Logarithmic-size (linkable) **threshold** ring signatures in the plain model

Abida Haque, Stephan Krenn, Daniel Slamanig, Christoph Striecks

PKC 2022

NC STATE

Cyber Security for Europe

AIT AUSTRIAN INSTITUTE OF TECHNOLOGY

PART I:
Background and Contribution

Ring Signatures and their Setting

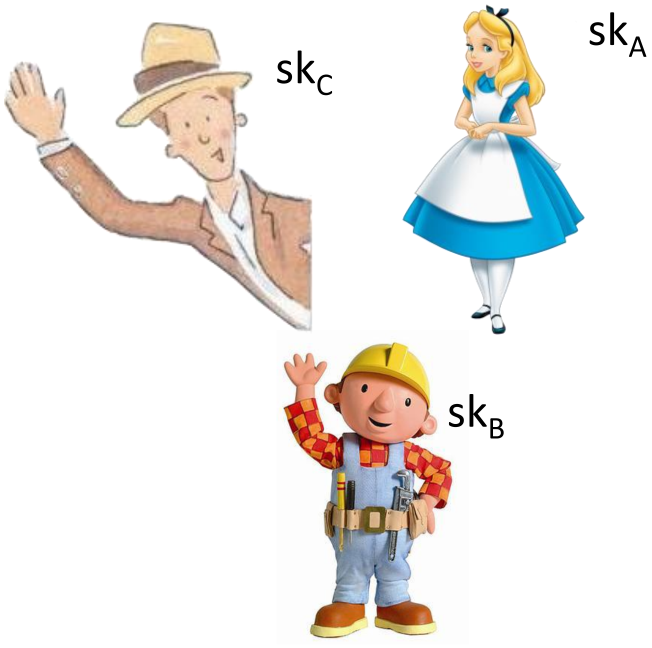
Ring Signatures and their Setting

$R=(vk_A$

sk_A

Ring Signatures and their Setting

$R=(vk_A, vk_B, vk_C)$



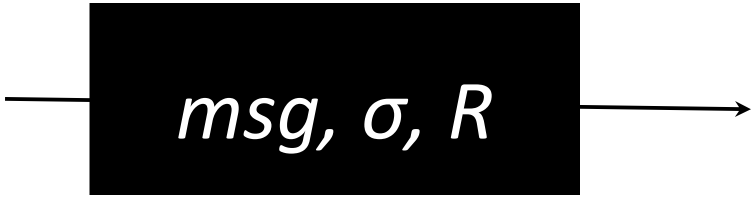
Ring Signatures and their Setting

$R=(vk_A, vk_B, vk_C)$

sk_C

sk_A

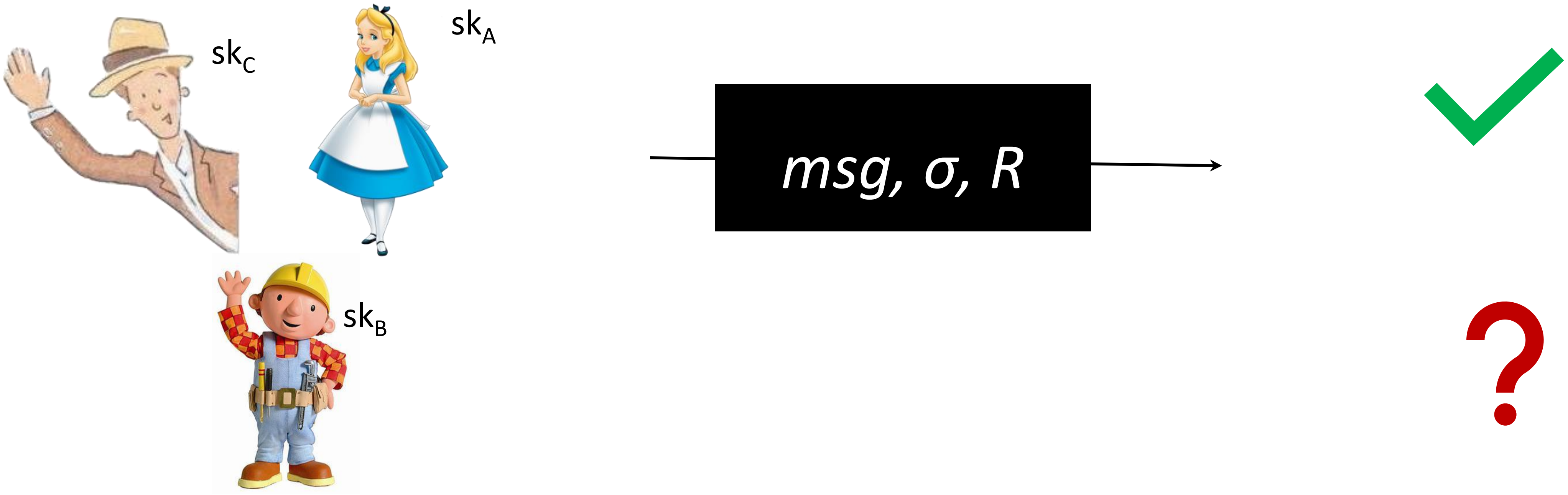
sk_B



msg, σ, R

Ring Signatures and their Setting

$R=(vk_A, vk_B, vk_C)$



sk_C

sk_A

sk_B

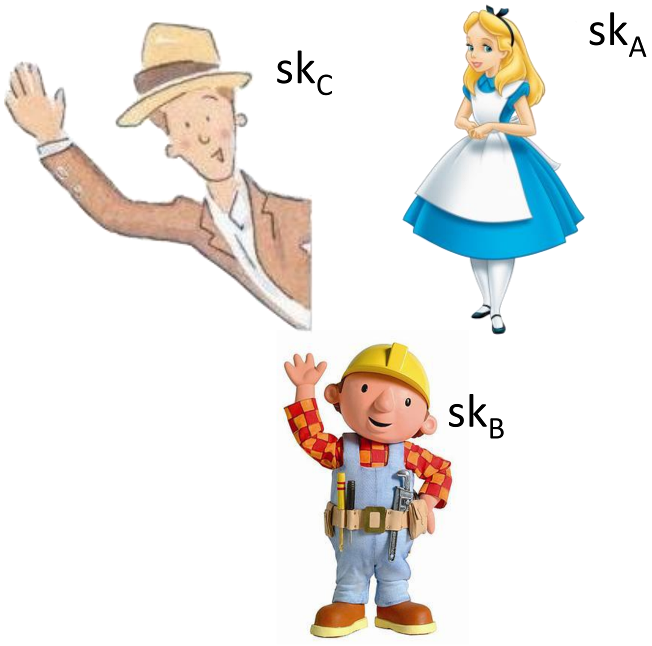
msg, σ, R

✓

?

