Concurrently Secure Blind Schnorr Signatures

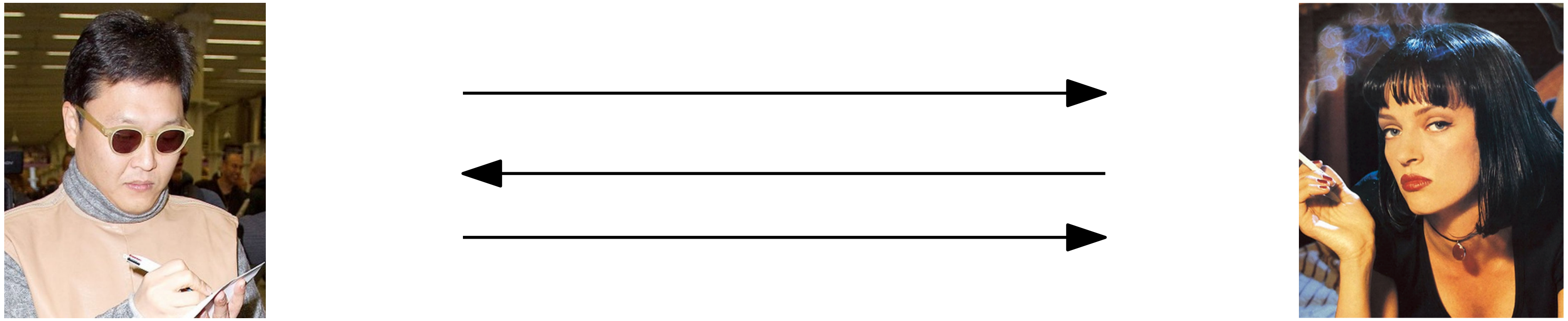
Georg Fuchsbauer
Mathias Wolf

EUROCRYPT’24

Blind signatures

PSY

Uma



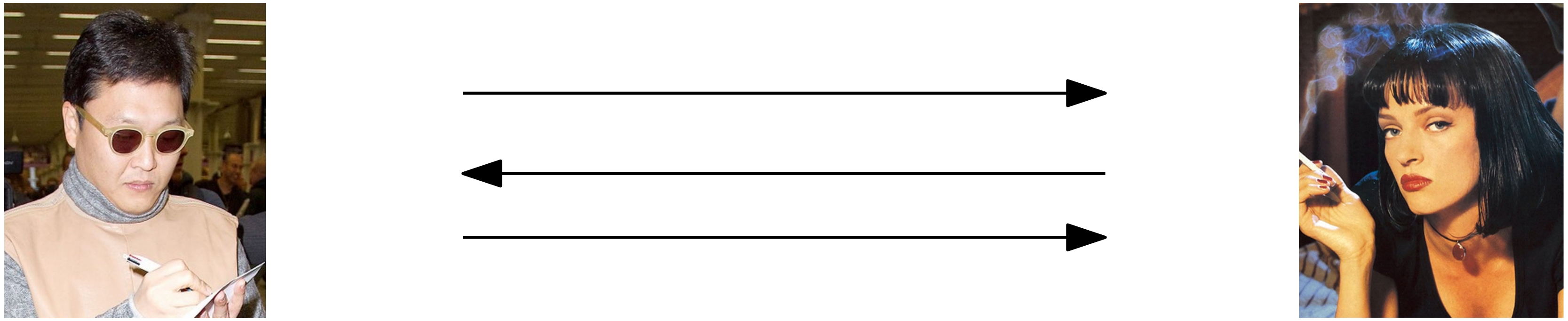
Signer (vk, sk)

User (vk, m)

$\sigma : \mathsf{Ver}(vk, m, \sigma) = 1$

- **Blindness**
- (one-more) **Unforgeability**

Blind signatures



Signer (vk, sk)

User (vk, m)

$$\downarrow$$

$$\sigma : \mathsf{Ver}(vk, m, \sigma) = 1$$

- e-cash
- e-voting
- anonymous credentials
- contact tracing
- advanced VPNs
- Bitcoin blind coin swaps

- private relays
- private access tokens
- Privacy Pass

Our contributions

- Many schemes . . .
  . . . but with *non-standard* signature format

- Here: blind signing for
  Schnorr signatures

. . . but **no concurrently secure blind signing protocol**

(1) Construct **secure protocol** (using NIZKs)

