On the (in)security of ROS

Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, Mariana Raykova

https://ia.cr/2020/945

On the (in)security of (some) blind signatures

Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, Mariana Raykova

https://ia.cr/2020/945

On the (in)security of (some) multi-signatures

Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, Mariana Raykova

https://ia.cr/2020/945

On the (in)security of (some) threshold signatures

Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, Mariana Raykova

https://ia.cr/2020/945

On the (in)security of (some) e-cash systems

Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, Mariana Raykova

https://ia.cr/2020/945

On the (in)security of (some) anonymous credentials

Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, Mariana Raykova

https://ia.cr/2020/945

ROS problem
[Schnorr01]

Random inhomogeneities in a Overdetermined Solvable system of linear equations.

Find $\overrightarrow{c} \in \mathbb{F}_p^\ell$ **and** $\overrightarrow{\rho_0}, \ldots, \overrightarrow{\rho_\ell} \in \mathbb{F}_p^\ell$ **such that:**

$$\mathsf{H}(\overrightarrow{\rho}_j) = \langle \overrightarrow{\rho}_j, \overrightarrow{c} \rangle$$

ROS problem

$$\ell + 1 \; \begin{bmatrix} \mathsf{H}(\vec{\rho}_0) \\ \mathsf{H}(\vec{\rho}_1) \\ \mathsf{H}(\vec{\rho}_2) \\ \vdots \\ \mathsf{H}(\vec{\rho}_{\ell-1}) \\ \mathsf{H}(\vec{\rho}_{\ell}) \end{bmatrix} = \overbrace{\begin{bmatrix} \rho_{0,0} & \cdots & \rho_{0,\ell-1} \\ \rho_{1,0} & \cdots & \rho_{1,\ell-1} \\ \rho_{2,0} & \cdots & \rho_{2,\ell-1} \\ & \vdots & \\ \rho_{\ell-1,0} & \cdots & \rho_{\ell-1,\ell-1} \\ \rho_{\ell,0} & \cdots & \rho_{\ell,\ell-1} \end{bmatrix}}^{\ell} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{\ell-1} \end{bmatrix} \; \ell$$

ROS problem: partial solutions

$$\ell+1 \left\{ \begin{bmatrix} H(100\cdots0) \\ H(010\cdots0) \\ H(001\cdots0) \\ \vdots \\ H(000\cdots1) \\ H(?) \end{bmatrix} \right. = \overbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ \multicolumn{5}{c}{\text{??}} \end{bmatrix}}^{\ell} \cdot \left. \begin{bmatrix} H(100\cdots0) \\ H(010\cdots0) \\ H(001\cdots0) \\ \vdots \\ H(000\cdots1) \end{bmatrix} \right\} \ell$$

ROS problem: partial solutions

$$\ell+1\left\{\begin{pmatrix} \mathrm{H}(200\cdots0) \\ \mathrm{H}(020\cdots0) \\ \mathrm{H}(002\cdots0) \\ \vdots \\ \mathrm{H}(000\cdots2) \\ \mathrm{H}(?) \end{pmatrix}\right. = \overbrace{\begin{pmatrix} 2 & 0 & 0 & \cdots & 0 \\ 0 & 2 & 0 & \cdots & 0 \\ 0 & 0 & 2 & \cdots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \cdots & 2 \\ \multicolumn{5}{c}{\text{??}} \end{pmatrix}}^{\ell} \cdot \left.\begin{pmatrix} {}^{1}/_{2}\mathrm{H}(200\cdots0) \\ {}^{1}/_{2}\mathrm{H}(020\cdots0) \\ {}^{1}/_{2}\mathrm{H}(002\cdots0) \\ \vdots \\ {}^{1}/_{2}\mathrm{H}(000\cdots2) \end{pmatrix}\right\}\ell$$

ROS problem: previous results

[Wagner02]

- Generalised birthday attack:
  - Input lists $L_0, \ldots, L_\ell$ with random elements
  - Find elements in the list $x_0, \ldots, x_\ell$ in the lists such that

$$x_0 + \cdots + x_\ell = 0$$

➡ For $\ell = 1$: birthday attack $O(\sqrt{p})$

➡ Runs in time $O((\ell + 1) \cdot \sqrt[1 + \lfloor \log(\ell + 1) \rfloor]{p})$.