Ring Signatures and their Setting

$R=(vk_A, vk_B, vk_C)$



msg, σ, R

✓

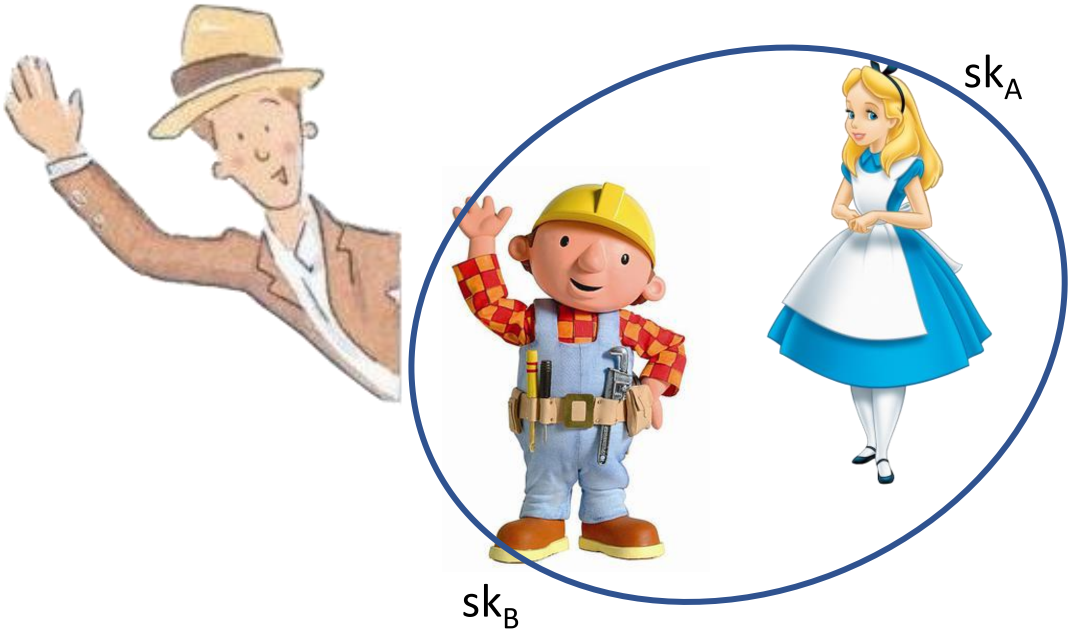
unforgeability

?

anonymity

Variant: Thring Signatures

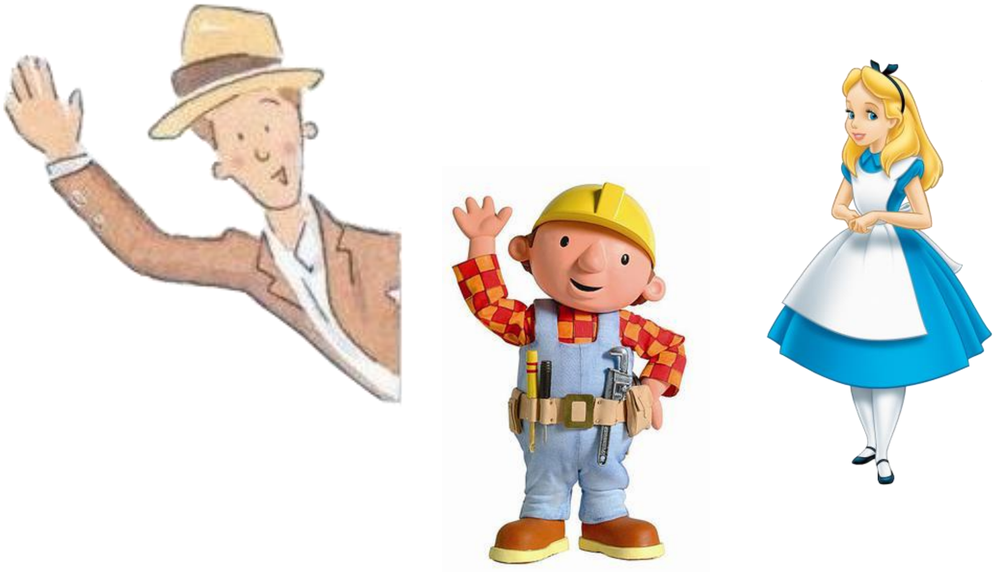
Variant: Thring Signatures



Variant: Thring Signatures

1. Interaction between parties

Variant: Thring Signatures



1. Interaction between parties
2. Intersigner anonymous

Intersigner Anonymity

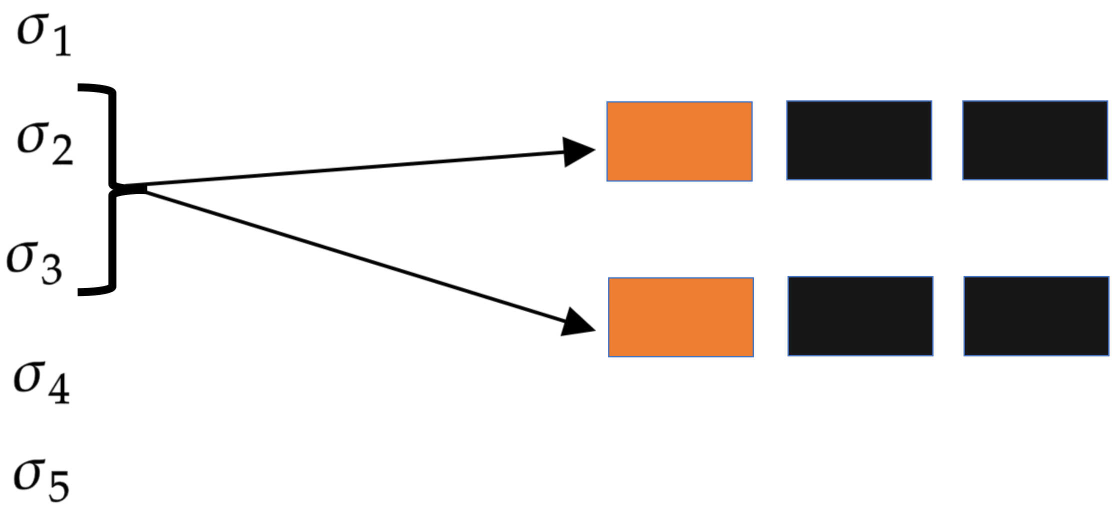
- Signature has a deterministic part

Intersigner Anonymity

- Signature has a deterministic part
- Given two signatures, check if that part is equal

