Making Classical (Threshold) Signatures Post-Quantum for Single Use on a Public Ledger

Laurane Marco, Abdullah Talayhan, Serge Vaudenay

Speaker: Abdullah Talayhan (abdullah.talayhan@eplf.ch)
https://abdullahtalayhan.com

EPFL
LASEC

Background

Discrete Log Problem (DLP)

$$x \rightleftarrows g^x \mod p$$

(p is a prime with some structure)

One-way Hash Functions

$$x \rightleftarrows H(x)$$

Commitment Schemes

$\mathsf{com} \leftarrow \mathit{Commit}(x, \mathit{open})$

$\xrightarrow{\mathsf{com}}$

$\xrightarrow{x,\, \mathit{open}}$ $\mathsf{com} \stackrel{?}{=} \mathit{Commit}(x, \mathit{open})$

Making Classical (Threshold) Signatures Post-Quantum for Single Use on a Public Ledger

EPFL LASEC

Making Classical (Threshold) Signatures Post-Quantum for Single Use on a Public Ledger

EPFL LASEC

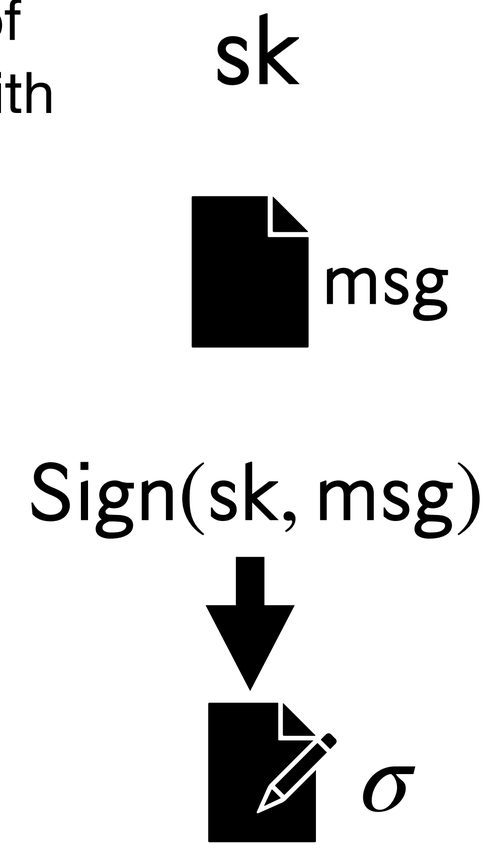
Digital Signatures

A digital signature scheme is a tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ with key space $\mathscr{D}$ and message space $\mathscr{M}$ such that:

$\mathsf{KeyGen}(1^\lambda) \rightarrow (\mathsf{sk}, \mathsf{pk})$

$\mathsf{Sign}(\mathsf{sk}, \mathsf{msg}) \rightarrow \sigma$

$\mathsf{Verify}(\sigma, \mathsf{pk}, \mathsf{msg}) \rightarrow 0/1$



pk

$\mathsf{Verify}(\sigma, \mathsf{pk}, \mathsf{msg})$

Discrete Log Problem (DLP)

$$x \quad \rightleftarrows \quad g^x$$

KeyGen

x $\rightleftarrows$ g^x

sk pk

Making Classical (Threshold) Signatures Post-Quantum for Single Use on a Public Ledger

EPFL
LASEC

Threshold Keys

x $\rightleftarrows$ g^x

sk pk

Threshold Keys

sk $\quad x = x_1 + x_2 + x_3$

$\mathsf{sk}_1 \quad x_1 \longrightarrow g^{x_1}$

$\mathsf{sk}_2 \quad x_2 \longrightarrow g^{x_2}$

$\mathsf{sk}_3 \quad x_3 \longrightarrow g^{x_3}$

Threshold Keys

sk $\quad x = x_1 + x_2 + x_3$

$\mathsf{sk}_1 \quad x_1 \longrightarrow g^{x_1}$

$\mathsf{sk}_2 \quad x_2 \longrightarrow g^{x_2}$

$\mathsf{sk}_3 \quad x_3 \longrightarrow g^{x_3}$

g^x

Threshold Keys

sk $\quad x = x_1 + x_2 + x_3$

$\mathsf{sk}_1 \quad x_1 \rightarrow g^{x_1}$

$\mathsf{sk}_2 \quad x_2 \rightarrow g^{x_2}$

$\mathsf{sk}_3 \quad x_3 \rightarrow g^{x_3}$

$$g^{x}$$

$$g^{x_1+x_2+x_3} = g^{x_1} \cdot g^{x_2} \cdot g^{x_3}$$

Threshold Keys

sk $\quad x = x_1 + x_2 + x_3$

$\mathsf{sk}_1 \quad x_1 \longrightarrow g^{x_1}$

$\mathsf{sk}_2 \quad x_2 \longrightarrow g^{x_2}$

$\mathsf{sk}_3 \quad x_3 \longrightarrow g^{x_3}$

g^{x}

$g^{x_1+x_2+x_3} = g^{x_1} \cdot g^{x_2} \cdot g^{x_3}$

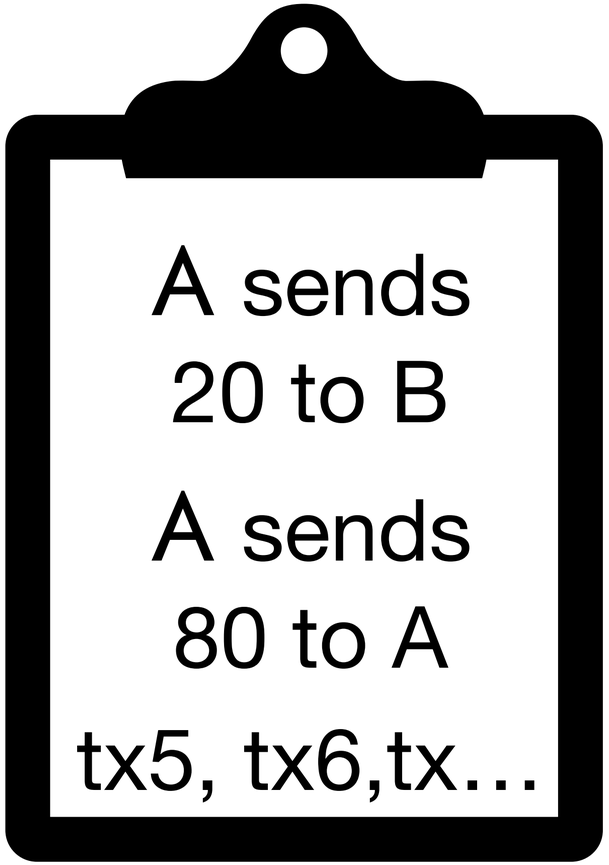
pk

Making Classical (Threshold) Signatures Post-Quantum for Single Use on a Public Ledger

