# On the unforgeability of the Chaum-Pedersen blind signature

Liliya Akhmetzyanova, Alexandra Babueva
CryptoPro LLC

CRYPTOPRO

CTCrypt'2024

# Outline

1. Motivation
2. Chaum-Pedersen blind signature
3. Analysis: strong unforgeability
4. Analysis: weak unforgeability

# Outline

1. Motivation
2. Chaum-Pedersen blind signature
3. Analysis: strong unforgeability
4. Analysis: weak unforgeability

# Perspective blind signatures for standardization

ElGamal based schemes

Tessaro-Zhu scheme

Abe scheme

Chaum-Pedersen scheme

Schnorr scheme

Okamoto-Schnorr scheme

# Perspective blind signatures for standardization

ElGamal based schemes

Tessaro-Zhu scheme


Abe scheme

Chaum-Pedersen scheme

Schnorr scheme

Okamoto-Schnorr scheme

Benhamouda F. et al «On the (in)security of ROS», 2021

# Perspective blind signatures for standardization

Tessaro-Zhu scheme

Chaum-Pedersen scheme

Abe scheme

Tessaro S., Zhu C. «Short Pairing-Free Blind Signatures with Exponential Security», 2022

# Open problem

Tessaro-Zhu scheme

Chaum-Pedersen scheme

used in U-Prove!

Abe scheme

How secure is Chaum-Pedersen blind signature scheme?

# Outline

1. Motivation
2. Chaum-Pedersen blind signature
3. Analysis: strong unforgeability
4. Analysis: weak unforgeability

# Blind signatures

- $(sk, pk) \leftarrow KeyGen(\,)$: key generation algorithm

- $(b, \sigma) \leftarrow \langle Signer(sk), User(pk, m) \rangle$: interactive signing protocol that is run between a Signer and a User

- $b \leftarrow Verify(pk, m, \sigma)$: verification algorithm

security notions

unforgeability

blindness

# Chaum-Pedersen scheme

Original description is given for multiplicative group of finite field

Chaum D., Pedersen T. P. «Wallet databases with observers», 1992

Base blocks:

- elliptic curve $\mathcal{E}$ of prime order $q$ with base point $P$
- hash function $H\colon \{0,1\}^* \to \mathbb{Z}_q^*$
- hash function $\mathcal{H}\colon \{0,1\}^* \to \mathcal{E}$

  hash-to-curve constructions: RFC 9380 «Hashing to elliptic curves»

# Chaum-Pedersen scheme

Original description is given for multiplicative group of finite field

Chaum D., Pedersen T. P. «Wallet databases with observers», 1992

Base blocks:

- elliptic curve $\mathcal{E}$ of prime order $q$ with base point $P$
- hash function $H\colon \{0,1\}^* \to \mathbb{Z}_q^*$
- hash function $\mathcal{H}\colon \{0,1\}^* \to \mathcal{E}$
  hash-to-curve constructions: RFC 9380 «Hashing to elliptic curves»

Key generation algorithm:

$$\underline{\text{KeyGen}(\,)}$$

$$d \leftarrow_\$ \mathbb{Z}_q$$

$$Q \leftarrow dP$$

$$\text{return } (d, Q)$$

# Chaum-Pedersen signature

Sign $(d, m)$

$M \leftarrow \mathcal{H}(m)$

$\xleftarrow{\quad M \quad}$

$Z \leftarrow dM$
$k \leftarrow_{\$} \mathbb{Z}_q$
$A \leftarrow kP, B \leftarrow kM$

$\xrightarrow{\quad Z, A, B \quad}$

$c \leftarrow H(M \parallel Z \parallel A \parallel B)$

$\xleftarrow{\quad c \quad}$

$s \leftarrow k + cd$

$\xrightarrow{\quad s \quad}$

$\sigma \leftarrow (s, c, Z)$
return $\sigma$

# Chaum-Pedersen signature

Sign $(d, m)$

$M \leftarrow \mathcal{H}(m)$ hash-to-curve

$\xleftarrow{M}$

$Z \leftarrow dM$
$k \leftarrow_{\$} \mathbb{Z}_q$
$A \leftarrow kP, B \leftarrow kM$