Intersigner Anonymity

- Signature has a deterministic part
- Given two signatures, check if that part is equal



σ_1

σ_2

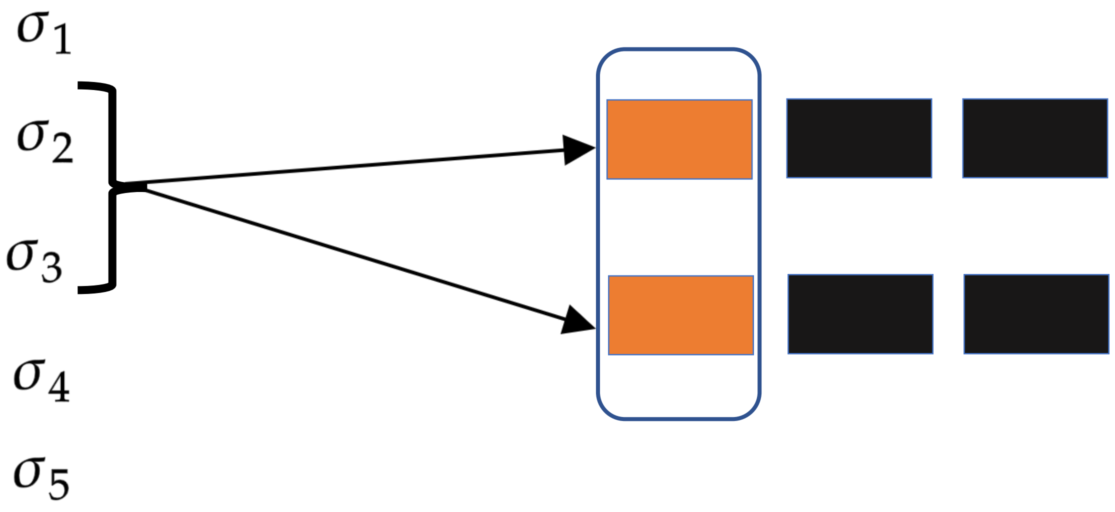
σ_3

σ_4

σ_5

Intersigner Anonymity

- Signature has a deterministic part
- Given two signatures, check if that part is equal

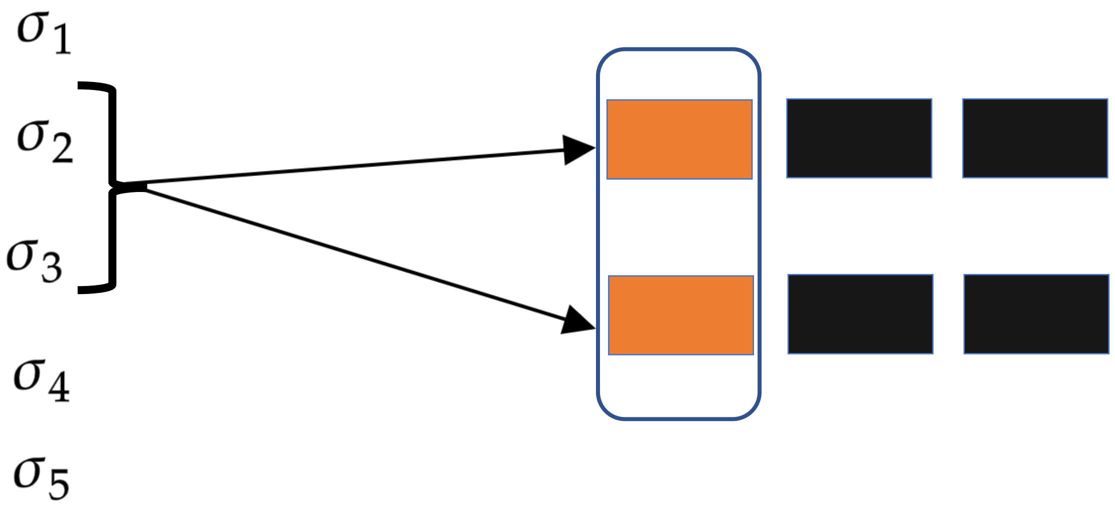


≠ different

= same

Intersigner Anonymity

- Signature has a deterministic part
- Given two signatures on same m, R, check if that part is equal



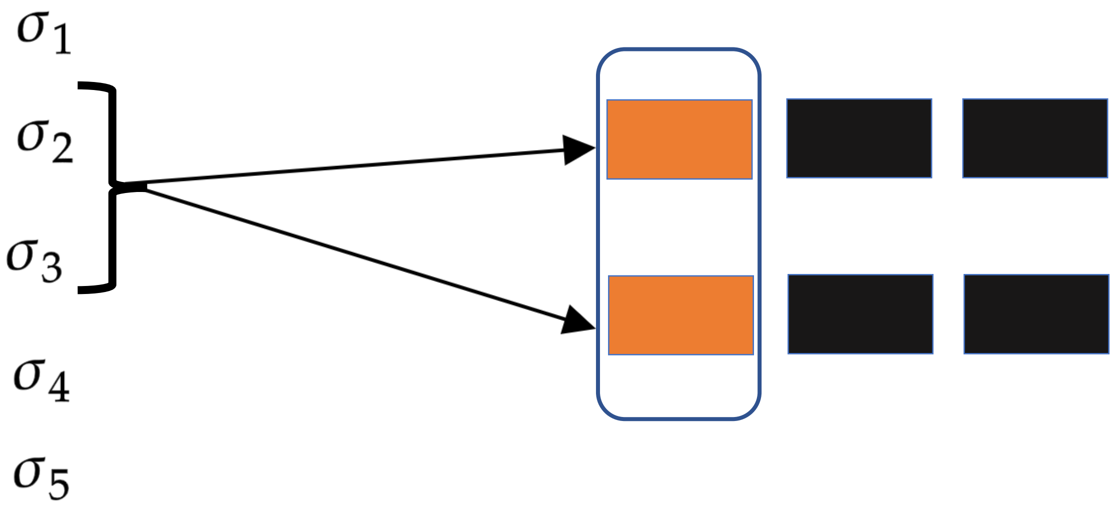
≠ different

= same

Intersigner Anonymity

- Signature has a deterministic part
- Given two signatures on same m, R, check if that part is equal

