CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

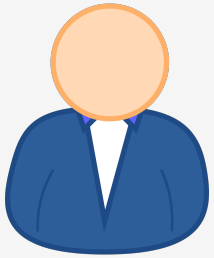
Non-interactive Blind Signatures for Random Messages

Lucjan Hanzlik

Eurocrypt 2023

Two-move Blind Signatures

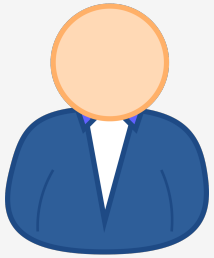
User/Recipient



Signer

Two-move Blind Signatures

User/Recipient

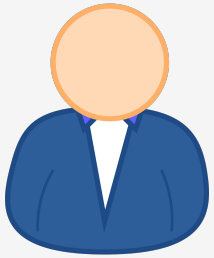


(req, St) ← Request(m, pk)

Signer

Two-move Blind Signatures

User/Recipient



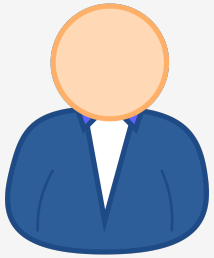
Signer

(req, St) ← Request(m, pk)

req

Two-move Blind Signatures

User/Recipient



(req, St) ← Request(m, pk)

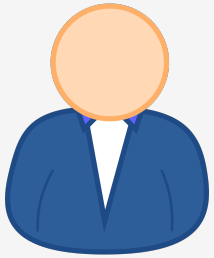
req

Signer

pre ← Issue(req, sk)

Two-move Blind Signatures

User/Recipient



Signer

(req, St) ← Request(m, pk)

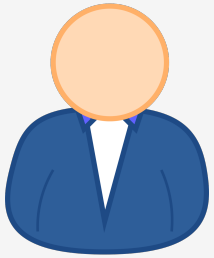
req

pre ← Issue(req, sk)

pre

Two-move Blind Signatures

User/Recipient



Signer

(req, St) ← Request(m, pk)

req

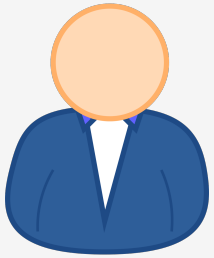
pre ← Issue(req, sk)

pre

sig ← Obtain(pre, St, pk)

Two-move Blind Signatures

User/Recipient



Signer

Unforgeability

(req, St) ← Request(m, pk)