(2) Propose **generalization** of blind signatures

(3) **Implement** different instantiations

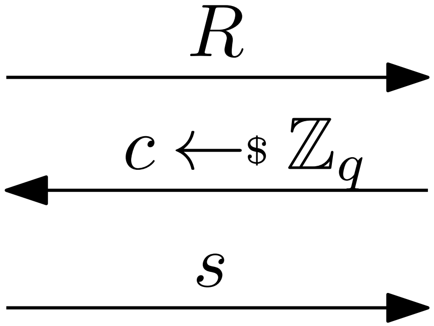
Schnorr signatures

- Group $(q, \mathbb{G}, G)$, hash fct H: $\{0,1\}^* \to \mathbb{Z}_q$
- $sk = x \leftarrow_{\$} \mathbb{Z}_q$, $vk := X = xG$
- Prove I know x

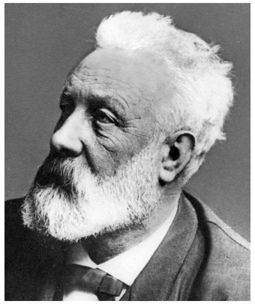
$$r \leftarrow_{\$} \mathbb{Z}_q;\, R := rG$$

$$s := [r + cx]_q$$

Prover



$$sG \overset{?}{=} R + cX$$



Verifier

Schnorr signatures

- Group $(q, \mathbb{G}, G)$, hash fct H: $\{0,1\}^* \to \mathbb{Z}_q$
- $sk = x \leftarrow_{\$} \mathbb{Z}_q$, $vk := X = xG$

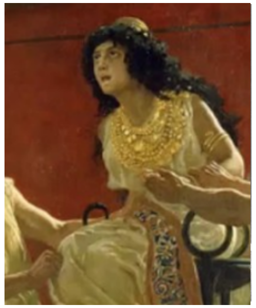
- Sign(x, m):

  $r \leftarrow_{\$} \mathbb{Z}_q; R := rG$

  $c := \mathsf{H}(R, X, m)$

  $s := [r + cx]_q$

  $\sigma := (R, s)$



Oracle

$$sG \overset{?}{=} R + cX$$

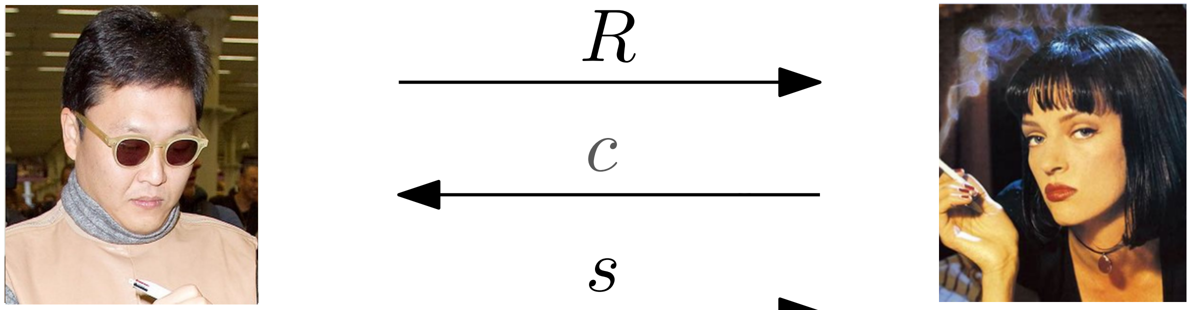
Schnorr signatures are **unforgeable**
in the ROM assuming $\mathbb{G}$ is DL-hard

Blind Schnorr signatures

- Group $(q, \mathbb{G}, G)$, hash fct H: $\{0,1\}^* \to \mathbb{Z}_q$
- $sk = x \leftarrow_\$ \mathbb{Z}_q$, $vk := X = xG$

$$R := rG$$

$$s := [r + cx]_q$$



Signer x

User m

Verifier:

- $c := \mathsf{H}(R, X, m)$

. . . but signer knows (R, s)

Blind Schnorr signatures

$$
\begin{aligned}
s'G &= \underbrace{sG} + \alpha G \\
&= \underbrace{R} + cX \quad + \alpha G \\
&= R' - \alpha G - \beta X \quad + \underbrace{cX} \quad + \alpha G \\
&= R' - \alpha \cancel{G} - \beta \cancel{X} + (c' - \cancel{\beta})X + \cancel{\alpha} G \\
&= R' + c'X
\end{aligned}
$$