Wagner's attack

$$
\ell + 1 \left\{ \begin{pmatrix} \mathrm{H}(\overrightarrow{\rho}_0) \\ \mathrm{H}(\overrightarrow{\rho}_1) \\ \mathrm{H}(\overrightarrow{\rho}_2) \\ \vdots \\ \mathrm{H}(\overrightarrow{\rho}_{\ell-1}) \\ \mathrm{H}(\mathbf{1}) \end{pmatrix} \right. = \overbrace{\begin{pmatrix} & & & \overrightarrow{\rho}_0 & \\ & & & \overrightarrow{\rho}_1 & \\ & & & \overrightarrow{\rho}_2 & \\ & & & \vdots & \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix}}^{\ell} \cdot \left. \begin{pmatrix} \mathrm{H}(\overrightarrow{\rho}_0) \\ \mathrm{H}(\overrightarrow{\rho}_1) \\ \mathrm{H}(\overrightarrow{\rho}_2) \\ \vdots \\ \mathrm{H}(\overrightarrow{\rho}_{\ell-1}) \end{pmatrix} \right\} \ell
$$

Breaking ROS in polynomial time

Our contribution

ROS Attack

$$\ell+1\;\begin{pmatrix} H(\overrightarrow{\rho}_0) \\ H(\overrightarrow{\rho}_1) \\ H(\overrightarrow{\rho}_2) \\ \vdots \\ H(\overrightarrow{\rho}_{\ell-1}) \\ H(\overrightarrow{\rho}_\ell) \end{pmatrix} = \overset{\ell}{\begin{pmatrix} \overrightarrow{\rho}_0 \\ \overrightarrow{\rho}_1 \\ \overrightarrow{\rho}_2 \\ \vdots \\ \overrightarrow{\rho}_\ell \end{pmatrix}} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{\ell-1} \end{pmatrix}\;\ell$$

ROS Attack: idea

For each $j \in \{0, \ldots, \ell - 1\}$:

$$f_j(x_j) := \begin{cases} 0 & \text{if } x_j = \mathrm{H}(\overrightarrow{\rho}_j^{(0)}) \\ 2^j & \text{if } x_j = \mathrm{H}(\overrightarrow{\rho}_j^{(1)}) \end{cases}$$

$\overrightarrow{\rho}_0^{(0)} = (1, 0, \cdots, 0)$, etc.

$\overrightarrow{\rho}_0^{(1)} = (2, 0, \ldots, 0)$, etc.

** Skipping the constant factor in the coefficient for simplicity

ROS Attack: idea

For each $j \in \{0,\ldots,\ell-1\}$:

$$f_j(x_j) := 2^j \cdot \frac{x_j - \mathsf{H}(\overrightarrow{\rho}_j^{(0)})}{\mathsf{H}(\overrightarrow{\rho}_j^{(1)}) - \mathsf{H}(\overrightarrow{\rho}_j^{(0)})}$$

** Skipping the constant factor in the coefficient for simplicity

ROS Attack: idea

For each $j \in \{0,\ldots,\ell - 1\}$:

$$f_j(x_j) := 2^j \cdot \frac{x_j - c_j^{(1)}}{c_j^{(1)} - c_j^{(0)}}$$

ROS Attack: idea

$$\rho(x_0, \ldots, x_{\ell-1}) = f_0 + f_1 + \ldots + f_{\ell-1}$$

$$= \rho_0 x_0 + \ldots + \rho_{\ell-1} x_{\ell-1} + \rho_\ell$$

ROS Attack: idea

Let $\ell > \log p$. For any $n \in \mathbb{F}_p$:

$$n = \sum_{j=0}^{\ell-1} 2^j \cdot b_j$$ [Binary decomposition]

$$= \sum_{j=0}^{\ell-1} f_j(c_j^{(b_j)})$$ [By definition of f_j]

$$= \rho(c_0^{(b_0)}, \ldots, c_{\ell-1}^{(b_{\ell-1})})$$ [Because $\rho(x_0, \ldots, x_{\ell-1}) := f_0 + f_1 + \ldots + f_{\ell-1}$]