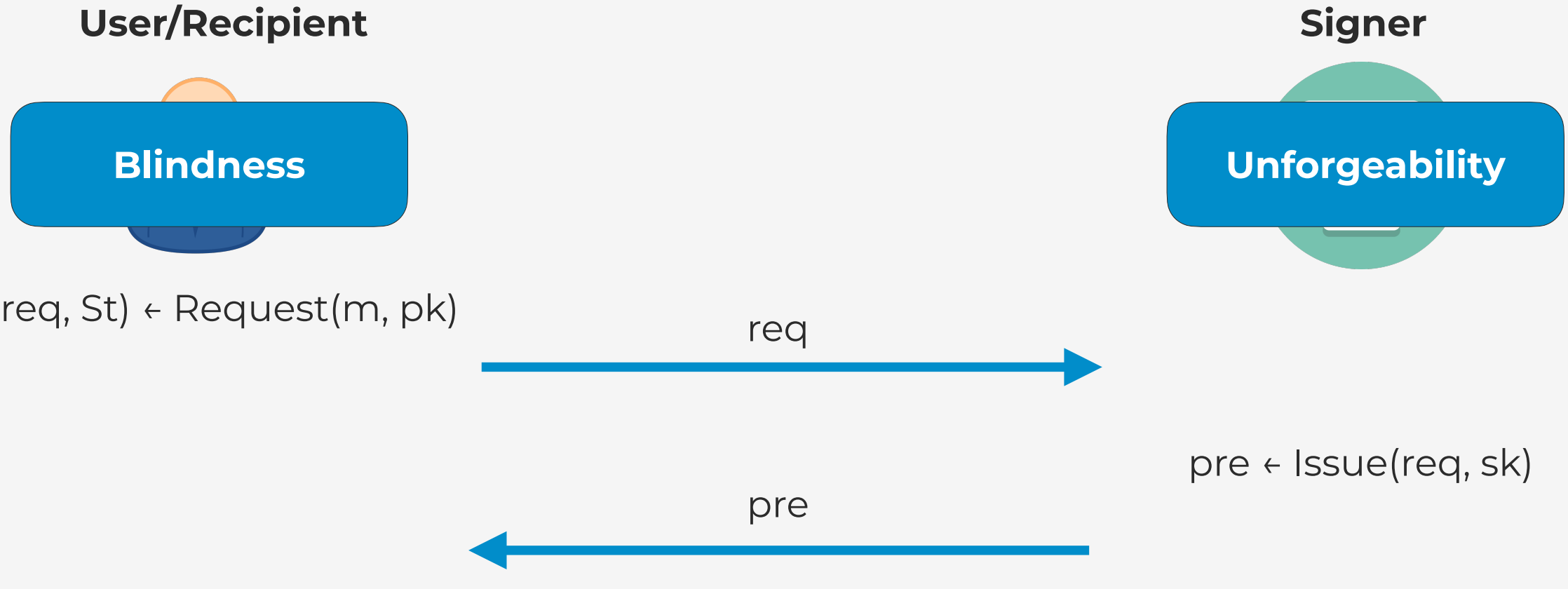
req

pre ← Issue(req, sk)

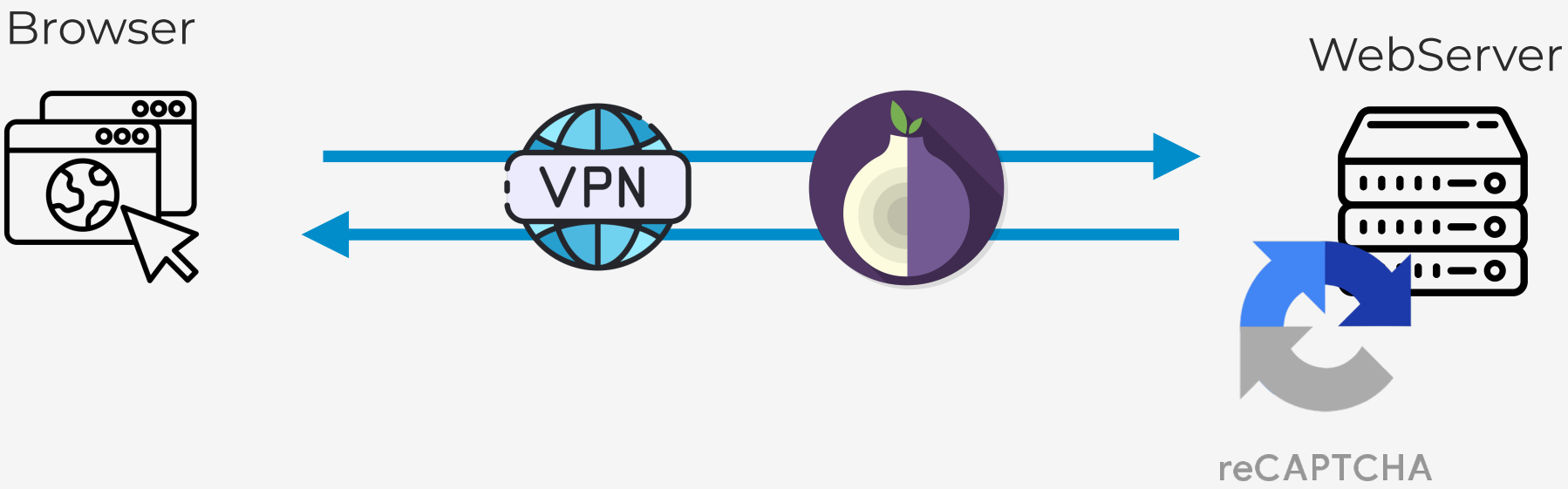
pre

sig ← Obtain(pre, St, pk)

Two-move Blind Signatures

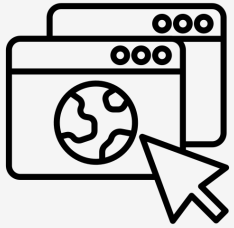


Privacy Pass



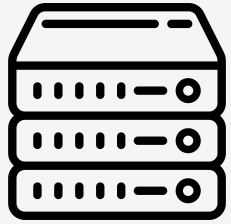
Privacy Pass

Browser



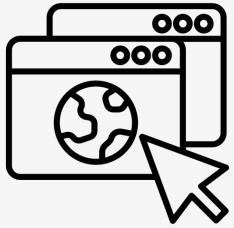
Issuer

WebServer



Privacy Pass

Browser



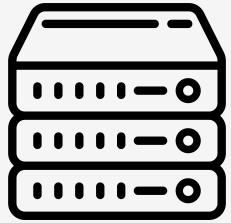
(req, St) ← Request(m, pk)

pre ← Issue(req, sk)