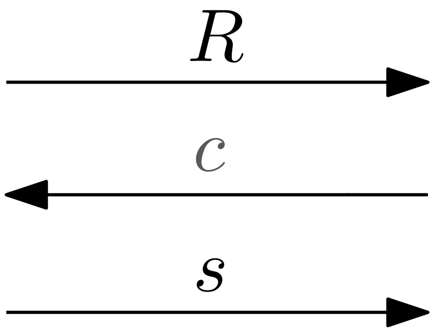
$: \{0,1\}^* \to \mathbb{Z}_q$

cG

$R := rG$

$s := [r + cx]_q$

Signer x



User m

User:

- $\alpha, \beta \leftarrow_\$ \mathbb{Z}_q$
- $R' := R + \alpha G + \beta X$
- $c := [\mathsf{H}(R', X, m) + \beta)]_q$

Then $(R', s' := [s + \alpha]_q)$ *is a signature on* m*!*

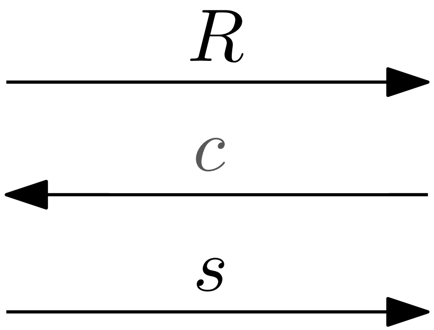
Blind Schnorr signatures

- Group $(q, \mathbb{G}, G)$, hash fct $\mathsf{H}\colon \{0,1\}^* \to \mathbb{Z}_q$

Blindness

- R' and c' are perfectly hidden from signer
- s' is unique element so signature valid

(One-more) **Unforgeability**?

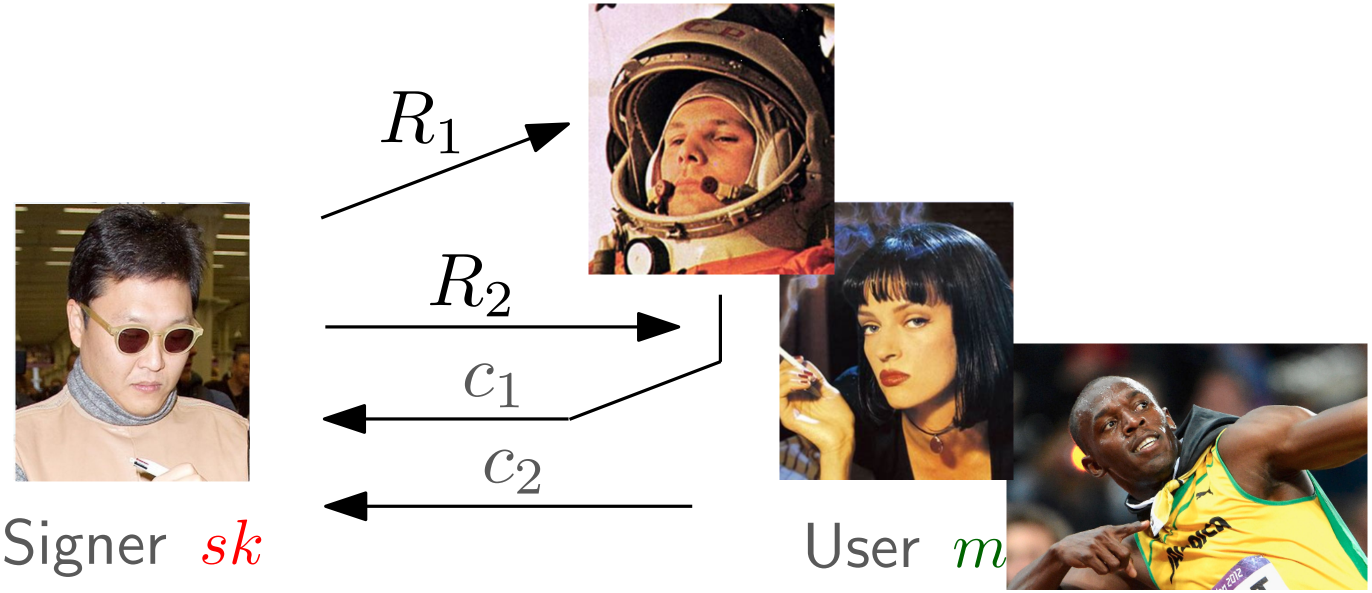


User m

User:
- $\alpha, \beta \leftarrow_{\$} \mathbb{Z}_q$
- $R' := R + \alpha G + \beta X$
- $c := [\mathsf{H}(R', X, m) + \beta)]_q$