$\xrightarrow{Z, A, B}$

$c \leftarrow H(M \parallel Z \parallel A \parallel B)$

$\xleftarrow{c}$

$s \leftarrow k + cd$

$\xrightarrow{s}$

$\sigma \leftarrow (s, c, Z)$
return $\sigma$

# Chaum-Pedersen signature

Sign $(d, m)$

$M \leftarrow \mathcal{H}(m)$

$M$ ←

$Z \leftarrow dM$

$k \leftarrow_{\$} \mathbb{Z}_q$

$A \leftarrow kP, B \leftarrow kM$

→ $Z, A, B$

$c \leftarrow H(M \parallel Z \parallel A \parallel B)$

$c$ ←

$s \leftarrow k + cd$

→ $s$

$\sigma \leftarrow (s, c, Z)$

return $\sigma$

proving DLog equality: $\log_P Q = \log_M Z$, provides unforgeability

# Chaum-Pedersen signature

Sign $(d, m)$

$M \leftarrow \mathcal{H}(m)$

$\xleftarrow{\quad M \quad}$

$Z \leftarrow dM$
$k \leftarrow_{\$} \mathbb{Z}_q$
$A \leftarrow kP, B \leftarrow kM$

$\xrightarrow{\quad Z, A, B \quad}$

$c \leftarrow H(M \parallel Z \parallel A \parallel B)$

$\xleftarrow{\quad c \quad}$

$s \leftarrow k + cd$

$\xrightarrow{\quad s \quad}$

$\sigma \leftarrow (s, c, Z)$
return $\sigma$

Verify $(Q, m, \sigma)$: $c \stackrel{?}{=} H\big(\mathcal{H}(m) \parallel Z \parallel (sP - cQ) \parallel (s\mathcal{H}(m) - cZ)\big)$

# Chaum-Pedersen blind signature

| | | |
|---|---|---|
| Signer $(d)$ | | User $(Q, m)$ |
| | | $M \leftarrow \mathcal{H}(m)$ |
| | $\xleftarrow{\quad M \quad}$ | |
| $Z \leftarrow dM$ <br> $k \leftarrow_{\$} \mathbb{Z}_q$ <br> $A \leftarrow kP, B \leftarrow kM$ | | |
| | $\xrightarrow{\quad Z, A, B \quad}$ | |
| | | $c \leftarrow H(M \parallel Z \parallel A \parallel B)$ |
| | $\xleftarrow{\quad c \quad}$ | |
| $s \leftarrow k + cd$ | | |
| | $\xrightarrow{\quad s \quad}$ | |
| | | $\sigma \leftarrow (s, c, Z)$ |
| return $1$ | | return $\sigma$ |

# Chaum-Pedersen blind signature

| Signer ($d$) | | User ($Q, m$) |
|---|---|---|
| | | $M' \leftarrow \mathcal{H}(m)$ |
| | $\xleftarrow{M}$ | $\alpha \leftarrow_\$ \mathbb{Z}_q, M \leftarrow \alpha^{-1}M'$ |
| $Z \leftarrow dM$ | | |
| $k \leftarrow_\$ \mathbb{Z}_q$ | | |
| $A \leftarrow kP, B \leftarrow kM$ | $\xrightarrow{Z, A, B}$ | |
| | | $Z' \leftarrow \alpha Z$ |
| | | $u, v \leftarrow_\$ \mathbb{Z}_q$ |
| | | $A' \leftarrow uA + vP,$ |
| | | $B' \leftarrow u\alpha B + vM'$ |
| | | $c' \leftarrow H(M' \parallel Z' \parallel A' \parallel B')$ |
| | $\xleftarrow{c}$ | $c \leftarrow c'u^{-1}$ |
| $s \leftarrow k + cd$ | $\xrightarrow{s}$ | |
| | | $s' \leftarrow us + v$ |
| | | $\sigma \leftarrow (s', c', Z')$ |
| return $1$ | | return $\sigma$ |

for blindness!