EPFL LASEC

Public Ledger

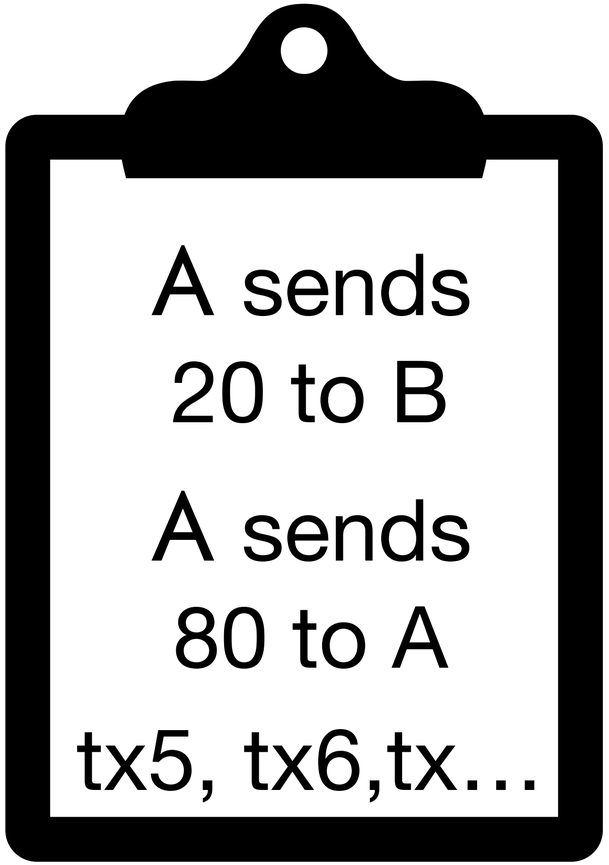
A
100



B

Public Ledger

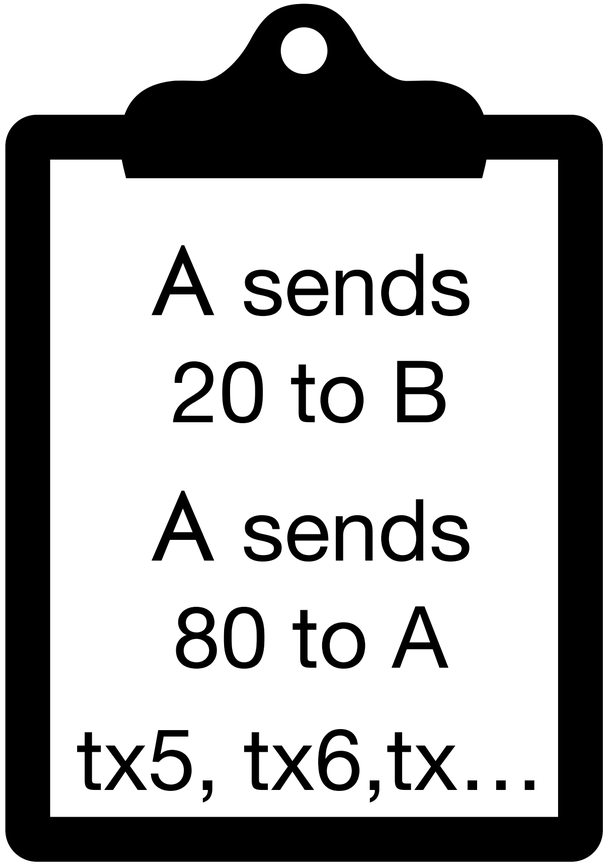
A
100



B

Public Ledger

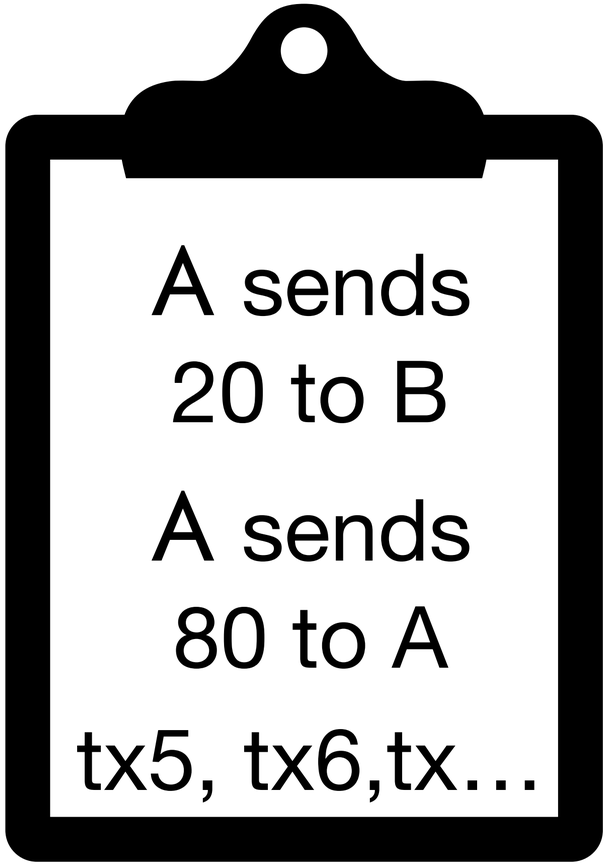
A
100



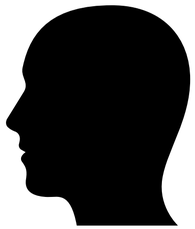
B

Public Ledger

A
100



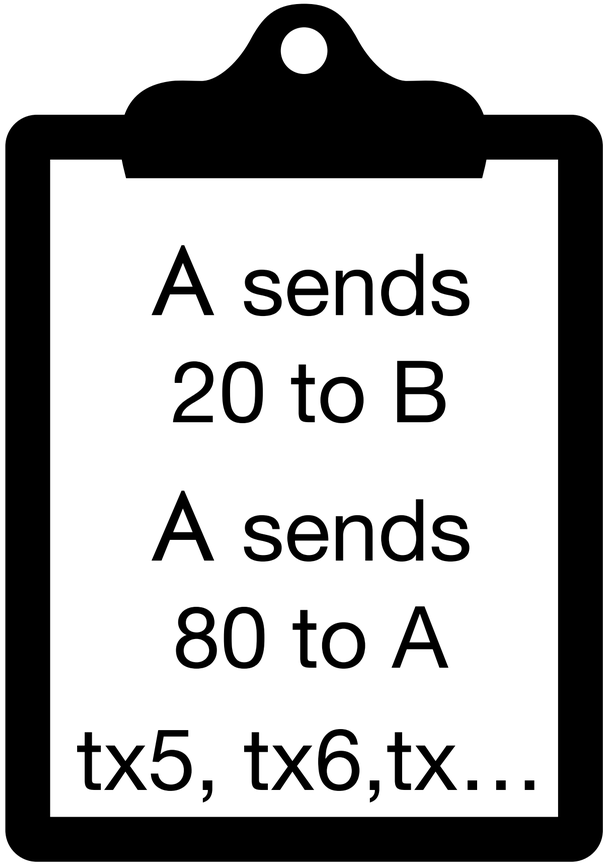
B



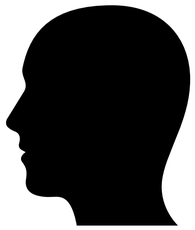
Public Ledger

A, sk_A, pk_A
100

$\sigma \leftarrow \text{Sign}(sk, tx)$



B, pk_B

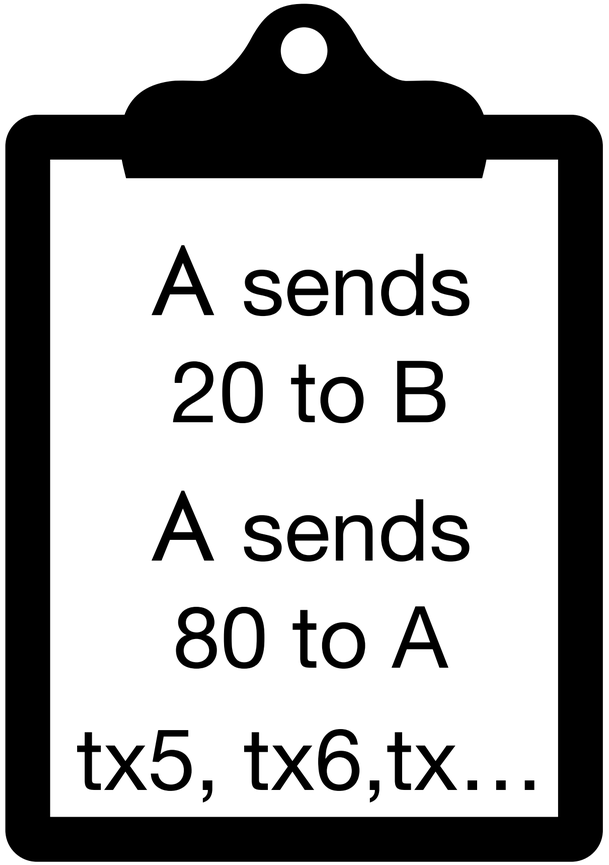


$\text{Verify}(\sigma, \text{pk}, \text{tx})$

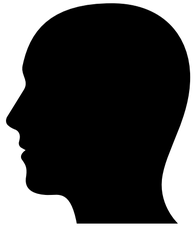
Public Ledger

A, sk_A, pk_A
100

$\sigma \leftarrow \text{Sign}(sk, tx)$



B, pk_B

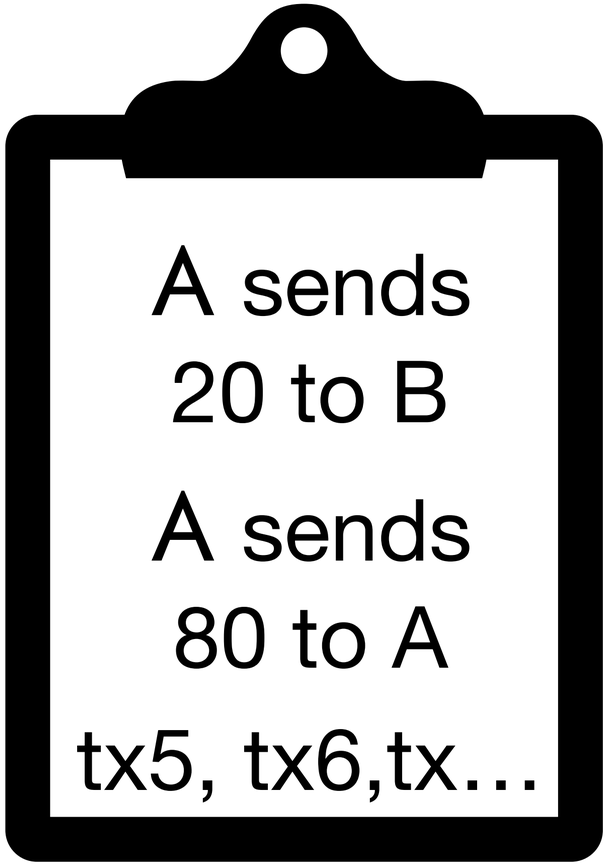


Verify(σ, pk, tx)