Issuer

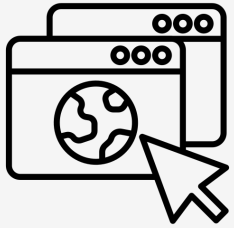
sig ← Obtain(pre, St, pk)

WebServer



Privacy Pass

Browser



(req, St) ← Request(m, pk)

pre ← Issue(req, sk)

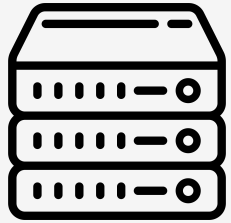
Issuer

sig ← Obtain(pre, St, pk)

(m, sig)

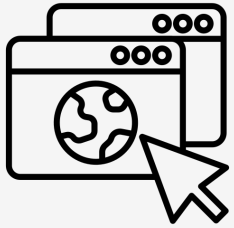
(m, sig)

WebServer



Privacy Pass

Browser



(req, St) ← Request(m, pk)

pre ← Issue(req, sk)

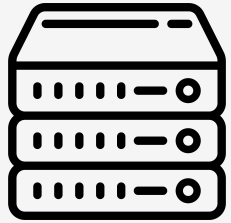
Issuer

sig ← Obtain(pre, St, pk)

(m, sig)

(m, sig)

WebServer

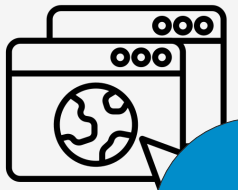


Verify(m, sig, pk)

Store m

Privacy Pass

Browser



(req, St) ← Request(m, pk)

Issuer

pre ← Issue(req, sk)

sig ← Obtain(pr

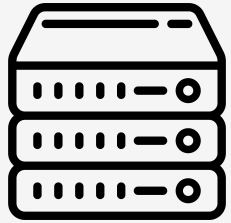
(m

Message is a random strings.

Can we use this?

(m, sig)

WebServer

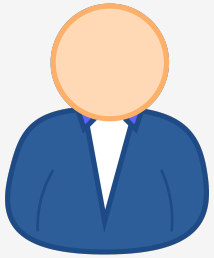


Verify(m, sig, pk)

Store m

Blind Signatures for Random Messages

User/Recipient



Signer

(req, St) ← Request(~~m~~, pk)

req

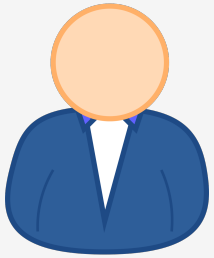
pre ← Issue(req, sk)

pre

(m, sig) ← Obtain(pre, St, pk)

Blind Signatures for Random Messages

User/Recipient



Signer

(req, St) ← Request

Not really interesting.

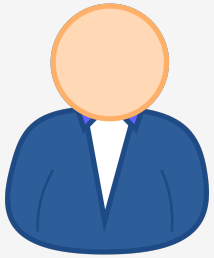
Efficient two-move BS exist and provide more features.

e ← Issue(req, sk)

(m, sig) ← Obtain(pre, St, pk)

Non-interactive Blind Signatures for Random Messages

User/Recipient (skr, pkr)



Signer

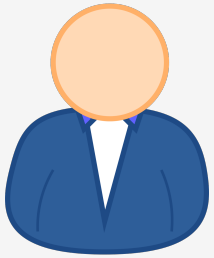
pre ← Issue(sk, pkr, nonce)

pre, nonce

(m, sig) ← Obtain(skr, pk, pre)

Non-interactive Blind Signatures for Random Messages

User/Recipient (skr, pkr)



Signer

pre ← Issue(sk, pkr, nonce)

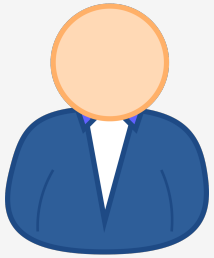
pre, nonce

(m, sig) ← Obtain(skr, pk, pre)

m = f(skr,nonce)

NIBS without PKI

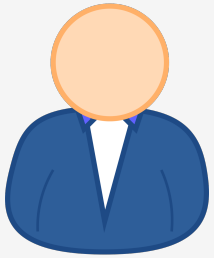
User/Recipient



Signer

NIBS without PKI

User/Recipient

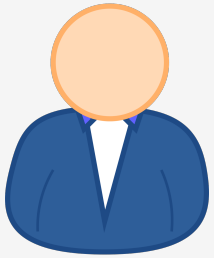


Signer

(skr,pkr) ← RKeyGen(secpar)