Then $(R', s' := [s + \alpha]_q)$ *is a signature on* m*!*

Unforgeability of blind Schnorr

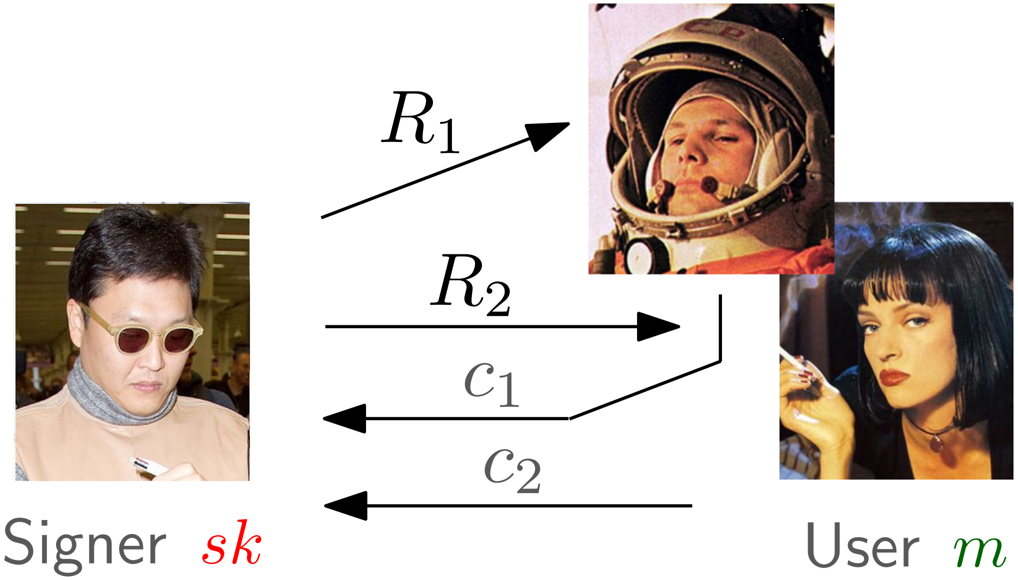


(One-more) **Unforgeability** (under concurrent attack)**?**

$\Longleftrightarrow$ "ROS" in GGM+ROM

[Sch01]

Unforgeability of blind Schnorr



(One-more) **Unforgeability** (~~under concurrent attack~~)**?**

- **Practical attack:**

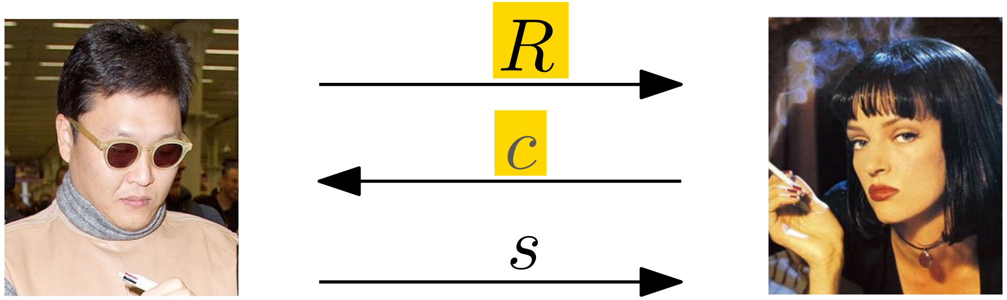
λ queries

$\Rightarrow \lambda + 1$ signatures

[Ben$^+$21]

Can we define a **secure** *blind-signing protocol for Schnorr signatures?*

Secure blind Schnorr signing. . .