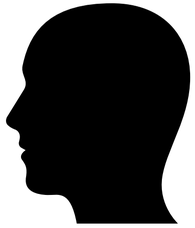
Public Ledger

A, sk_A, pk_A
100

$\sigma \leftarrow \text{Sign}(sk, tx)$



B, pk_B

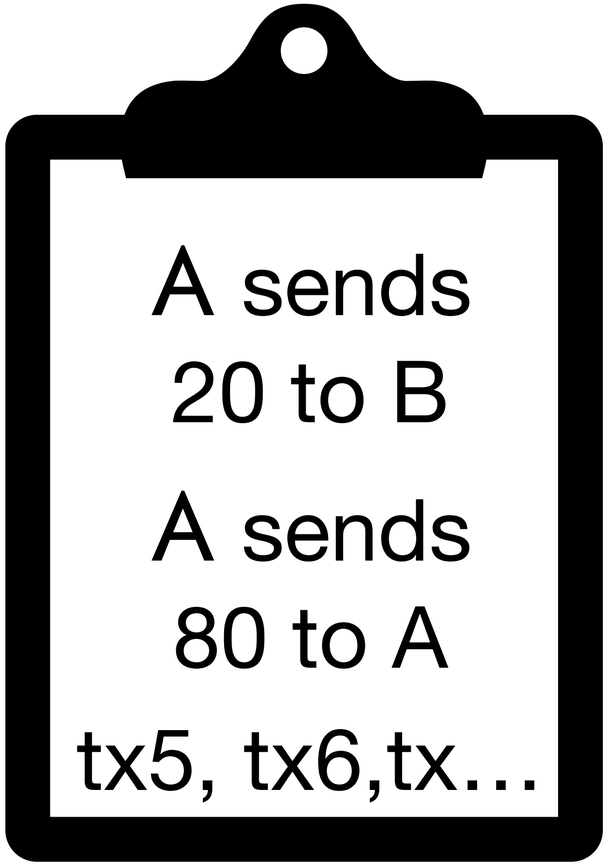


$\text{Verify}(\sigma, \text{pk}, \text{tx})$

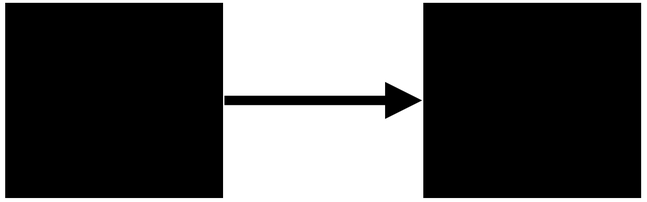
Public Ledger

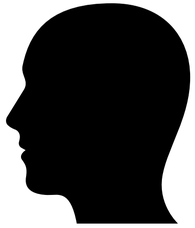
A, sk_A, pk_A
100

$\sigma \leftarrow \text{Sign}(sk, tx)$



B, pk_B



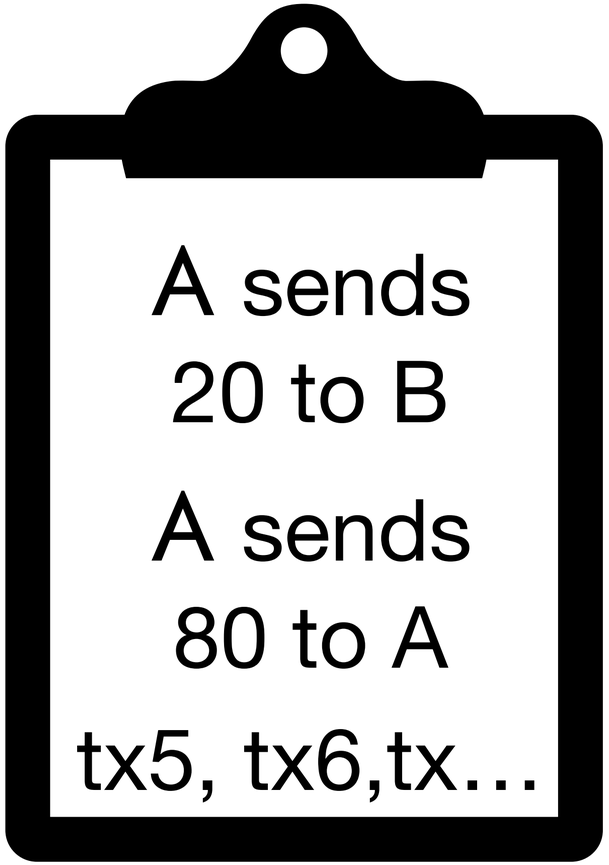


Verify(σ, pk, tx)

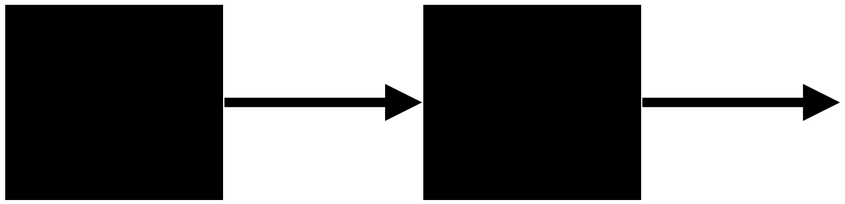
Public Ledger

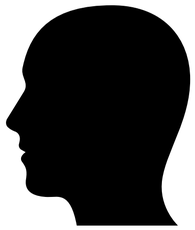
A, sk_A, pk_A
100

$\sigma \leftarrow \text{Sign}(sk, tx)$



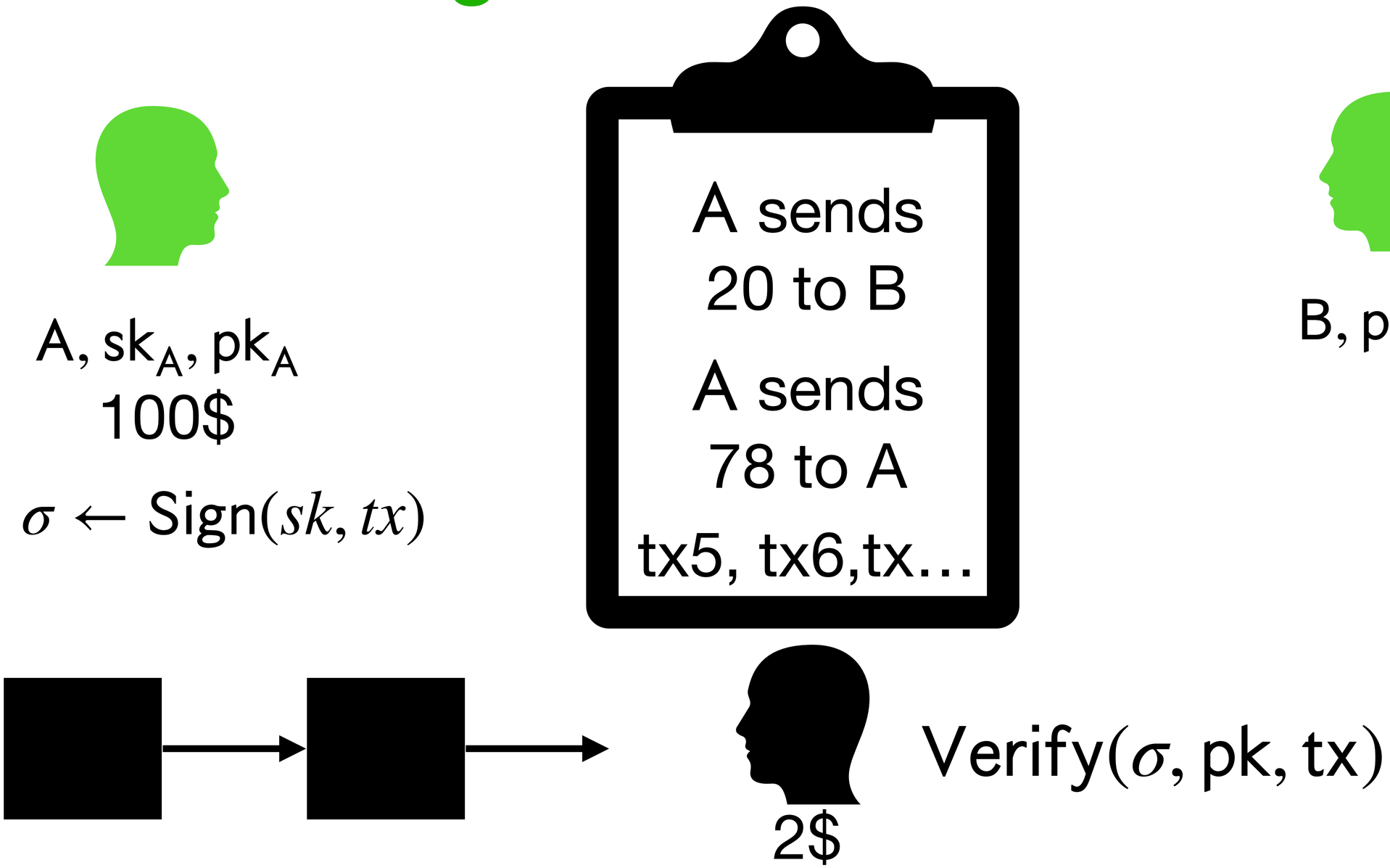
B, pk_B





Verify(σ, pk, tx)

Public Ledger



Making Classical (Threshold) Signatures Post-Quantum for Single Use on a Public Ledger

EPFL LASEC

Discrete Log Problem (DLP)

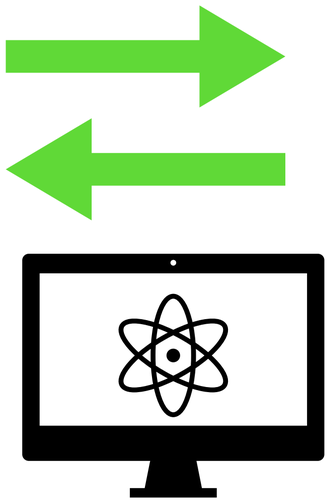
$$x \quad \rightleftarrows \quad g^x$$

Discrete Log Problem (DLP)

$$x \quad \rightleftarrows \quad g^x$$

Discrete Log Problem (DLP)