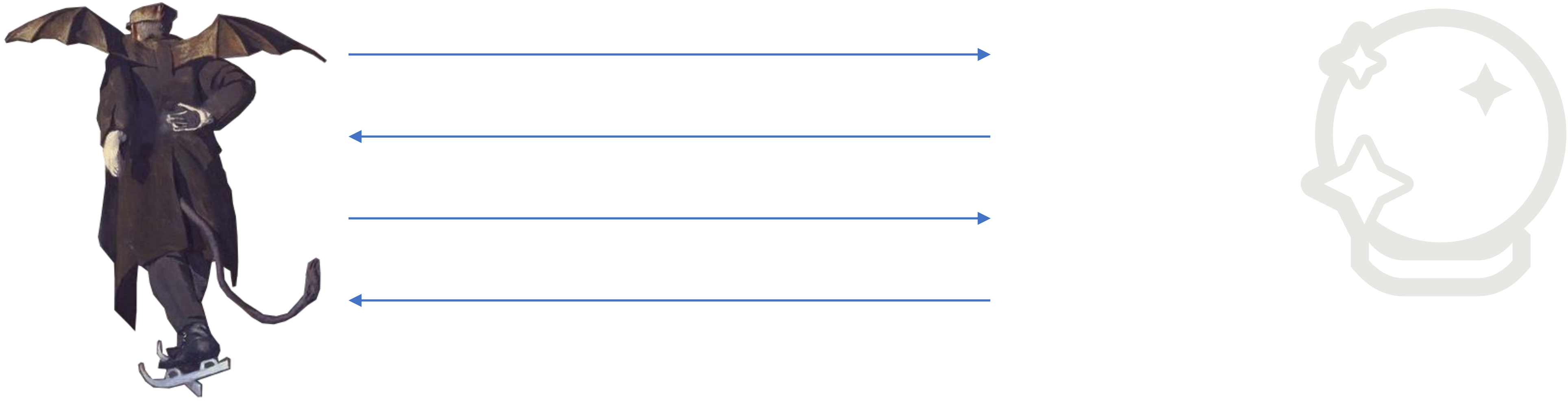
More fine-grained than linkability!