ROS Attack: idea

Let $\ell > \log p$. For $n = \mathsf{H}((\rho_0, \ldots, \rho_{\ell-1})) + \rho_\ell$

$$n = \sum_{j=0}^{\ell-1} 2^j \cdot b_j$$ [Binary decomposition]

$$= \sum_{j=0}^{\ell-1} f_j(c_j^{(b_j)})$$ [By definition of f_j]

$$= \rho(c_0^{(b_0)}, \ldots, c_{\ell-1}^{(b_{\ell-1})})$$ [Because $\rho(x_0, \ldots, x_{\ell-1}) := f_0 + f_1 + \ldots + f_{\ell-1}$]

$$\Longrightarrow \mathsf{H}((\rho_0, \ldots, \rho_{\ell-1})) = \sum_{j=0}^{\ell-1} \rho_j \, c_j^{(b_j)}$$

ROS Attack: idea

Let $\ell > \log p$. For $n = \mathsf{H}((\rho_0, \ldots, \rho_{\ell-1})) + \rho_\ell$

$$n = \sum_{j=0}^{\ell-1} 2^j \cdot b_j$$ [Binary decomposition]

$$= \sum_{j=0}^{\ell-1} f_j(c_j^{(b_j)})$$ [By definition of f_j]

$$= \rho(c_0^{(b_0)}, \ldots, c_{\ell-1}^{(b_{\ell-1})})$$ [Because $\rho(x_0, \ldots, x_{\ell-1}) := f_0 + f_1 + \ldots + f_{\ell-1}$]

$$\Longrightarrow \mathsf{H}((\rho_0, \ldots, \rho_{\ell-1})) = \sum_{j=0}^{\ell-1} \rho_j \, c_j^{(b_j)} = \langle (\rho_0, \ldots, \rho_{\ell-1}), \overrightarrow{c} \rangle \quad \blacksquare$$

Attack summary

- Pick different basis $\overrightarrow{\rho}_0^{(0)}, \ldots, \overrightarrow{\rho}_{\ell-1}^{(0)}$ and $\overrightarrow{\rho}_0^{(1)}, \ldots, \overrightarrow{\rho}_{\ell-1}^{(1)}$
- Hash them to obtain $c_j^{(b)}$.
- Interpolate and compute $\rho(x_0, \ldots, x_{\ell-1})$
- Decompose H$((\rho_0, \ldots, \rho_{\ell-1})) + \rho_\ell = (b_0, \ldots, b_{\ell-1})_2$.

Nontrivial solution: $(\rho_0, \ldots, \rho_{\ell-1})$

Easy solutions: $\overrightarrow{\rho}_0^{(b_0)}, \ldots, \overrightarrow{\rho}_{\ell-1}^{(b_{\ell-1})}$