NIBS without PKI

User/Recipient



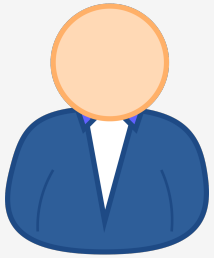
Signer

(skr,pkr) ← RKeyGen(secpar)

pkr

NIBS without PKI

User/Recipient



Signer

(skr,pkr) ← RKeyGen(secpar)

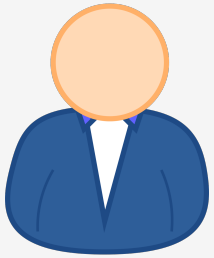
pkr

pre, nonce

pre ← Issue(sk, pkr, nonce)

NIBS without PKI

User/Recipient



Signer

(skr,pkr) ← RKeyGen(secpar)

pkr

pre, nonce

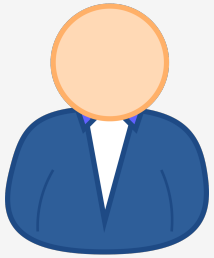
pre ← Issue(sk, pkr, nonce)

pre_2, $nonce_2$

pre_2 ← Issue(sk, pkr, $nonce_2$)

NIBS without PKI

User/Recipient



Signer

(skr,pkr) ← RKeyGen(

Can be used to batch and post issue tokens/e-cash.

pre, nonce

pre ← Issue(sk, pkr, nonce)

pre_2, $nonce_2$

pre_2 ← Issue(sk, pkr, $nonce_2$)

Unforgeability for NIBS

Experiment

Adversary can make up to k queries to a presignature oracle.

Oracle outputs presignatures given nonce and recipients public key.

Experiment

Winning conditions

Adversary outputs at least k+1 valid message-signature pairs.

Messages are distinct.

Winning conditions

Blindness for NIBS

Recipient Blindness

Signatures obtained by different recipient are unlinkable.

Preserves the privacy across recipients.

Recipient Blindness

Nonce Blindness

Signatures for the same recipient are unlinkable.

Allows to issue multiple presignatures without breaking blindness.