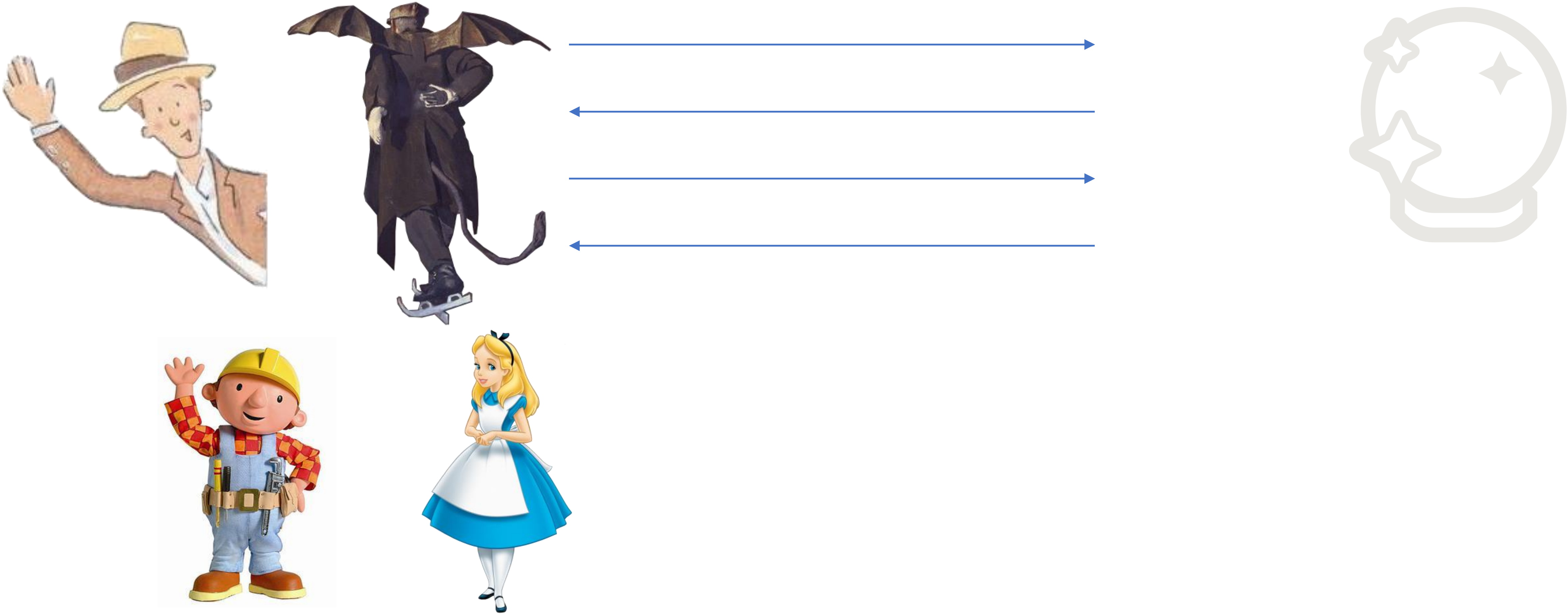
≠ different

= same

Adversary against Unforgeability and Anonymity



Adversary against Unforgeability and Anonymity



Oracles for the Active Adversary

OSign

OKGen

OCorr

OReg

PART II: Construction and Proofs

Building Blocks

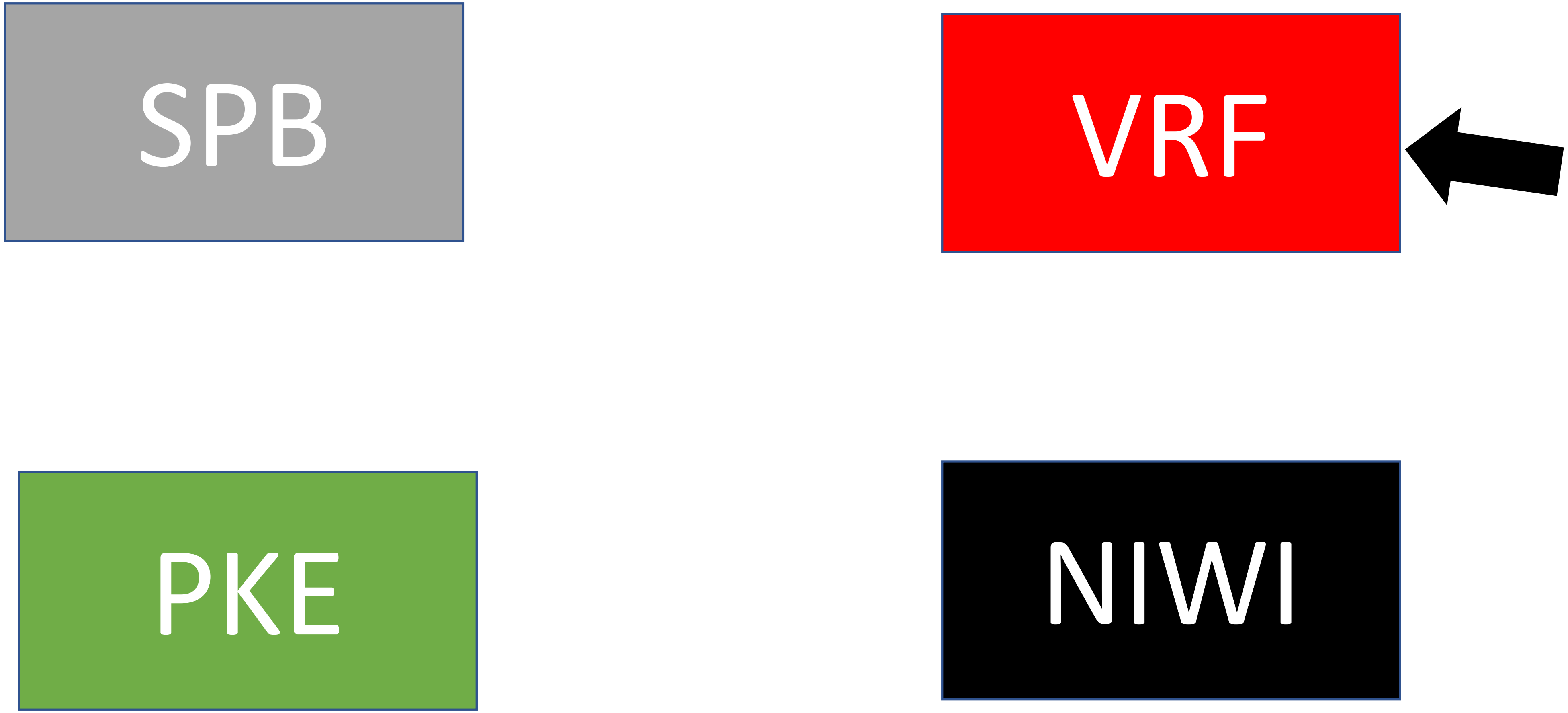
SPB

VRF

PKE

NIWI

Building Blocks



SPB

VRF

PKE

NIWI

Inspired by [BDHKS19]

VRF

Verifiable Random Function

- Like a PRF
- But can generate a proof

VRF

Verifiable Random Function

- Like a PRF
- But can generate a proof
- $v = Eval(sk, x)$
- $p = Prove(sk, x, v)$

VRF

Verifiable Random Function

- Like a PRF
- But can generate a proof
- $v = Eval(sk, x)$
- $p = Prove(sk, x, v)$