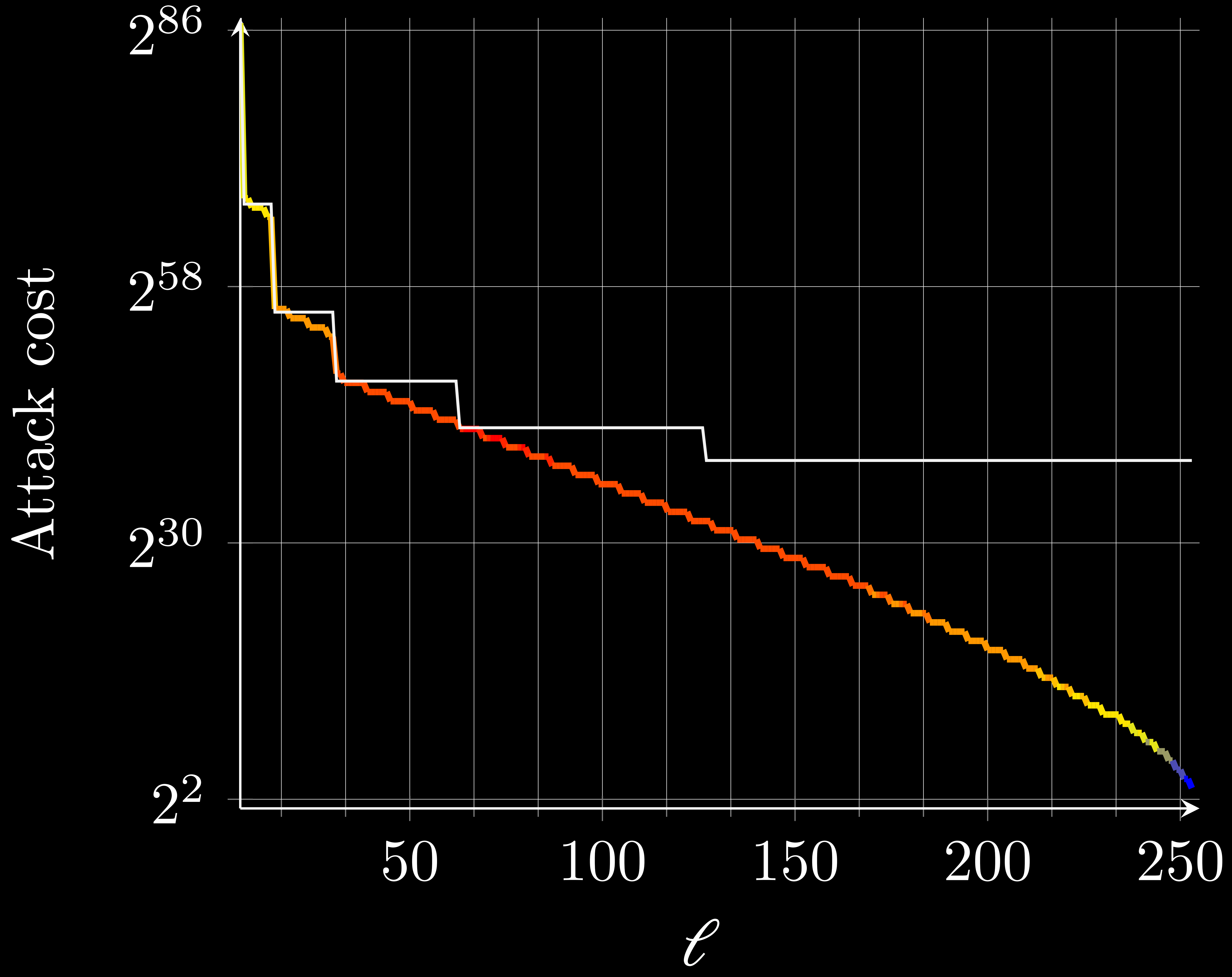
Attack summary - generalised

- Pick different basis $\overrightarrow{\rho}_0^{(0)}, \ldots, \overrightarrow{\rho}_{\ell-1}^{(0)}$ and $\overrightarrow{\rho}_0^{(1)}, \ldots, \overrightarrow{\rho}_{\ell-1}^{(1)}$
- Hash them to obtain $c_j^{(b)}$.
- Interpolate and compute $\rho(x_0, \ldots, x_{\ell-1})$
- Decompose $\mathrm{H}((\rho_0, \ldots, \rho_{\ell-1})) + \rho_\ell = (b_0, \ldots, b_{\ell-4}, b_{\ell-3}, b_{\ell-2}, b_{\ell-1})_2$.

Attack summary - generalised

- Pick different basis $\overrightarrow{\rho}_0^{(0)}, \ldots, \overrightarrow{\rho}_{\ell-1}^{(0)}$ and $\overrightarrow{\rho}_0^{(1)}, \ldots, \overrightarrow{\rho}_{\ell-1}^{(1)}$
- Hash them to obtain $c_j^{(b)}$.
- Interpolate and compute $\rho(x_0, \ldots, x_{\ell-1})$
- Decompose H$((\rho_0, \ldots, \rho_{\ell-1})) + \rho_\ell$ = $(b_0, \ldots, b_{\ell-4}, b_{\ell-3}, 0, 0)_2$.