Signer sk User m

$\alpha, \beta \leftarrow_{\$} \mathbb{Z}_q$

$R' := R + \alpha G + \beta X$

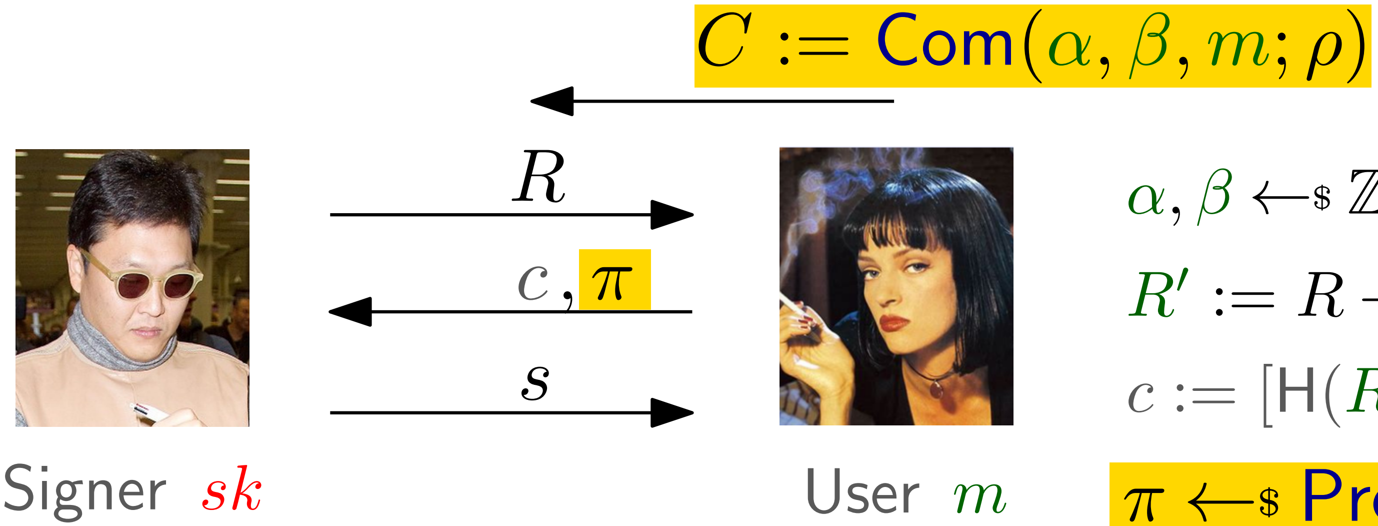
$c := [\mathsf{H}(R', X, m) + \beta)]_q$

return $(R', s' := [s + \alpha]_q)$

- **Practical attack:**
- c_i's depend on **all** R_i's

[Ben$^+$21]

Secure blind Schnorr signing. . .



$$\xleftarrow{C := \mathsf{Com}(\alpha, \beta, m; \rho)}$$

$$\xrightarrow{R}$$

$$\xleftarrow{c\,,\,\pi}$$

$$\xrightarrow{s}$$

Signer sk User m

$\alpha, \beta \leftarrow_\$ \mathbb{Z}_q$

$R' := R + \alpha G + \beta X$

$c := [\mathsf{H}(R', X, m) + \beta)]_q$

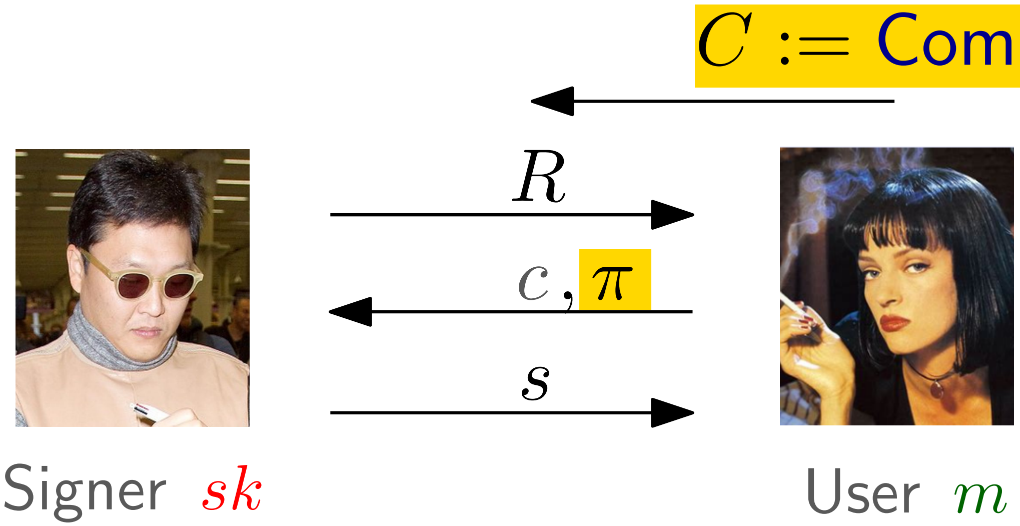
$\pi \leftarrow_\$ \mathsf{Prove}(\ldots$
$w = (\alpha, \beta, m, \rho))$

return $(R', s' := [s + \alpha]_q)$

$\Rightarrow$ Let U **commit** to her secrets **before**

. . . and **prove** (in ZK) that c is consistent with C

Secure blind Schnorr signing. . .