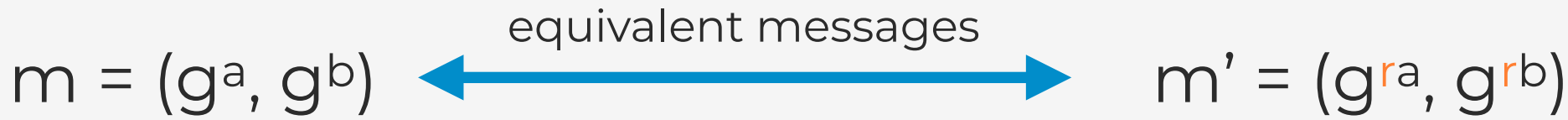
Nonce Blindness

Building Blocks

Signatures on Equivalence Classes

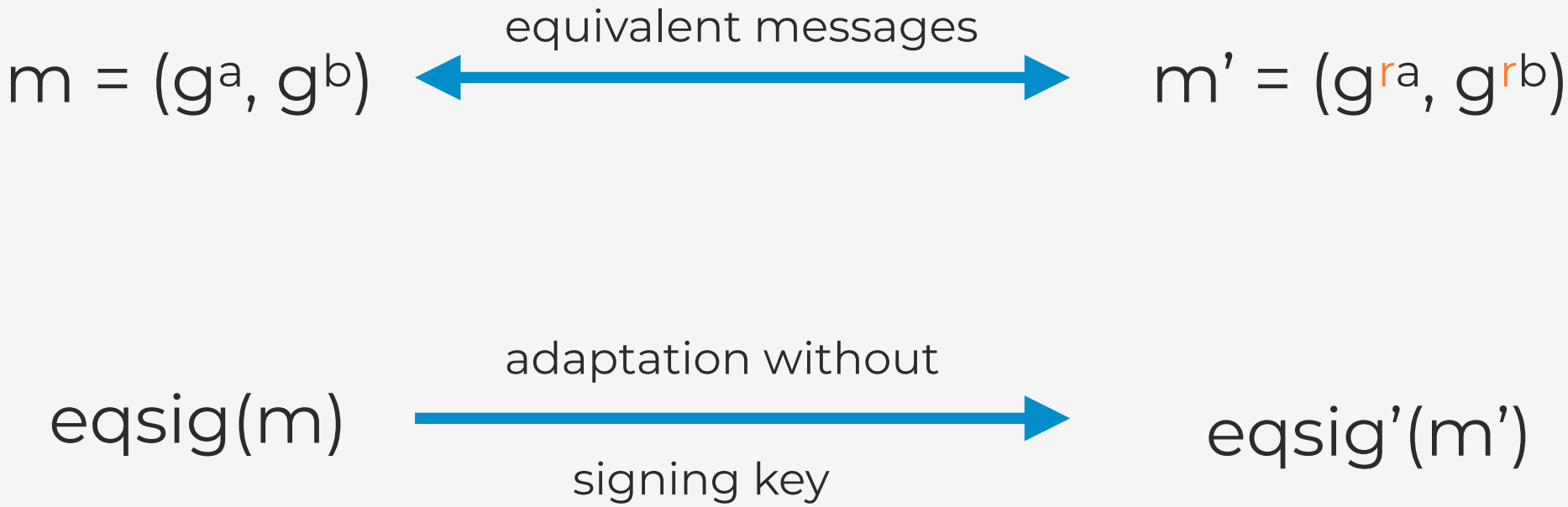
Building Blocks

Signatures on Equivalence Classes



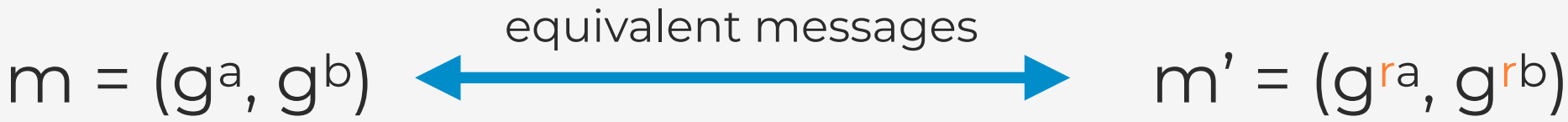
Building Blocks

Signatures on Equivalence Classes



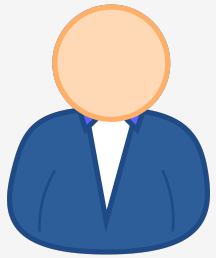
Building Blocks

Signatures on Equivalence Classes

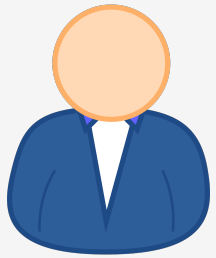


Random signature even if signer malicious

How to efficiently construct NIBS?

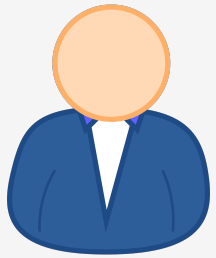


How to efficiently construct NIBS?



pre := eqsig(pkr, H(nonce))

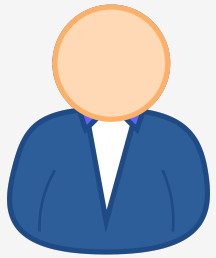
How to efficiently construct NIBS?



$m := H(nonce)^{skr^{-1}}$

pre := eqsig(pkr, H(nonce))

How to efficiently construct NIBS?

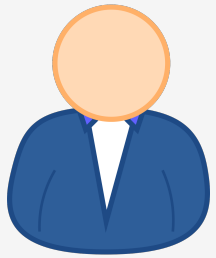


$m := H(nonce)^{skr^{-1}}$

$pre := eqsig(\ pkr, H(nonce)\ )$

$sig := adapt(\ pre, skr^{-1}\ ) = eqsig(\ (g,\ H(nonce)^{skr^{-1}}\ ))$

How to efficiently construct NIBS?



$m := H(nonce)^{skr^{-1}}$

$pre := eqsig(\ pkr, H(nonce)\ )$

$sig := adapt(\ pre, skr^{-1}\ ) = eqsig(\ (g,\ H(nonce)^{skr^{-1}}\ ))$

Recipients key is a standard DH key!