- Verification algorithm:
  - $Verify(v, p, vk) = 1/0$

PKE

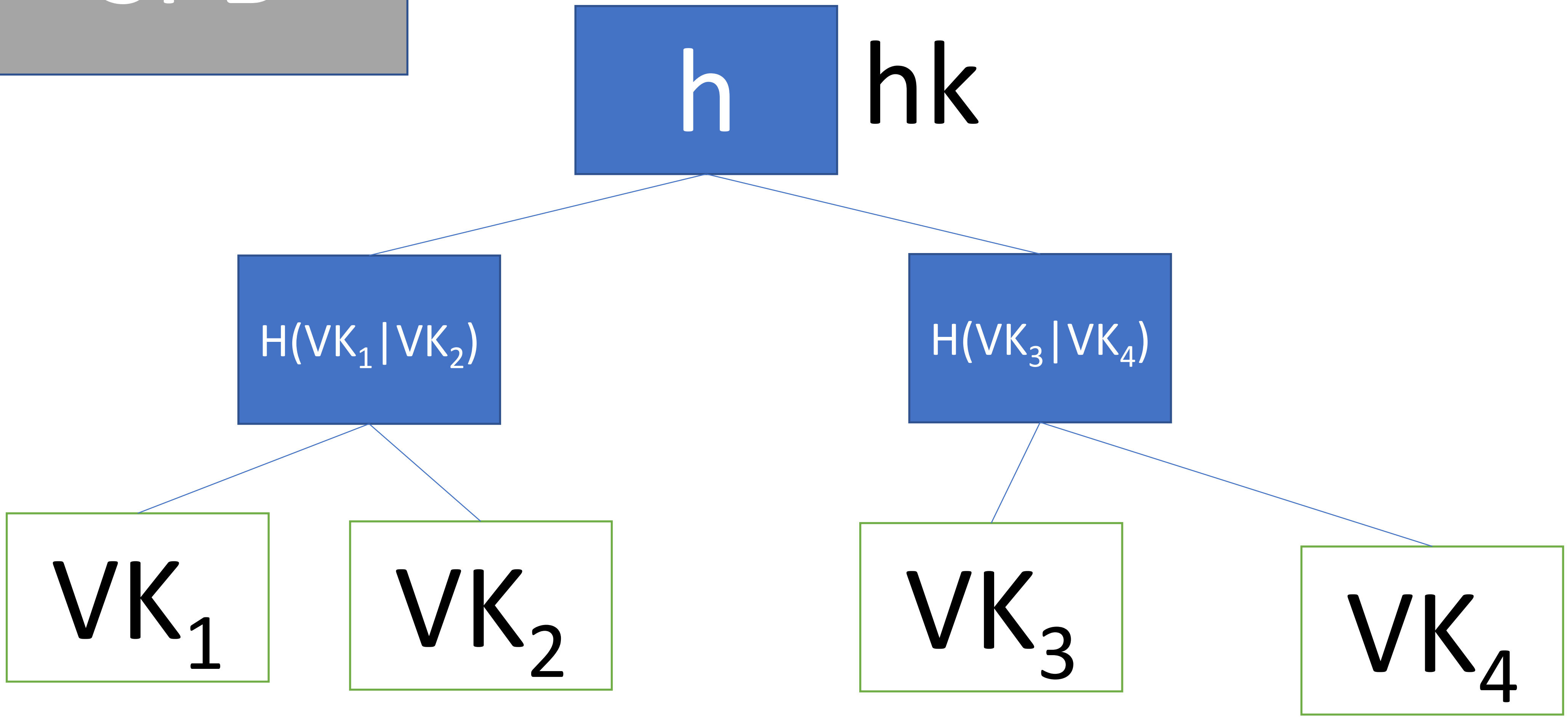
Public Key Encryption

- $ct \leftarrow PKE.Enc(pk_A, input)$
- $ct \leftarrow PKE.Enc(pk_B, input)$

- Key-privacy means you can’t tell from whom the encryption is!

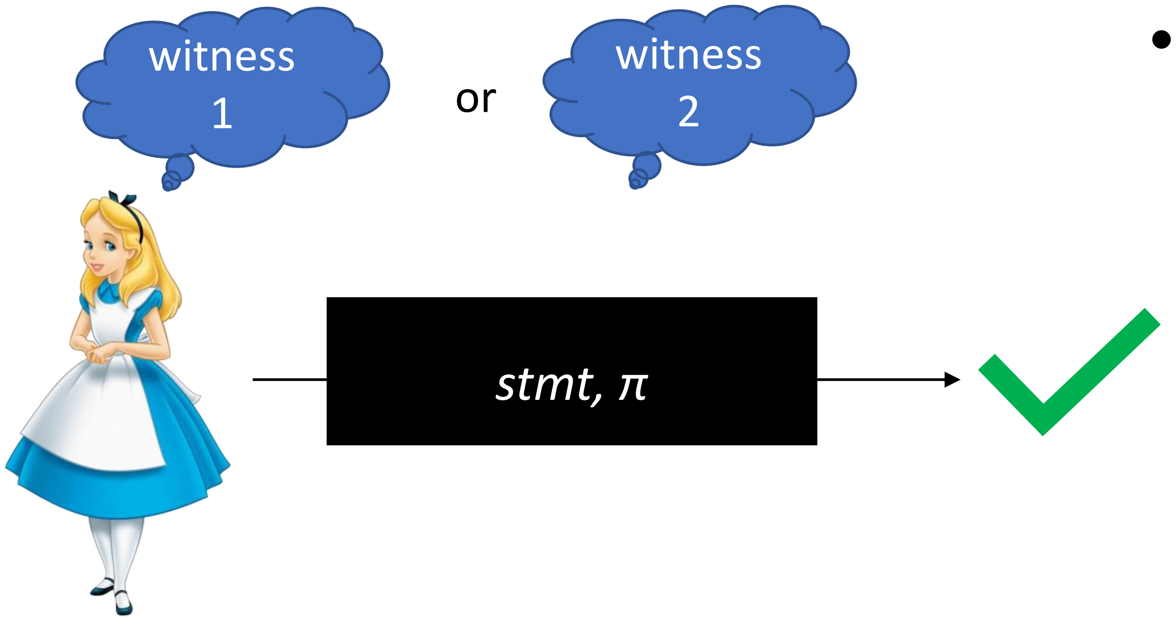
SPB

Somewhere Perfectly Binding



[HW15, OPWW15, BDH+19]

NIWI



witness 1

or

witness 2

stmt, π

- Verifier does not learn which witness the prover has in mind.
- NIWIs with perfect soundness: can’t prove a false statement.

PKE

VRF

Key Generation

Key Generation

PKE

VRF

$SK = VRF.sk$

$VK = VRF.vk, PKE.pk$

PKE

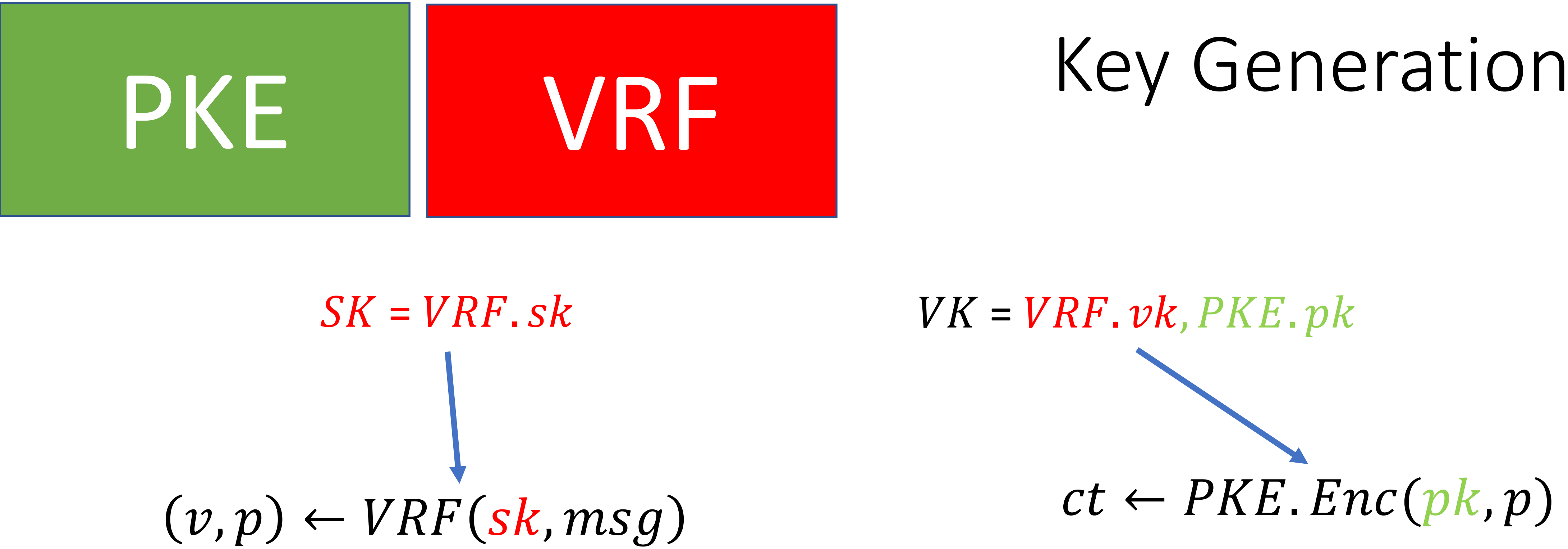
VRF

Key Generation

$$\begin{tikzcd} SK = VRF.sk \arrow[d] \\ (v, p) \leftarrow VRF(sk, msg) \end{tikzcd}$$