# Unforgeability property

*Attack:* adversary can act as a User and open parallel sessions of the Signing protocol

*Threat (one-more forgery)*: adversary generates $(\ell + 1)$ valid (message, signature) pairs after $\ell$ successful interactions with the Signer

strong unforgeability

all (message, signature) pairs are distinct

weak unforgeability

all messages are distinct

# Outline

1. Motivation
2. Chaum-Pedersen blind signature
3. Analysis: strong unforgeability
4. Analysis: weak unforgeability

# ROS-style attack

Chaum-Pedersen scheme does not provide strong unforgeability when the number of parallel sessions $\ell \geq \lceil \log q \rceil$.

Signer

(breaks one-more unforgeability)

$$\xleftarrow{M, \ldots, M}$$

$$\xrightarrow{(Z, A_1, B_1), \ldots, (Z, A_\ell, B_\ell)}$$

$\ell$ successful sessions

$$\xleftarrow{c_1, \ldots, c_\ell}$$

$$\xrightarrow{s_1, \ldots, s_\ell}$$

$(\ell$+1) valid signatures

$$(m, \sigma_1), \ldots, (m, \sigma_{\ell+1})$$

# ROS-style attack: distinct messages

Let $M_{\ell+1} = \mathcal{H}(m_{\ell+1})$ for some new message $m_{\ell+1}$

known $\log_P M_{\ell+1}$

unknown $\log_P M_{\ell+1}$

ROS attack works

ROS attack does not work

# ROS-style attack: distinct messages

Let $M_{\ell+1} = \mathcal{H}(m_{\ell+1})$ for some new message $m_{\ell+1}$

known $\log_P M_{\ell+1}$

unknown $\log_P M_{\ell+1}$

ROS attack works

ROS attack does not work

still hope....

# Outline

1. Motivation
2. Chaum-Pedersen blind signature
3. Analysis: strong unforgeability
4. Analysis: weak unforgeability

# Weak unforgeability: assumptions

Necessary conditions for security:

1) for $\mathcal{H}, \mathcal{E}$:

   for given $m$ it should be hard to find $\alpha$: $\mathcal{H}(m) = \alpha P$

# Weak unforgeability: assumptions

Necessary conditions for security:

2) for $H$:

for given $M', Z', s'$ it should be hard to find $c'$:

$$c' = H(M' \parallel Z' \parallel (s'P - c'Q) \parallel (s'M' - c'Z'))$$

# Weak unforgeability: assumptions

Necessary conditions for security:

3) Discrete Logarithm problem?

# Weak unforgeability: assumptions

Necessary conditions for security:

3) Discrete Logarithm problem?

Signer

$$\xleftarrow{\quad M \quad}$$

$$\xrightarrow{\quad Z = dM \quad}$$

# Weak unforgeability: assumptions

Necessary conditions for security:

3) Discrete Logarithm problem?

Signer

$$M_1 = Q$$

$$Z_1 = dM_1 = d^2 P$$

$$M_2 = Z_1$$

$$Z_2 = dM_2 = d^3 P$$

$$dP, d^2 P, \dots, d^{\ell+1} P$$

$$M_\ell = Z_{\ell-1}$$

$$Z_\ell = dM_\ell = d^{\ell+1} P$$

# Weak unforgeability: assumptions

Necessary conditions for security:

3) Strong Discrete Logarithm problem (SDL)

$$dP, d^2P, \ldots, d^sP \longrightarrow d$$

# Weak unforgeability: assumptions

Necessary conditions for security:

3) Strong Discrete Logarithm problem (SDL)

$$\boxed{dP, d^2P, \dots, d^sP \longrightarrow d}$$

Best known method: Cheon J. H., 2006
"Security analysis of the strong Diffie-Hellman problem"

$$T \approx \log q \cdot \left( \sqrt{\frac{q}{s}} + \sqrt{s} \right)$$ for $s$ that divide $(q-1)$

# Weak unforgeability: assumptions

Necessary conditions for security:

3) Strong Discrete Logarithm problem (SDL)

$$dP, d^2P, \dots, d^sP \longrightarrow d$$

| Curve | $\log q$ | $s_m$ | $T$ |
|---|---|---|---|
| id-tc26-gost-3410-2012-256-paramSetB | 256 | $\approx 2^{32}$ | $2^{120}$ |
| id-tc26-gost-3410-2012-256-paramSetC | 256 | $\approx 2^{62}$ | $2^{105}$ |
| id-tc26-gost-3410-2012-256-paramSetD | 256 | $\approx 2^{64}$ | $2^{104}$ |
| id-tc26-gost-3410-12-512-paramSetA | 512 | $\approx 2^{25}$ | $2^{252}$ |
| id-tc26-gost-3410-12-512-paramSetA | 512 | $\approx 2^{11}$ | $2^{259}$ |

$s_m$ – maximal divisor of $(q-1)$ such that $s_m \leq 2^{64}+1$

# Weak unforgeability: security bound

Restrictions on the set of adversaries:

- $\mathcal{H}$ is a random oracle
- $H$ is a random oracle
- Algebraic Group Model (AGM)

# Weak unforgeability: security bound

Restrictions on the set of adversaries:

- $\mathcal{H}$ is a random oracle
- $H$ is a random oracle
- Algebraic Group Model (AGM)

For each group element returned by the adversary, the adversary should provide the coefficients of decomposition of this element into a linear combination of all the received elements.

$\xrightarrow{X_1, \dots, X_n}$

$\xrightarrow{Z, (z_1, \dots, z_n)}$

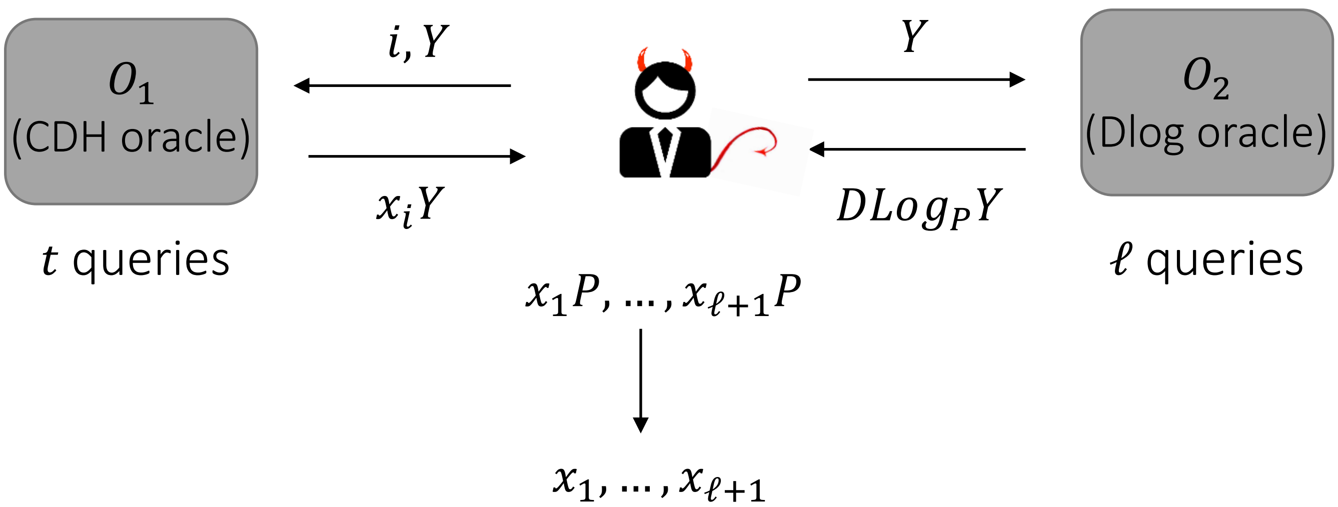
$$Z = \sum_{i=1}^{n} z_i X_i$$

# Weak unforgeability: security bound

Sufficient conditions for security:

1) Strong One-More Discrete Logarithm problem (SOMDL)