$C := \mathsf{Com}(\alpha, \beta, m; \rho)$

R

c, π

s

Signer sk

User m

$\alpha, \beta \leftarrow_{\$} \mathbb{Z}_q$

$R' := R + \alpha G + \beta X$

$c := [\mathsf{H}(R', X, m) + \beta)]_q$

$\pi \leftarrow_{\$} \mathsf{Prove}(\ldots$
$w = (\alpha, \beta, m, \rho))$

return $(R', s' := [s + \alpha]_q)$

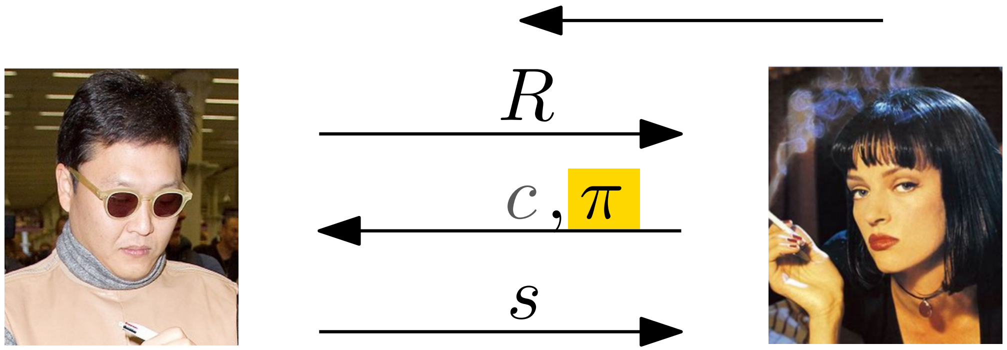
Theorem.

(One-more) **Unforgeability** (under concurrent attack)

$\Leftarrow$ unforgeability of Schnorr

$+$ soundness of Prove

Implementations



Signer sk

User m

$$\alpha, \beta \leftarrow_{\$} \mathbb{Z}_q$$

$$R' := R + \alpha G + \beta X$$

$$c := [\mathsf{H}(R', X, m) + \beta)]_q$$

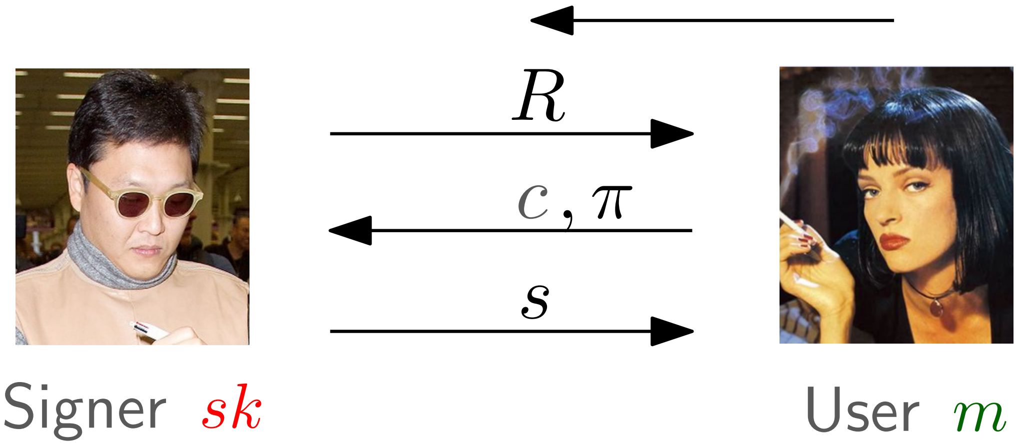
$$\pi \leftarrow_{\$} \mathsf{Prove}(\ldots \quad w = (\alpha, \beta, m, \rho))$$