$VK = VRF.vk, PKE.pk$

Key Generation



PKE

VRF

$SK = VRF.sk$

$VK = VRF.vk, PKE.pk$

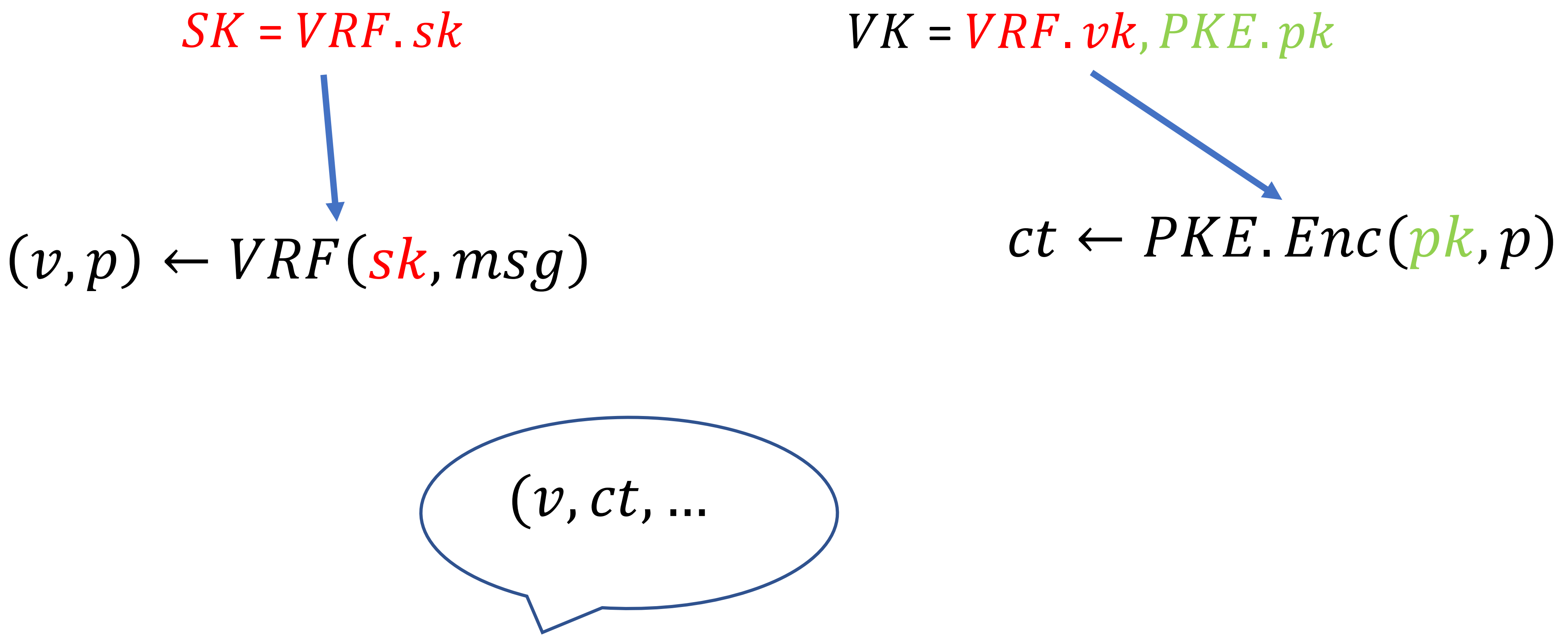
$(v, p) \leftarrow VRF(sk, msg)$

$ct \leftarrow PKE.Enc(pk, p)$

PKE

VRF

Key Generation



$SK = VRF.sk$

$VK = VRF.vk, PKE.pk$

$(v, p) \leftarrow VRF(sk, msg)$

$ct \leftarrow PKE.Enc(pk, p)$

$(v, ct, \ldots$

NIWI

Prove Membership

π: NIWI proof that

NIWI

Prove Membership

π: NIWI proof that

- ct encrypts a valid VRF proof p under some vk and
- The proof p verifies for v under vk

NIWI

Prove Membership

π: NIWI proof that

- ct encrypts a valid VRF proof p under some vk and
- The proof p verifies for v under vk
- ∃ an SPB opening showing vk is consistent with h, hk

