$

$\xleftarrow{c}$

$\boxed{c \leftarrow H(Y, m, \hat{m})}$

$r = y - c \cdot a$

$\xrightarrow{r}$

$\sigma = (c, r)$

We have $g^r \cdot h^c = Y$.

No way for the signer to check this!

Idea that Works [C:AO00]

Signer ($vk = h = g^a, sk = a$)

In Blind Schnorr, signer was implicitly proving knowledge of …

$$a \in \mathbb{Z}_p \text{ s.t. } h = g^a$$

In Partially Blind Schnorr, we modify so that the signer proves …

$$a \in \mathbb{Z}_p \text{ s.t. } h = g^a \vee G(m^*) = h^* = g^a$$

${}^*G$: random oracle

Why It Fails for Isogenies

Classical Group: $a \in \mathbb{Z}_p$ s.t. $h = g^a \vee G(m^*) = h^* = g^a$

Isogeny: $G(m^*) = H^* = [\mathfrak{g}^a] * E$

Why It Fails for Isogenies

Classical Group: $a \in \mathbb{Z}_p$ s.t. $h = g^a \vee G(m^*) = h^* = g^a$

Isogeny: $G(m^*) \xcancel{=} H^* = [\mathfrak{g}^a] * E$

In isogeny, we don't know how to hash into the set of elliptic curves w/o knowing <u>secret $\boldsymbol{a}$</u>.

Our Idea: Extending to a 2-out-of-3 Proof

Signer $(vk = (h_0, h_1) = (g^{a_0}, g^{a_1}), sk = a_b)$

Prove knowledge of 2-out-of-3 exponents.

$$h_0 = g^{a_0} \;\vee\; h_1 = g^{a_1} \;\vee\; h^* = g^{\boxed{a^*}} = g^{\boxed{G(m^*)}}$$

- Everybody knows secret a^* but this won’t be enough to sign.
- Can still blind this 2-out-of-3 protocol to build a PBS.

Omitted Details from Talk

- □ Formal security proof of CSI-Otter using [AC:KLX22]

- □ Optimizations using higher degree roots of unity.
  ⇒ New ζ_d-ring group action inverse problem

- □ On-going work:
  - ➢ On first glace, ROS attack does not apply.
  - ➢ One-more unf. in the poly-concurrent regime...?

Thank You For Listening ☺

A new Schnorr-type 3-round **blind signature** based on **isogenies** (CSIDH).

- The **first** (partially) blind signature from isogenies.
- Provable security for log-concurrent sessions.
- New hardness assumption for optimization.