3 **implementations** using zk-SNARKs

(1) Groth16 (2) $\mathcal{P}$lon$\mathcal{K}$

$\|\pi\|$	t_{Verify}
$\approx 1\,\text{kB}$	$\ll 1\,\text{sec}$

Implementations



$\alpha, \beta \leftarrow_\$ \mathbb{Z}_q$

$R' := R + \alpha G + \beta X$

$c := [\mathsf{H}(R', X, m) + \beta)]_q$

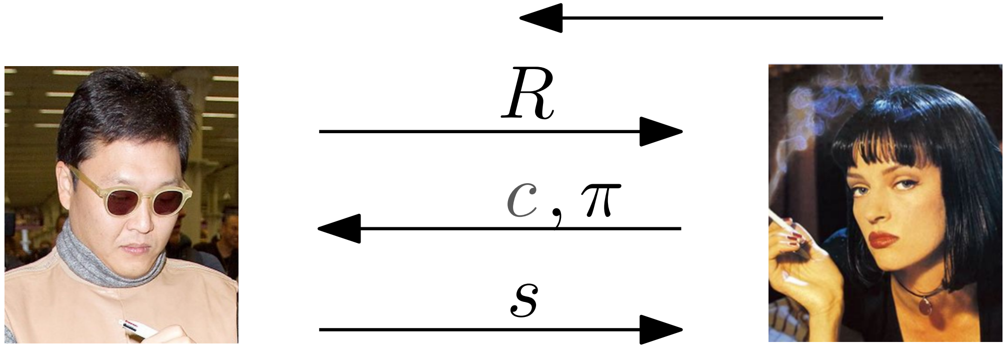
$\pi \leftarrow_\$ \mathsf{Prove}(\ldots$

$w = (\alpha, \beta, m, \rho))$

(1) Groth16 (2) $\mathcal{P}\mathfrak{lon}\mathcal{K}$

Elliptic curve	Hash function	$\lvert crs \rvert$	t_{Prove}
Baby JubJub	Poseidon	< 1 MB	≈ 1 sec
₿ secp256k1	SHA-256	550 MB	≈ 1 min (cf. ⓩ)

Implementations



Signer sk

User m

$\alpha, \beta \leftarrow_{\$} \mathbb{Z}_q$

$R' := R + \alpha G + \beta X$

$c := [\mathsf{H}(R', X, m) + \beta)]_q$

$\pi \leftarrow_{\$} \mathsf{Prove}(\ldots$

$w = (\alpha, \beta, m, \rho))$

(3) Spartan

	Elliptic curve	Hash function	$\lvert crs \rvert$	t_{Prove}
₿	secp256k1	SHA-256	36 kB	≈ 2.5 min

Implementations

Signer sk

$\xleftarrow{\qquad}$

$\xrightarrow{R}$

$\xleftarrow{c\,,\,\pi}$

$\xrightarrow{s}$

User m

$\alpha, \beta \leftarrow_{\$} \mathbb{Z}_q$

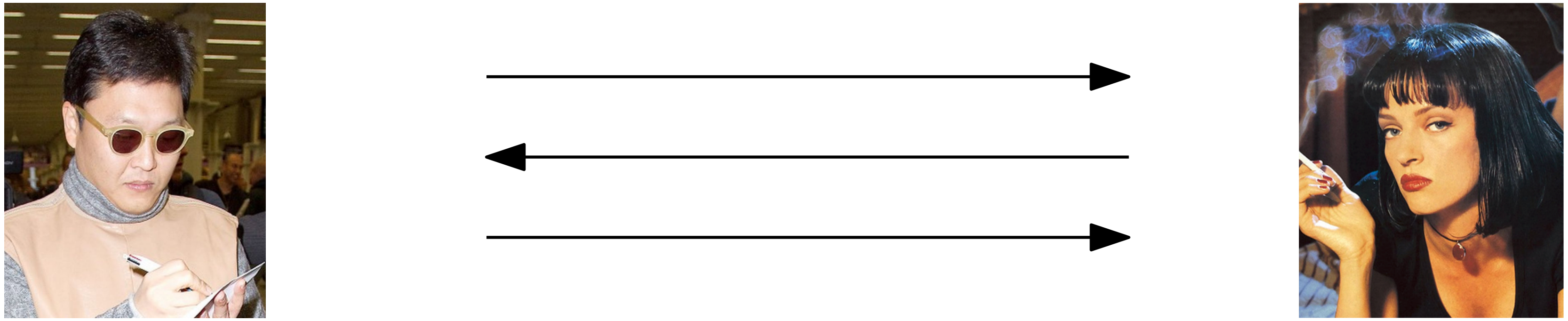
$R' := R + \alpha G + \beta X$

$c := [\mathsf{H}(R', X, m) + \beta)]_q$

$\pi \leftarrow_{\$} \mathsf{Prove}(\ldots$
$\quad w = (\alpha, \beta, m, \rho))$