x



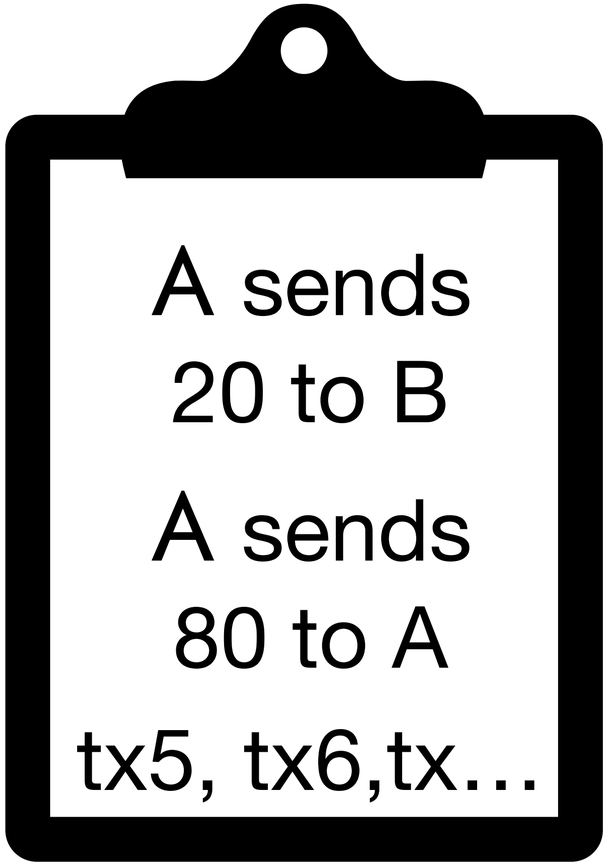
g^x

Quantum Computers

Public Ledger

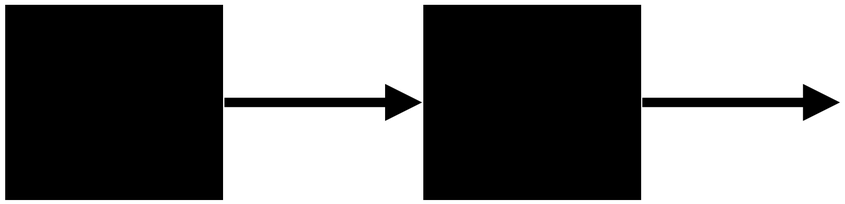
A, sk_A, pk_A
100

$\sigma \leftarrow \text{Sign}(sk, tx)$



B, pk_B

sk

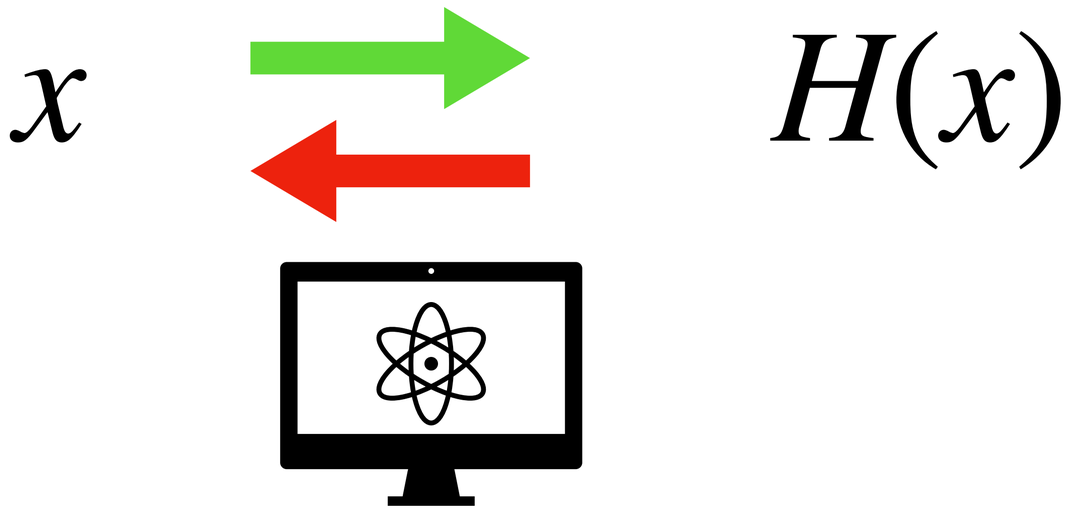


Verify(σ, pk, tx)

One-way Hash Functions

$$x \rightleftarrows H(x)$$

One-way Hash Functions



Quantum Computers

Making Classical (Threshold) Signatures Post-Quantum for Single Use on a Public Ledger

EPFL LASEC

Motivation

Motivation

Public Key Cryptography

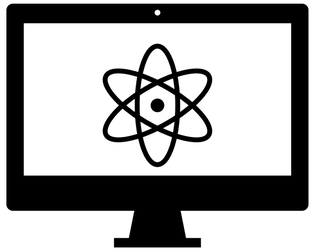
Motivation

+

Public Key Cryptography

Motivation

+



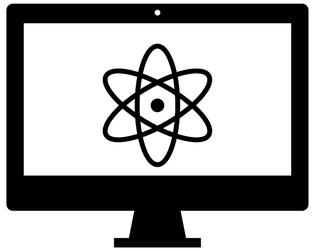
Public Key Cryptography

Quantum Computers

EPFL LASEC

Motivation

+



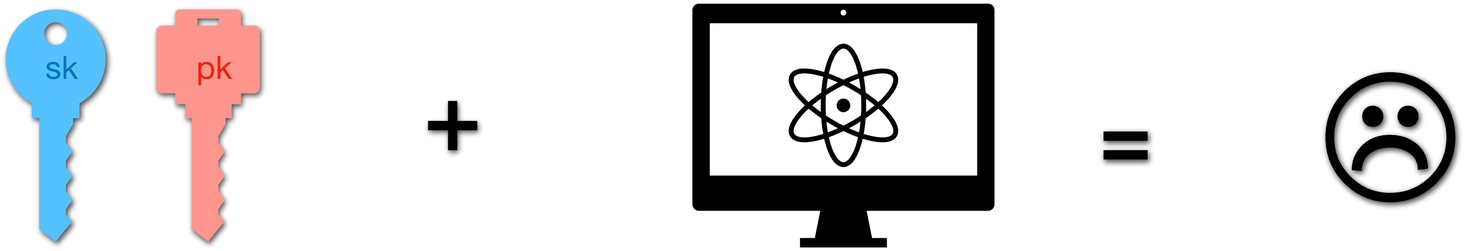
=

Public Key Cryptography

Quantum Computers

EPFL LASEC

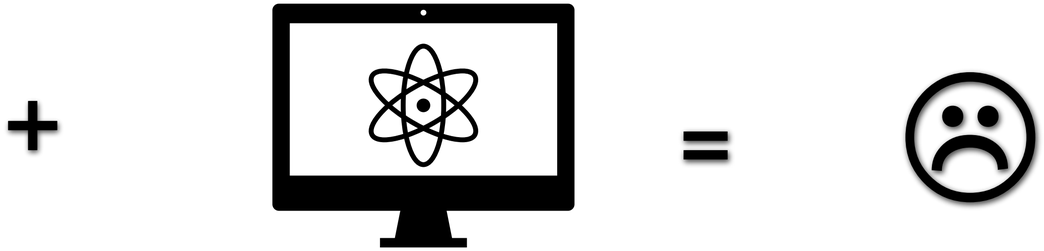
Motivation



Public Key Cryptography

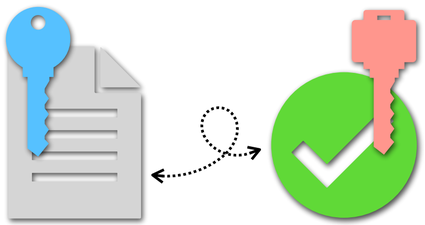
Quantum Computers

Motivation



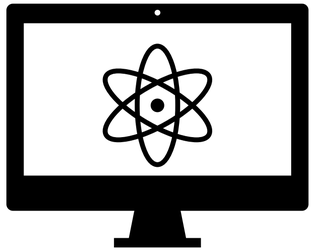
Quantum Computers

Motivation



Digital Signatures

+

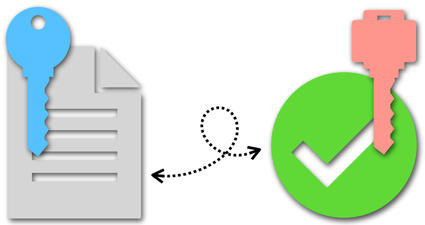


Quantum Computers

=

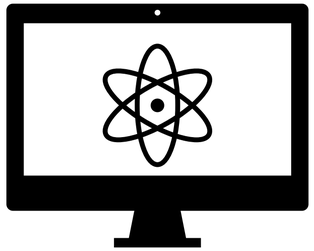
EPFL LASEC

Motivation



Digital Signatures

+



Quantum Computers

=

→

- Post-Quantum Signatures (NIST) : secure, not yet widely deployed
- Hiding the public key : one time secure, ready to use with classical algorithms

EPFL LASEC

P2PKH

P2PKH hides the public key behind a hash → Generalised HashedPK transform

HashedPK-KeyGen(λ)	HashedPK-Sign($\tilde{\mathsf{sk}}$, msg)	HashedPK-Verify($\tilde{\sigma}$, $\mathsf{pk}_{\mathcal{H}}$, msg)
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\$ \mathsf{KeyGen}_{\Sigma}(\lambda)$	$(\mathsf{sk}, \mathsf{pk}) \leftarrow \tilde{\mathsf{sk}}$	$(\sigma, \mathsf{pk}) \leftarrow \tilde{\sigma}$
$\mathsf{pk}_{\mathcal{H}} \leftarrow H(\mathsf{pk})$	$\sigma \leftarrow \mathsf{Sign}_{\Sigma}(\mathsf{sk}, \mathsf{msg})$	**if** $\mathsf{pk}_{\mathcal{H}} = H(\mathsf{pk})$:
$\tilde{\mathsf{sk}} \leftarrow (\mathsf{sk}, \mathsf{pk})$	$\tilde{\sigma} \leftarrow (\sigma, \mathsf{pk})$	**return** $\mathsf{Verify}_{\Sigma}(\sigma, \mathsf{pk}, \mathsf{msg})$
return $(\tilde{\mathsf{sk}}, \mathsf{pk}_{\mathcal{H}})$	**return** $\tilde{\sigma}$	**return** 0

Hiding the public key

$\Sigma = (\mathsf{KeyGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$
with additive public key domain
+ hash function H

$\xrightarrow{\text{Generic Transform}}$

Signature scheme with hidden public key

EPFL LASEC

Hiding the public key

$\Sigma = (\mathsf{KeyGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$
with additive public key domain
+ hash function H

$\xrightarrow{\text{Generic Transform}}$

Signature scheme with hidden public key

HiddenPK-KeyGen(λ)

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \mathsf{KeyGen}_\Sigma(\lambda)$
2 : $\rho \leftarrow\$ \mathcal{D}$
3 : $\tilde{\mathsf{sk}} \leftarrow (\mathsf{sk}, \rho)$
4 : $\tilde{\mathsf{pk}} \leftarrow (H(\rho), \mathsf{pk} + \rho)$
5 : **return** $(\tilde{\mathsf{sk}}, \tilde{\mathsf{pk}})$

Hiding the public key

$\Sigma = (\mathsf{KeyGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$
with additive public key domain
+ hash function H

$\xrightarrow{\text{Generic Transform}}$

Signature scheme with hidden public key

HiddenPK-KeyGen(λ)

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \mathsf{KeyGen}_\Sigma(\lambda)$
2 : $\rho \leftarrow\$ \mathcal{D}$
3 : $\tilde{\mathsf{sk}} \leftarrow (\mathsf{sk}, \rho)$
4 : $\tilde{\mathsf{pk}} \leftarrow (H(\rho), \mathsf{pk} + \rho)$
5 : **return** $(\tilde{\mathsf{sk}}, \tilde{\mathsf{pk}})$

Hiding the public key

$\Sigma = (\mathsf{KeyGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$
with additive public key domain
+ hash function H