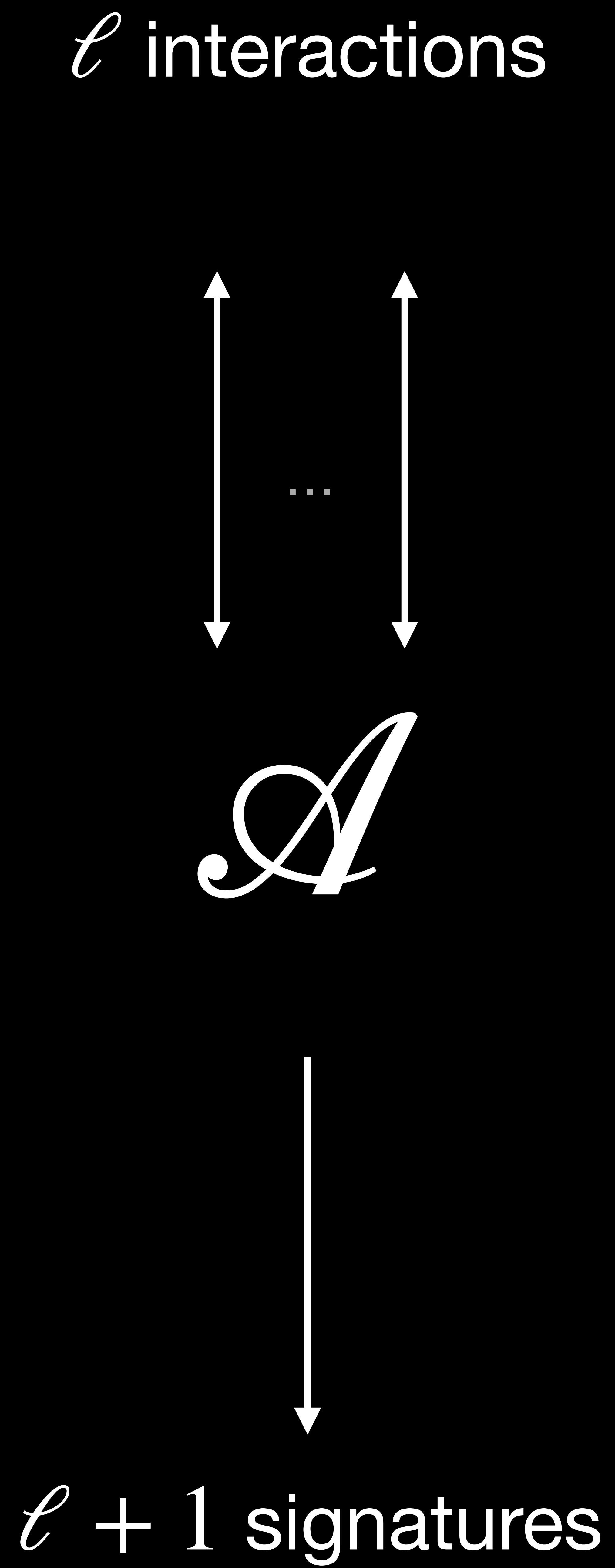
Complexity



Attacking
Blind signatures

Blind signatures

Unforgeability



Blind signatures

Blindness

[not covered]

Schnorr blind signature

Signer $x \in \mathbb{F}_p$

$(\mathbb{G}, p, G)$

User $m \in \{0,1\}^*$

$k \leftarrow \mathbb{Z}_p$

$K := kG$

$$\xrightarrow{\quad K \quad}$$

$\alpha, \beta \leftarrow \mathbb{Z}_p$

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

$c' := \mathsf{H}(K', m)$

$c := c' - \beta$

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

$r := cx + k$

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

$r' := r + \alpha$

Check: (K, c, r)

Final signature: (K', r')

Schnorr blind signature

Signer $x \in \mathbb{F}_p$

$(\mathbb{G}, p, G)$

User $m \in \{0,1\}^*$

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

$$K := kG$$

$$\xrightarrow{K}$$

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

$$\xleftarrow{c}$$

$$r := cx + k$$

$$\xrightarrow{r}$$

Verification:

$$rG = cX + K$$

$$c = \mathsf{H}(K, m)$$

Schnorr blind signatures

One-more unforgeability

$K = \text{Sign}_0()$

$\longrightarrow$

$r = \text{Sign}_1(c)$

$\longleftarrow$

$\mathcal{A}$

(ℓ interactions)

Attack on Schnorr blind signatures
[Schnorr01]