Parameters: $t, \ell$

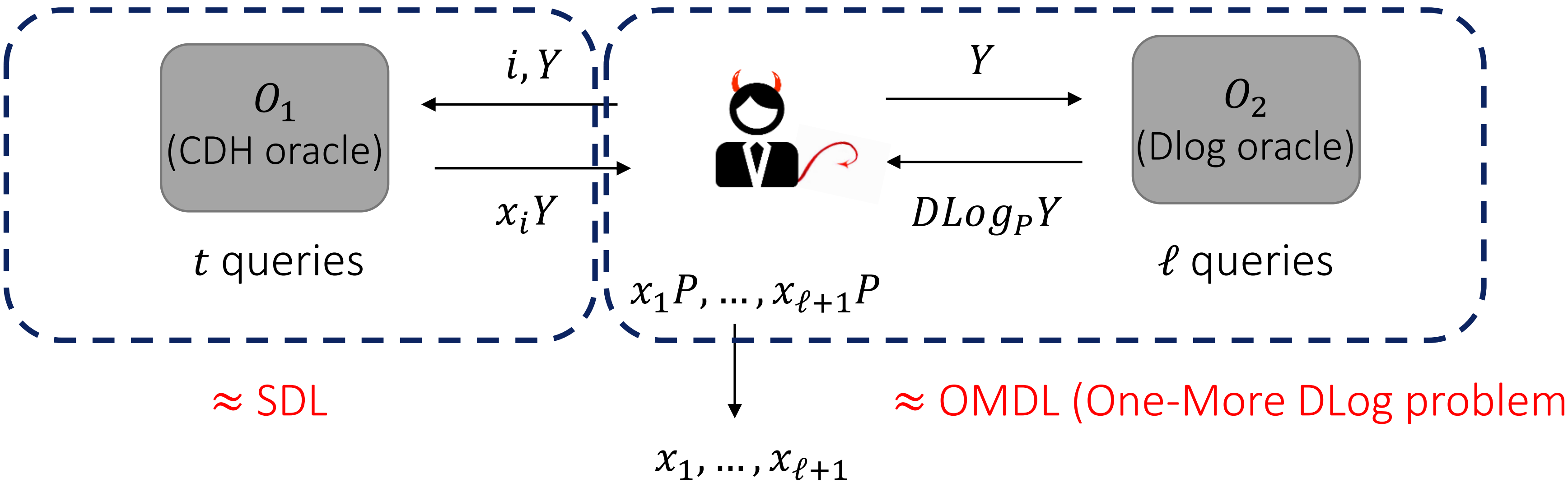


# Weak unforgeability: security bound

Sufficient conditions for security:

1) Strong One-More Discrete Logarithm problem (SOMDL)

Parameters: $t, \ell$



# Weak unforgeability: security bound

Sufficient conditions for security:

2) Representation problem (REPR)

Parameters: $s$

$$x_1P, \dots, x_sP \longrightarrow \alpha_1, \dots, \alpha_s, \beta\colon\ \alpha_1 x_1 + \cdots + \alpha_s x_s + \beta = 0$$

Best known method: solving DLog problem or finding the collision between input points

# Weak unforgeability: security bound

$$Adv^{wUF}_{CP-BS}(\mathcal{A}) \leq 2 \cdot Adv^{SOMDL}_{G,2t,t}(\mathcal{B}) + Adv^{REPR}_{G,q_1+\ell+1}(\mathcal{C}) + \frac{2(\ell+1)+q_2}{q},$$

where

- $q_1$ – number of queries to random oracle $\mathcal{H}$
- $q_2$ – number of queries to random oracle $H$
- $t$ – number of open sessions
- $\ell$ – number of closed sessions