$\xrightarrow{\text{Generic Transform}}$

Signature scheme with hidden public key

HiddenPK-KeyGen(λ)	HiddenPK-Sign($\tilde{\mathsf{sk}}$, msg)
$(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \mathsf{KeyGen}_\Sigma(\lambda)$	$(\mathsf{sk}, \rho) \leftarrow \tilde{\mathsf{sk}}$
$\rho \leftarrow\$ \mathcal{D}$	$\sigma \leftarrow \mathsf{Sign}_\Sigma(\mathsf{sk}, \mathsf{msg})$
$\tilde{\mathsf{sk}} \leftarrow (\mathsf{sk}, \rho)$	$\tilde{\sigma} \leftarrow (\sigma, \rho)$
$\tilde{\mathsf{pk}} \leftarrow (H(\rho), \mathsf{pk} + \rho)$	**return** $\tilde{\sigma}$
return $(\tilde{\mathsf{sk}}, \tilde{\mathsf{pk}})$	

Hiding the public key

$\Sigma = (\mathsf{KeyGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$
with additive public key domain
+ hash function H

$\xrightarrow{\text{Generic Transform}}$

Signature scheme with hidden public key

HiddenPK-KeyGen(λ)	HiddenPK-Sign($\tilde{\mathsf{sk}}$, msg)	HiddenPK-Verify($\tilde{\sigma}$, $\tilde{\mathsf{pk}}$, msg)
$(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \mathsf{KeyGen}_\Sigma(\lambda)$	$(\mathsf{sk}, \rho) \leftarrow \tilde{\mathsf{sk}}$	$(\sigma, \rho) \leftarrow \tilde{\sigma}$
$\rho \leftarrow\$ \mathcal{D}$	$\sigma \leftarrow \mathsf{Sign}_\Sigma(\mathsf{sk}, \mathsf{msg})$	$(\tilde{\mathsf{pk}}_1, \tilde{\mathsf{pk}}_2) \leftarrow \tilde{\mathsf{pk}}$
$\tilde{\mathsf{sk}} \leftarrow (\mathsf{sk}, \rho)$	$\tilde{\sigma} \leftarrow (\sigma, \rho)$	**if** $H(\rho) = \tilde{\mathsf{pk}}_1$
$\tilde{\mathsf{pk}} \leftarrow (H(\rho), \mathsf{pk} + \rho)$	**return** $\tilde{\sigma}$	$\mathsf{pk} = \tilde{\mathsf{pk}}_2 - \rho$
return $(\tilde{\mathsf{sk}}, \tilde{\mathsf{pk}})$		**return** $\mathsf{Verify}_\Sigma(\sigma, \mathsf{pk}, \mathsf{msg})$
		return 0

Hiding the public key

$\Sigma = (\mathsf{KeyGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$
with additive public key domain
+ hash function H

$\xrightarrow{\text{Generic Transform}}$

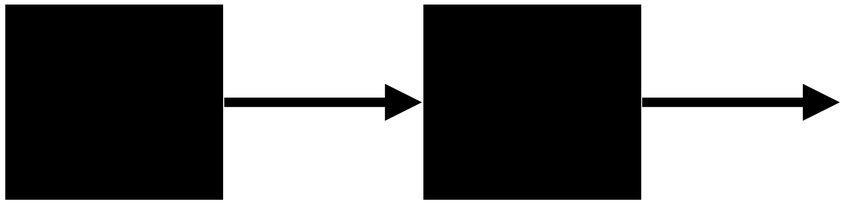
Signature scheme with hidden public key

HiddenPK-KeyGen(λ)	HiddenPK-Sign($\tilde{\mathsf{sk}}$, msg)	HiddenPK-Verify($\tilde{\sigma}$, $\tilde{\mathsf{pk}}$, msg)
$(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \mathsf{KeyGen}_\Sigma(\lambda)$	$(\mathsf{sk}, \rho) \leftarrow \tilde{\mathsf{sk}}$	$(\sigma, \rho) \leftarrow \tilde{\sigma}$
$\rho \leftarrow\$ \mathcal{D}$	$\sigma \leftarrow \mathsf{Sign}_\Sigma(\mathsf{sk}, \mathsf{msg})$	$(\tilde{\mathsf{pk}}_1, \tilde{\mathsf{pk}}_2) \leftarrow \tilde{\mathsf{pk}}$
$\tilde{\mathsf{sk}} \leftarrow (\mathsf{sk}, \rho)$	$\tilde{\sigma} \leftarrow (\sigma, \rho)$	**if** $H(\rho) = \tilde{\mathsf{pk}}_1$
$\tilde{\mathsf{pk}} \leftarrow (H(\rho), \mathsf{pk} + \rho)$	**return** $\tilde{\sigma}$	$\mathsf{pk} = \tilde{\mathsf{pk}}_2 - \rho$
return $(\tilde{\mathsf{sk}}, \tilde{\mathsf{pk}})$		**return** $\mathsf{Verify}_\Sigma(\sigma, \mathsf{pk}, \mathsf{msg})$
		return 0