. . . *but* U *could also prove any* **predicate** *of the message*

Predicate blind signatures



Signer sk vk, *pred* User m

…is guaranteed
m satisfies *pred*

…is guaranteed
signer learns
nothing else

- *Generalization of* **partially blind** *signatures*

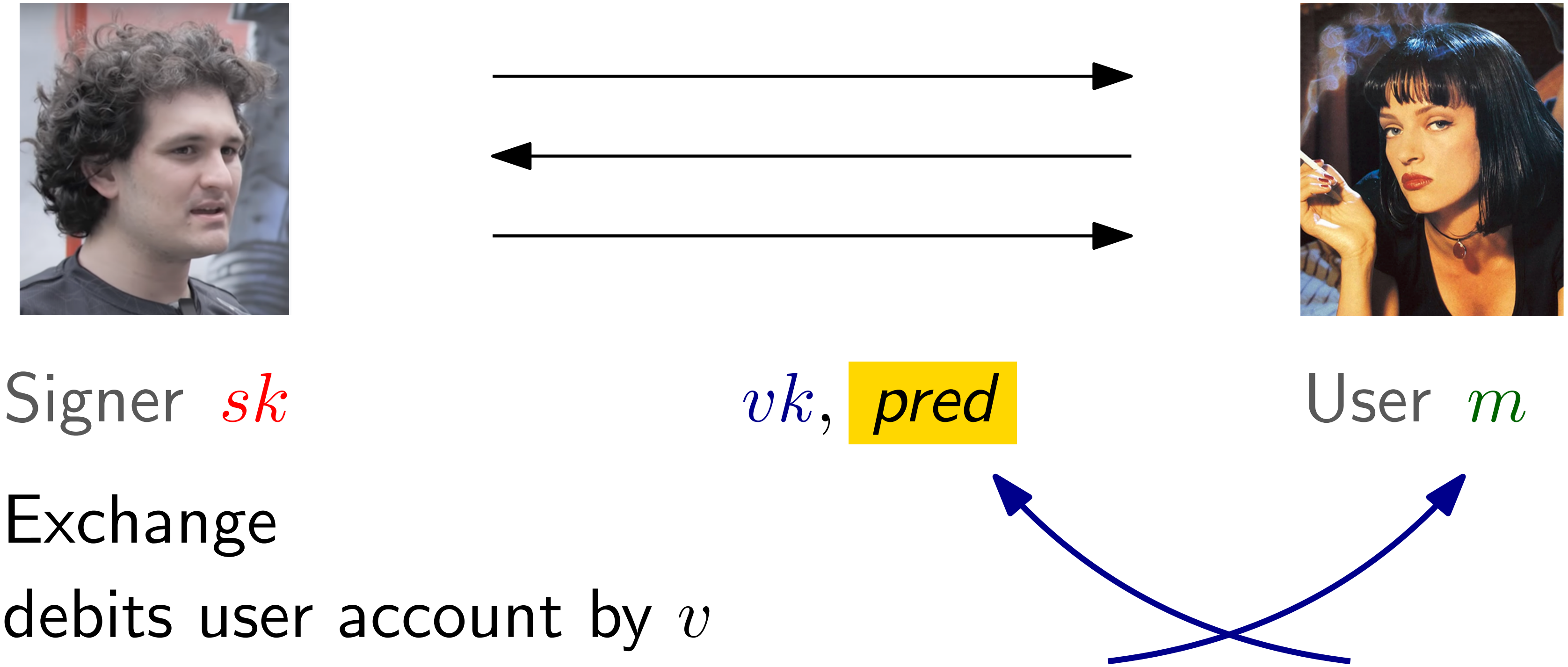
Applications

- Blind signing for signatures already deployed

- e-cash
- e-voting
- anonymous credentials
- contact tracing
- advanced VPNs
- Bitcoin blind coin swaps

- private relays
- private access tokens
- Privacy Pass

Blind Bitcoin payments



Signer sk

vk, pred

User m

$\Rightarrow$ Exchange

debits user account by v

BTC tx, “amount $\leq v$”