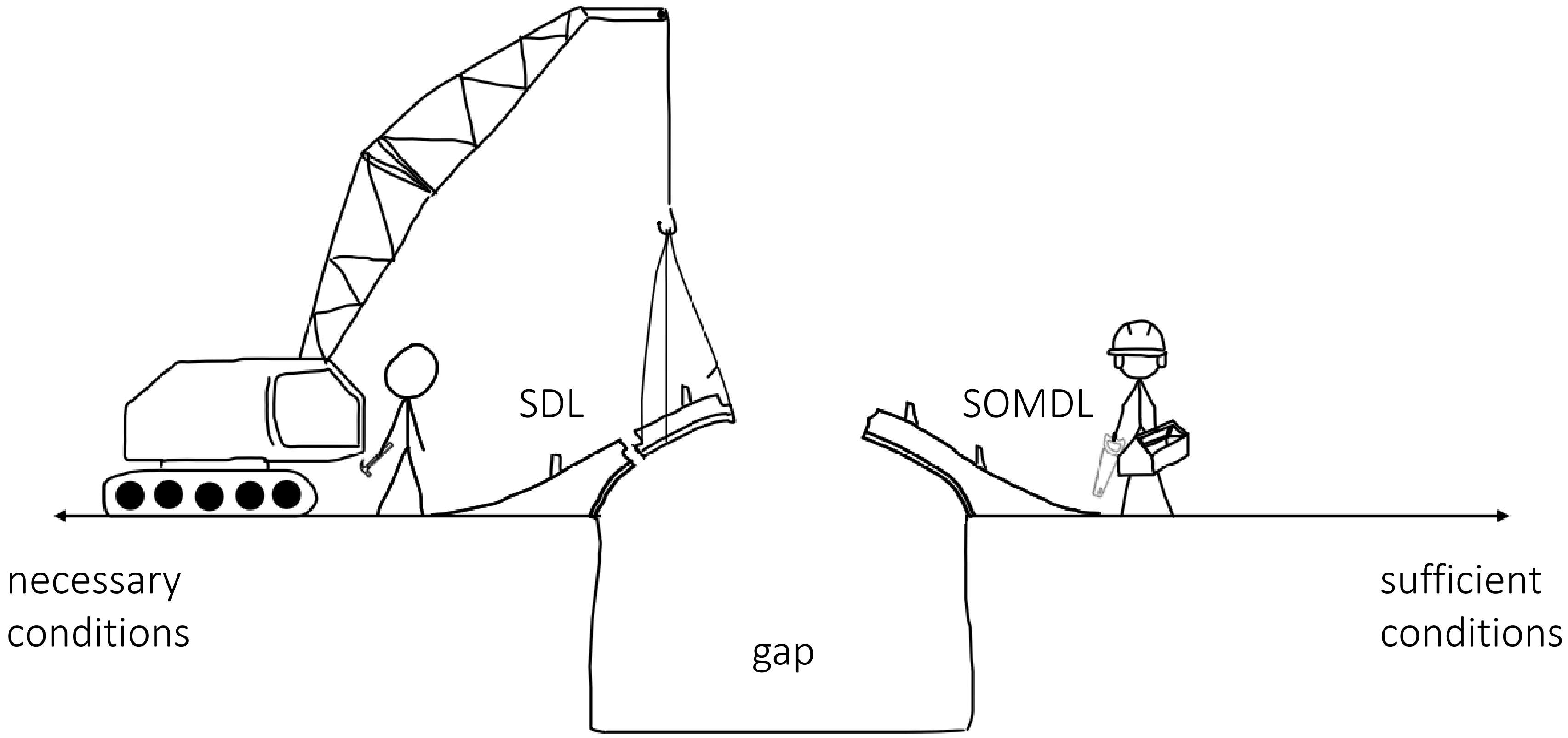
# Weak unforgeability: summary

| Necessary conditions | Sufficient conditions (in AGM) |
| --- | --- |
| 1) for $\mathcal{H}$, $\mathcal{E}$:<br><br>for given $m$ it should be hard to find $\alpha$: $\mathcal{H}(m) = \alpha P$ | 1) hard REPR problem<br>(under assumption that $\mathcal{H}$ is RO) |
| 2) for $H$:<br><br>for given $M'$, $Z'$, $s'$ it should be hard to find $c'$:<br>$c' = H(M' \| Z' \| (s'P - c'Q) \| (s'M' - c'Z'))$ | 2) sufficiently big $q$<br>(under assumption that $H$ is RO) |
| 3) hard SDL problem | 3) hard SOMDL problem |

# Conclusion

- Chaum-Pedersen scheme does not provide strong unforgeability
- Necessary condition for weak unforgeability – SDL problem that is not harder than DLog
- Need hash-to-curve construction

# Future work




*The picture is taken from: NIST Crypto Reading Club, M. Backendal & M. Haller, Thriving in Between Theory and Practice: How Applied Cryptography Bridges the Gap

Thank you for your attention!

Questions?

babueva@cryptopro.ru
lah@cryptopro.ru