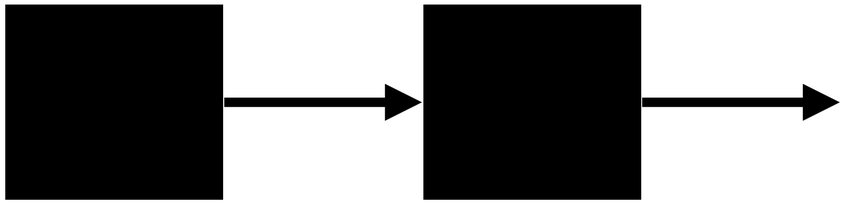
Problem
Front Running

EPFL
LASEC

Front Running



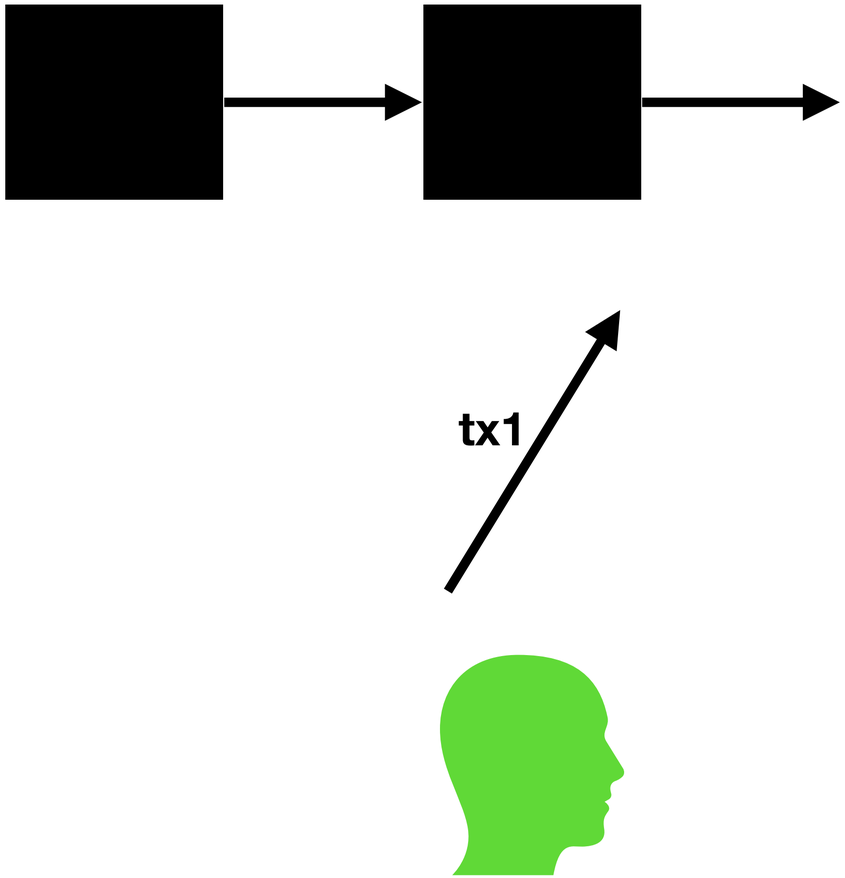
Front Running



$\tilde{\mathsf{pk}} : (H(\rho), \mathsf{pk} + \rho)$

EPFL LASEC

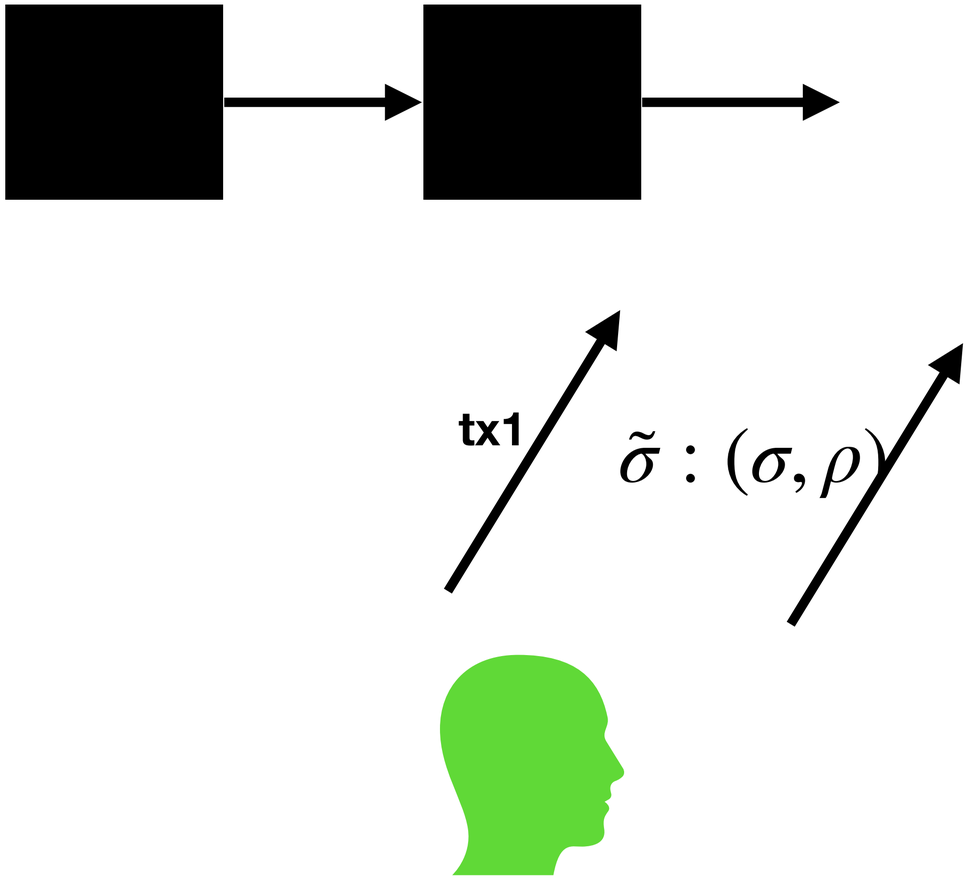
Front Running



$\tilde{\mathsf{pk}} : (H(\rho), \mathsf{pk} + \rho)$

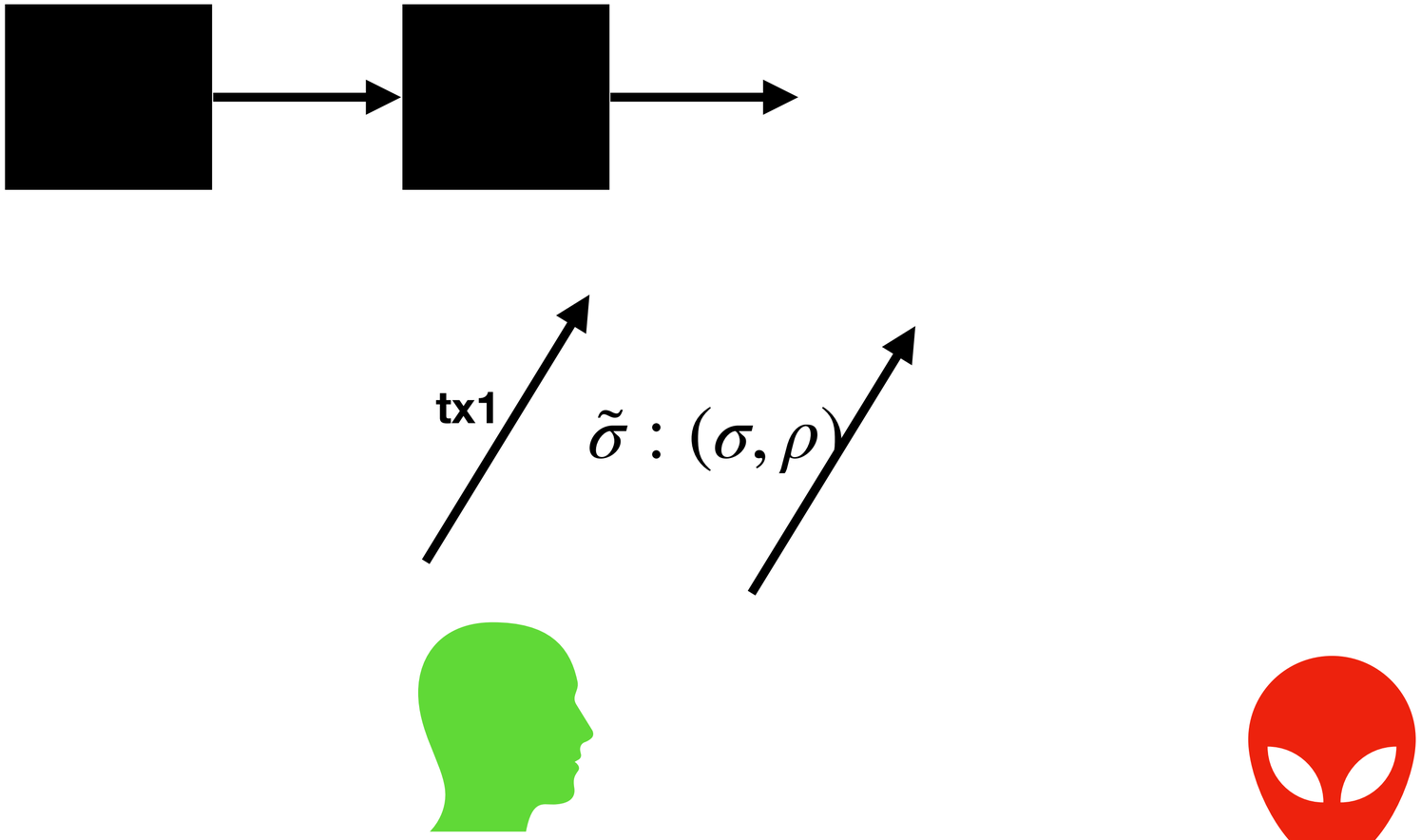
EPFL LASEC

Front Running



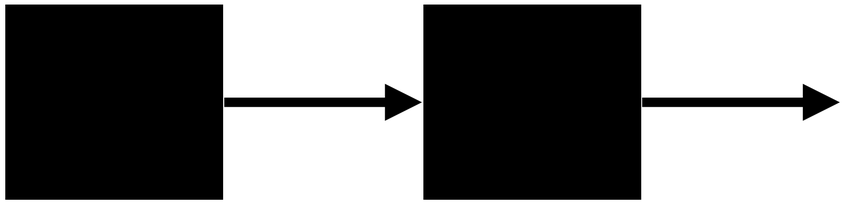
EPFL LASEC

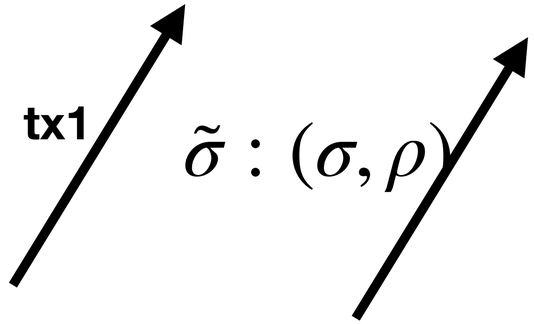
Front Running



EPFL LASEC

Front Running



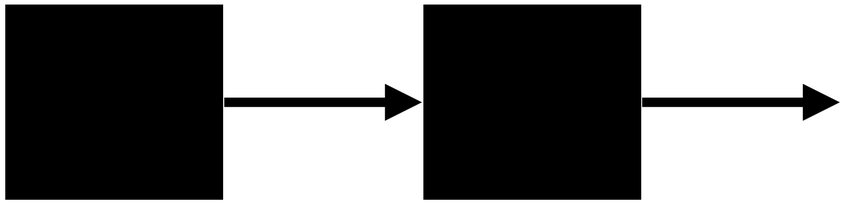


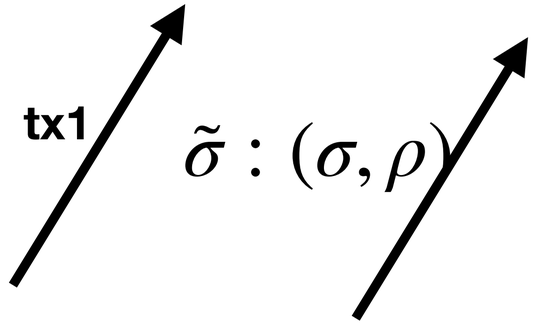
$\tilde{\sigma} : (\sigma, \rho)$

$\tilde{\mathsf{pk}} : (H(\rho), \mathsf{pk} + \rho)$

EPFL LASEC

Front Running





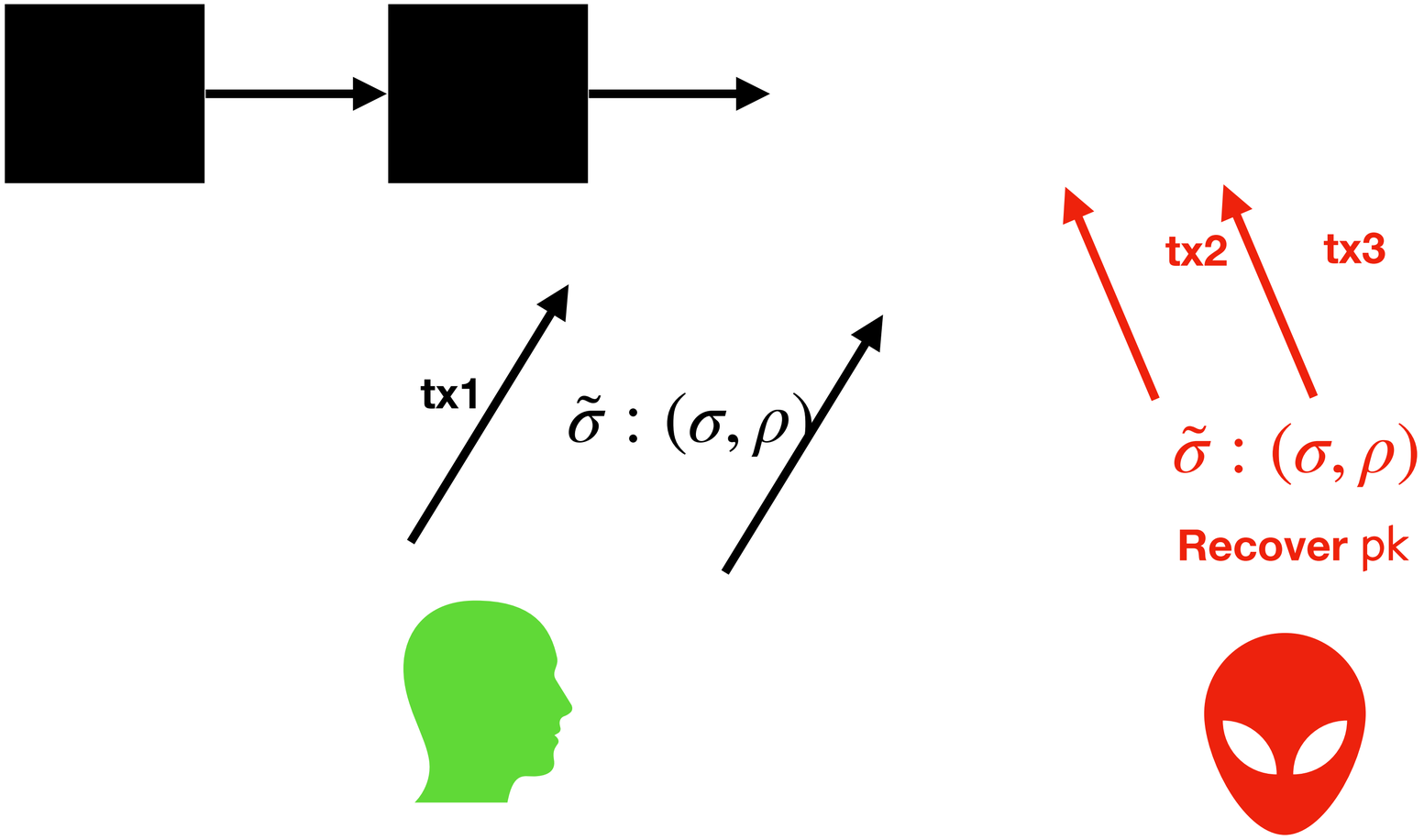
$\tilde{\mathsf{pk}} : (H(\rho), \mathsf{pk} + \rho)$