Security

- Unforgeability from signatures on equivalence signatures
- $H(\text{nonce})^{skr^{-1}}$ is a PRF for the recipient's key
- Blindness from inverse/strong DDH

Summary and Open Problems

- NIBS and tag-based NIBS definitions
- Efficient constructions that works with standard PKI keys
- Generic construction from VRF and NIZK

Can we construct:

- Post-quantum NIBS/TNIBS
- NIBS/TNIBS without ROM
- NIBS/TNIBS without pairings

Contact

E-mail

hanzlik@cispa.de

Eprint

eprint.iacr.org/2023/388

Unforgeability for NIBS

Adversary

Challenger

Unforgeability for NIBS

Adversary

pkr_1, $nonce_1$

pre_1

Challenger

Unforgeability for NIBS

Adversary

Challenger

pkr_1, $nonce_1$

pre_1

• • •

pkr_k, $nonce_k$

pre_k

Unforgeability for NIBS

Adversary

Challenger

pkr_1, $nonce_1$

pre_1

• • •

pkr_k, $nonce_k$

pre_k

(m_1,sig_1), ... , (m_l,sig_l)

Unforgeability for NIBS

Adversary

Challenger

pkr_1, $nonce_1$

pre_1

• • •

pkr_k, $nonce_k$

pre_k

1) valid signatures
2) distinct messages
3) queries k < l

(m_1,sig_1), ... , (m_l,sig_l)

Recipient Blindness

Challenger

Adversary

Recipient Blindness

Challenger

pkr_0, pkr_1

Adversary

Recipient Blindness

Challenger

Adversary

pkr_0, pkr_1

$pre_0, nonce_0, pre_1, nonce_1, pk$

Recipient Blindness

Challenger

Adversary

pkr_0, pkr_1

$pre_0, nonce_0, pre_1, nonce_1, pk$

$(m_0, sig_0) \leftarrow Obtain(skr_0, pk, pre_0)$

$(m_1, sig_1) \leftarrow Obtain(skr_1, pk, pre_1)$

Recipient Blindness

Challenger

Adversary

pkr_0, pkr_1

$pre_0, nonce_0, pre_1, nonce_1, pk$

$(m_0, sig_0) \leftarrow Obtain(skr_0, pk, pre_0)$

$(m_1, sig_1) \leftarrow Obtain(skr_1, pk, pre_1)$

$(m_b, sig_b), (m_{1-b}, sig_{1-b})$

Recipient Blindness

Challenger

Adversary

pkr_0, pkr_1

$pre_0, nonce_0, pre_1, nonce_1, pk$

$(m_0, sig_0) \leftarrow Obtain(skr_0, pk, pre_0)$

$(m_1, sig_1) \leftarrow Obtain(skr_1, pk, pre_1)$

$(m_b, sig_b), (m_{1-b}, sig_{1-b})$

b’

Recipient Blindness

Challenger

Adversary

pkr_0, pkr_1

$pre_0, nonce_0, pre_1, nonce_1, pk$

$(m_0, sig_0) \leftarrow Obtain(skr_0, pk, pre_0)$

$(m_1, sig_1) \leftarrow Obtain(skr_1, pk, pre_1)$

Adversary wins if b' = b

Nonce Blindness

Challenger

Adversary

Nonce Blindness

Nonce Blindness

Challenger

Adversary

pkr

pre_0, $nonce_0$, pre_1, $nonce_1$, pk

Nonce Blindness

Challenger

Adversary

pkr →

← pre_0, $nonce_0$, pre_1, $nonce_1$, pk

$(m_0, sig_0) \leftarrow Obtain(skr, pk, pre_0)$

$(m_1, sig_1) \leftarrow Obtain(skr, pk, pre_1)$

Nonce Blindness

Challenger

Adversary

pkr

pre_0, $nonce_0$, pre_1, $nonce_1$, pk

$(m_0, sig_0) \leftarrow Obtain(skr, pk, pre_0)$

$(m_1, sig_1) \leftarrow Obtain(skr, pk, pre_1)$

$(m_b, sig_b), (m_{1-b}, sig_{1-b})$

Nonce Blindness

Challenger

Adversary

pkr

pre_0, $nonce_0$, pre_1, $nonce_1$, pk

$(m_0, sig_0) \leftarrow Obtain(skr, pk, pre_0)$

$(m_1, sig_1) \leftarrow Obtain(skr, pk, pre_1)$

$(m_b, sig_b), (m_{1-b}, sig_{1-b})$

b’

Nonce Blindness

Challenger

Adversary

pkr →

← pre_0, $nonce_0$, pre_1, $nonce_1$, pk

$(m_0, sig_0) \leftarrow Obtain(skr, pk, pre_0)$

$(m_1, sig_1) \leftarrow Obtain(skr, pk, pre_1)$

Adversary wins if b’ = b