Signer $\xrightarrow{K_0, \ldots, K_{\ell-1}}$ $\mathscr{A}$

Signer $\xleftarrow{c_0, \ldots, c_{\ell-1}}$ $\mathscr{A}$

Signer $\xrightarrow{r_0, \ldots, r_{\ell-1}}$ $\mathscr{A}$

$\mathscr{A} \longleftarrow (\overrightarrow{\rho}_i)_{i=0}^{\ell}, \overrightarrow{c}$

$\forall 0 < i \leq \ell\colon \quad \langle \overrightarrow{\rho}_i, \overrightarrow{c} \rangle = \mathsf{H}(\overrightarrow{\rho}_i) = \tilde{\mathsf{H}}(\sum_j^{\ell} \rho_{i,j} K_j, m_i)$

$\mathscr{A} \xrightarrow{\ell + 1 \text{ forgeries}}$

$$K_i^\star = \sum_{j=0}^{\ell-1} \rho_{i,j} \cdot K_j \qquad c_i^\star = \sum_{j=0}^{\ell-1} \rho_{i,j} \cdot c_j \qquad r_i^\star = \sum_{j=0}^{\ell-1} \rho_{i,j} \cdot r_j$$

$$r_i^\star G = \sum_j \rho_{i,j} \cdot r_j G = \sum_j \rho_{i,j} \cdot (c_j X + K_j) = c_i^\star X + K_i^\star$$

How practical?

55LOC.

```
# public parameters: secp256k1
Zq = GF(0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffefffffc2f)
E = EllipticCurve(Zq, [0, 7])
G = E.lift_x(0x79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798)
p = G.order()
Zp = GF(p)

def random_oracle(hinput, _table=dict()):
    if hinput not in _table:
        _table[hinput] = Zp.random_element()
    return _table[hinput]

def verify(message, K, e, s):
    assert random_oracle((K, message)) == e, "random oracle fails"
    assert G * int(s) + X * int(e) == K, "verification equation fails"
    return True

def inner_product(coefficients, values):
    return sum(y*int(x) for x, y in zip(coefficients, values))

# server: generate public key
x = Zp.random_element()
X = G * int(x)

# adversary: open `ell` sessions
ell = 256

# server: generate commitments
k = [Zp.random_element() for i in range(ell)]
K = [G * int(k_i) for k_i in k]

# adversary: generate challenges
e = [[random_oracle((K_i, b)) for b in range(2)] for K_i in K]
P = ([-sum([Zp(2)^i * e[i][0]/(e[i][1] - e[i][0]) for i in range(ell)])] +
     [Zp(2)^i / (e[i][1] - e[i][0]) for i in range(ell)])

forged_K = inner_product(P, [G+X] + K)
forged_message = "message"
forged_e = random_oracle((forged_K, forged_message))
bits = [int(b) for b in bin(forged_e)[2:].rjust(256, '0')][::-1]
chosen_e = [e[i][b] for (i, b) in enumerate(bits)]

# server: generate the responses
s = [k[i] - chosen_e[i]*x for i in range(ell)]

# attacker: generate the forged response
forged_s = inner_product(P, [1] + s)

## check all previous signatures were valid
print(all(
    # l signatures generated honestly
    [verify(m_i, K_i, e_i, s_i) for (m_i, K_i, e_i, s_i) in zip(bits, K, chosen_e, s)] +
    # final signature
    [verify(forged_message, forged_K, forged_e, forged_s)]
))
```

Do not use blind Schnorr signatures.

Still in the game: blind RSA, Blind BLS, Abe blind signatures, clause Schnorr blind signatures.

Tessaro-Zhu blind signatures

Signer $x \in \mathbb{F}_p$

$(\mathbb{G}, p, G)$

User $m \in \{0,1\}^*$

$k \leftarrow \mathbb{Z}_p, \quad y \leftarrow \mathbb{Z}_p$

$K := kG$

$$\xrightarrow{\quad K \quad \mathsf{Com}(y) \quad}$$

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

$r := cyx + k$

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

check $\mathsf{Com}(y)$

Verification: $rG = cyX + K$

$c = \mathsf{H}(K, m, \mathsf{Com}(y))$