$\tilde{\sigma} : (\sigma, \rho)$

Recover pk

EPFL LASEC

Front Running



EPFL LASEC

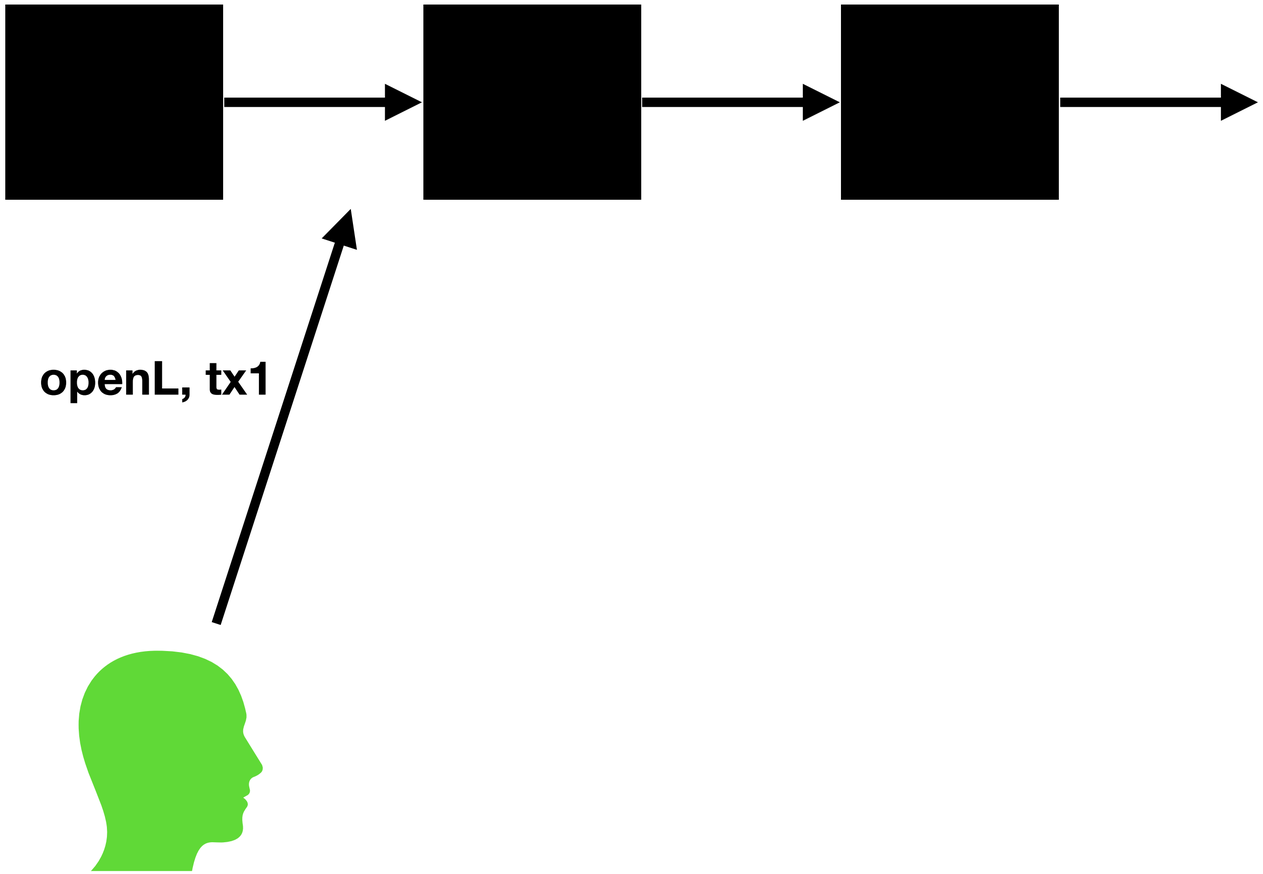
One-Time Signature

with Delayed Ledger

EPFL LASEC

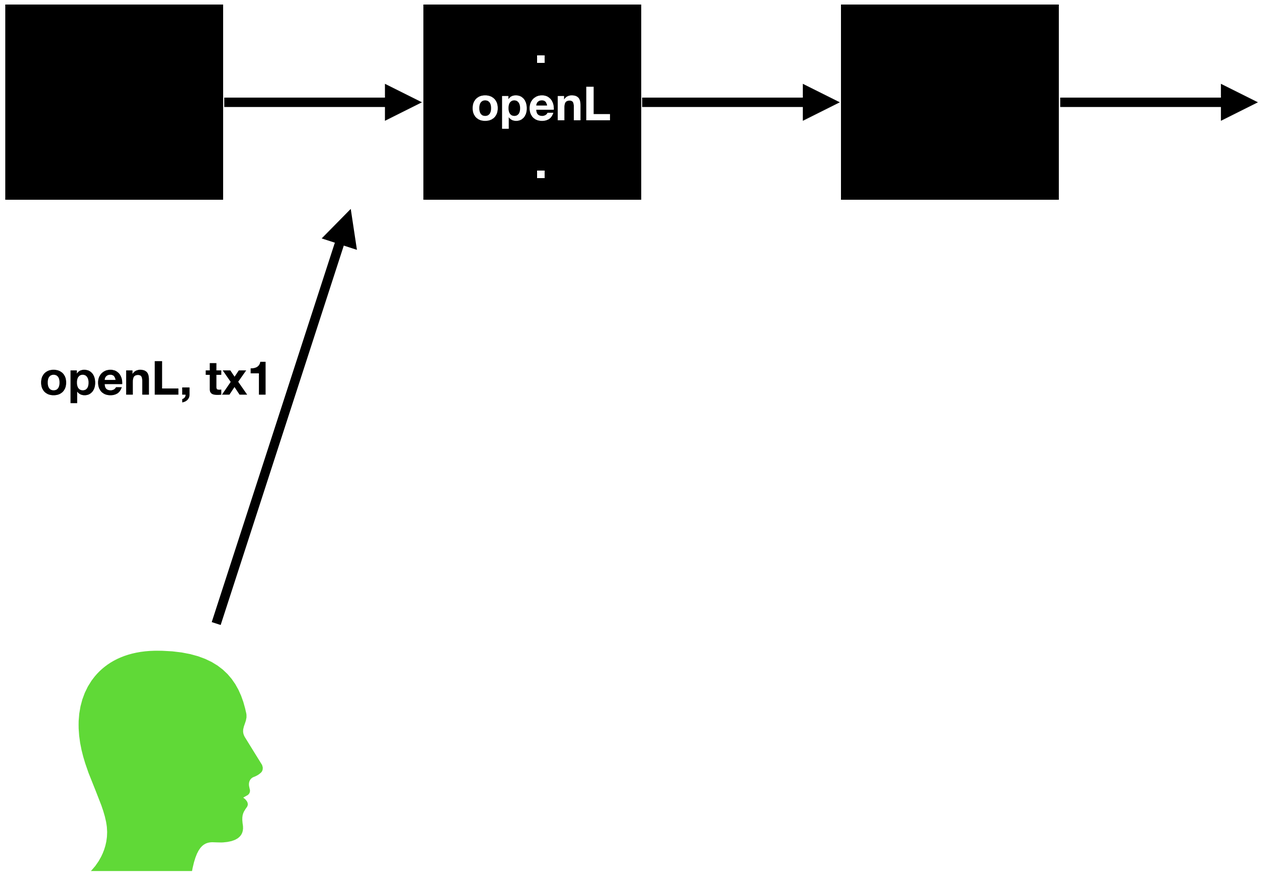
One-Time Signature

with Delayed Ledger



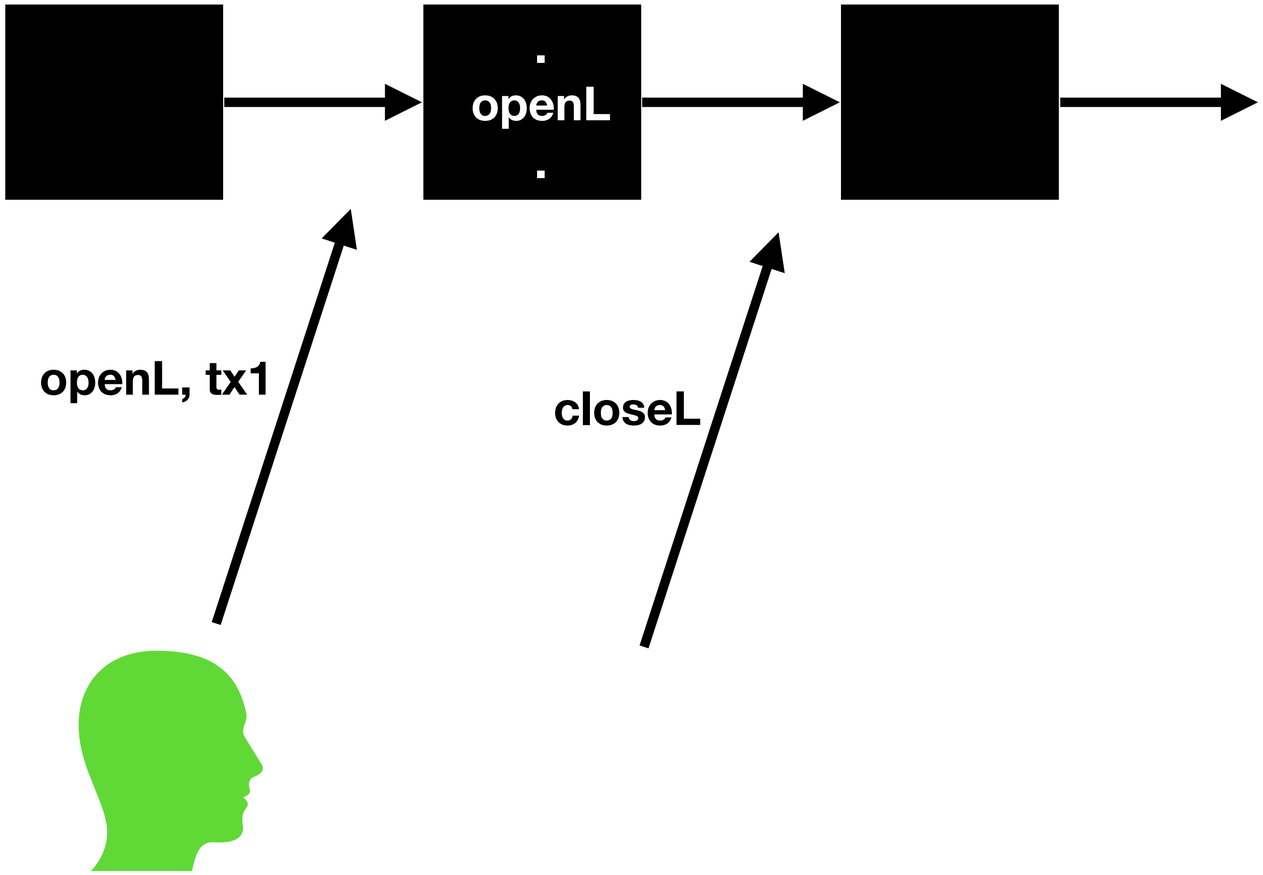
One-Time Signature

with Delayed Ledger



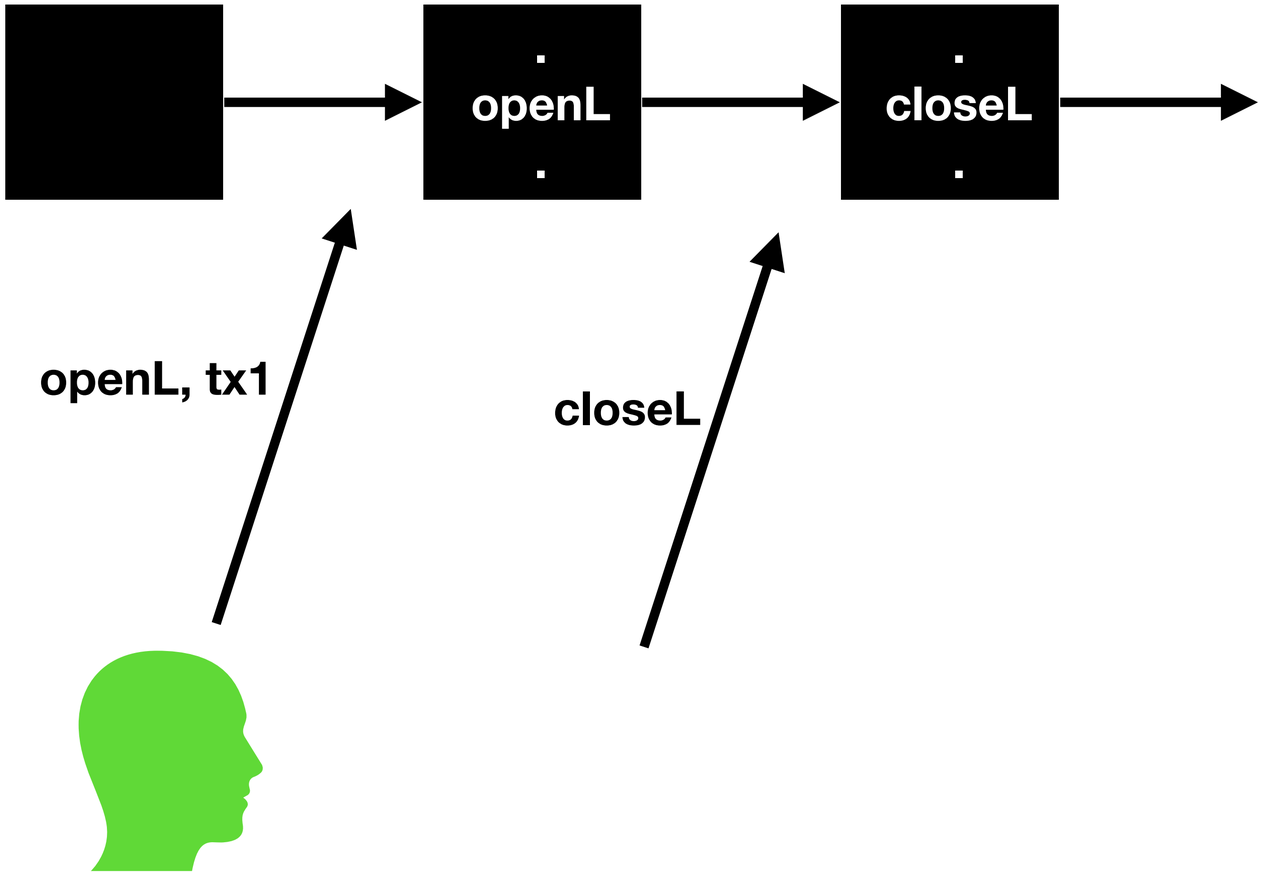
One-Time Signature

with Delayed Ledger



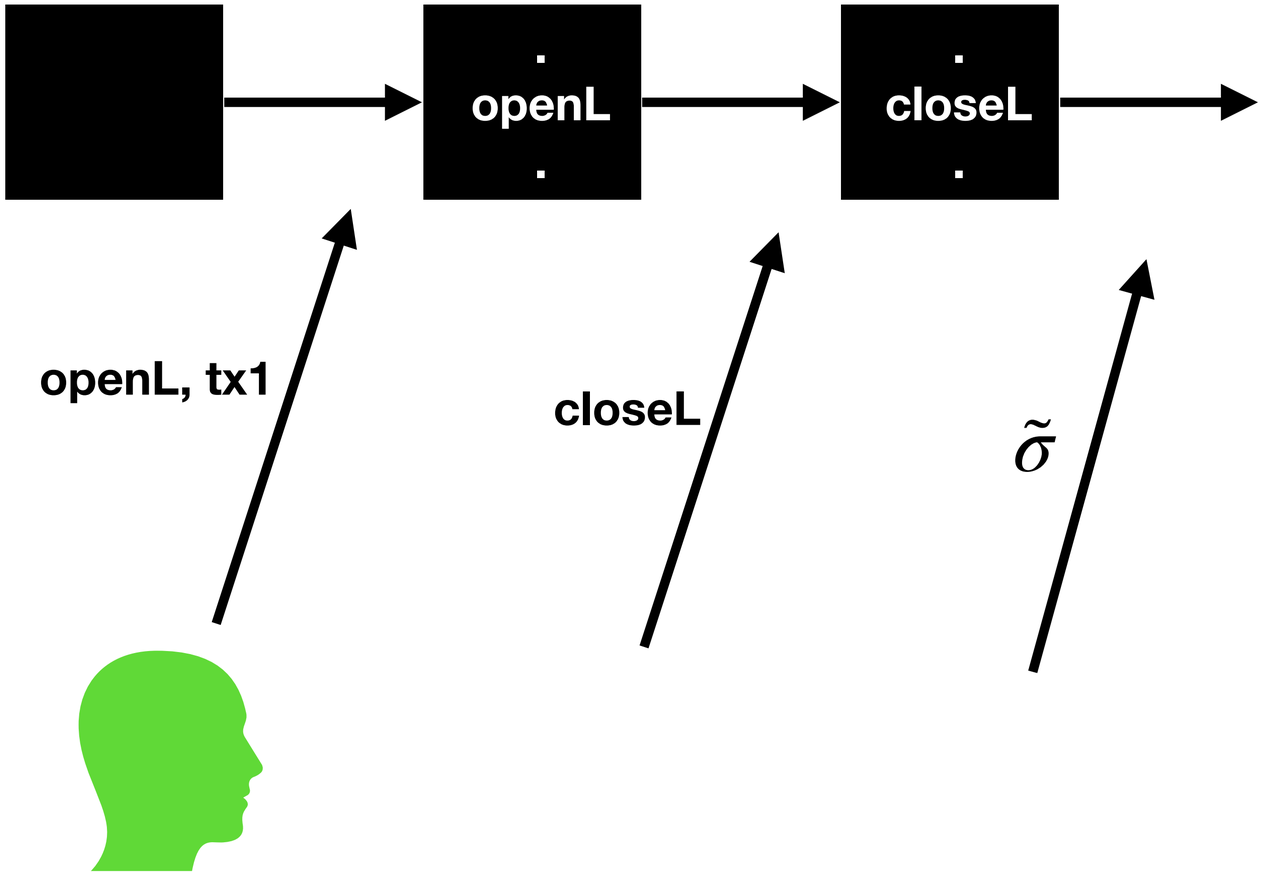
One-Time Signature

with Delayed Ledger



One-Time Signature

with Delayed Ledger



Conclusion & Takeaway

Conclusion & Takeaway

- Formal study of hiding a public key in a public ledger setting.

Conclusion & Takeaway

- Formal study of hiding a public key in a public ledger setting.
- HiddenPK is easier to extend to threshold setting compared to P2PKH.

EPFL LASEC

Conclusion & Takeaway

- Formal study of hiding a public key in a public ledger setting.
- HiddenPK is easier to extend to threshold setting compared to P2PKH.
- Practical instantiations require improvement in DDoS protection & miner motivation.

EPFL LASEC

Thanks!

EPFL
LASEC