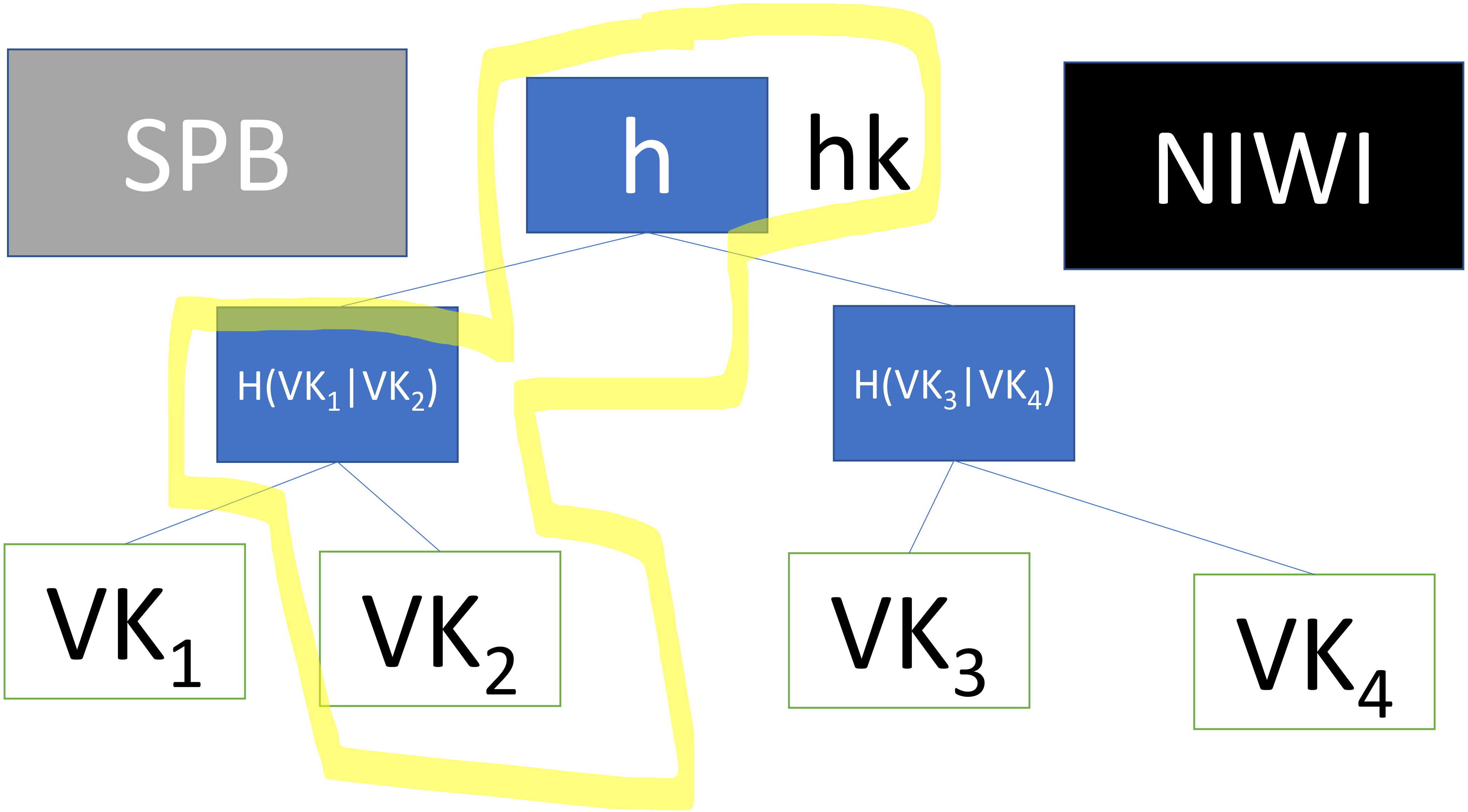
NIWI

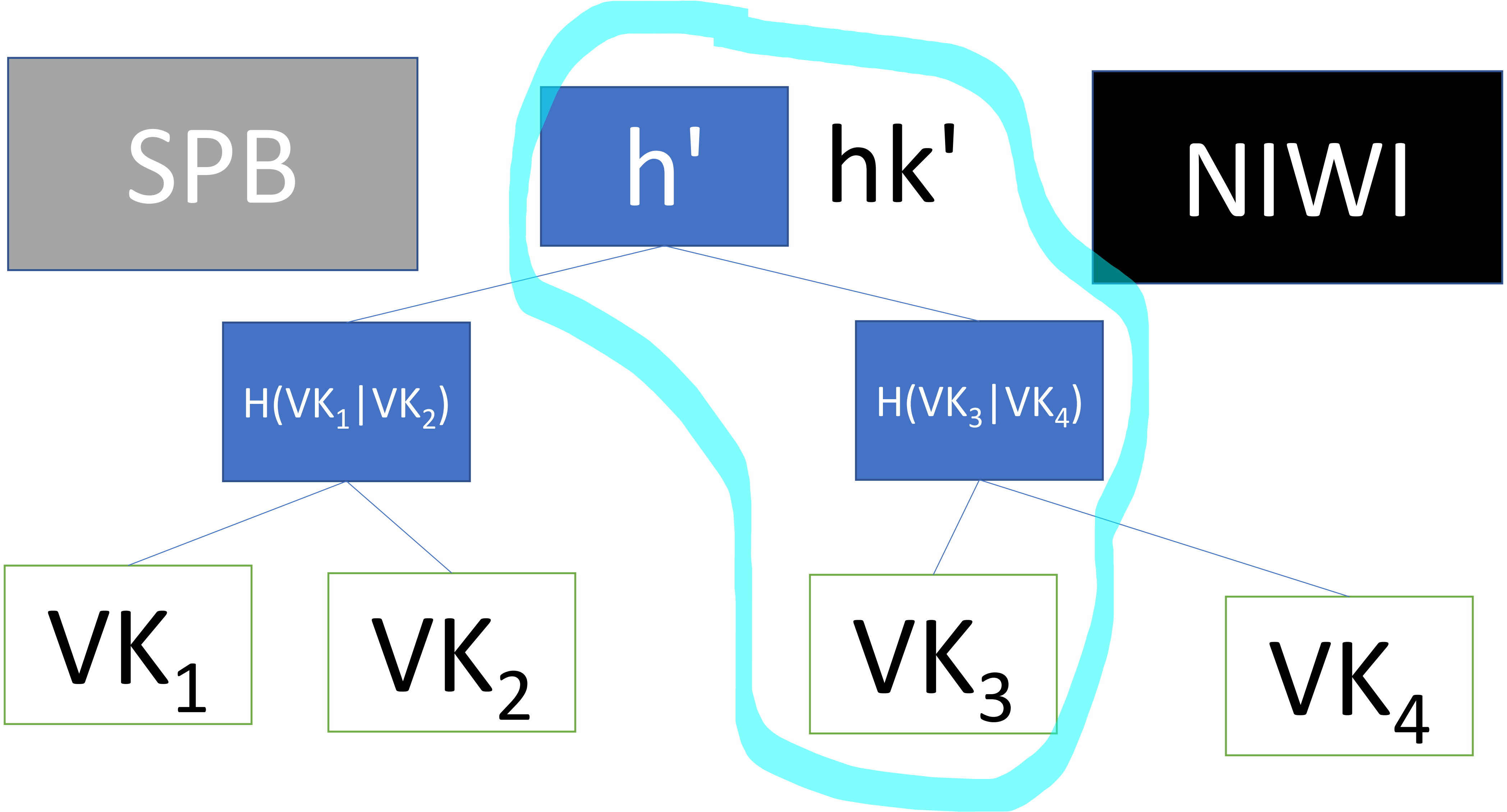
Prove Membership

π: NIWI proof that

- ct encrypts a valid VRF proof p under some vk and
- The proof p verifies for v under vk
- ∃ an SPB opening showing vk is consistent with h, hk

$(v, ct, h, hk, \dots \pi)$


SPB
h
hk
NIWI
H(VK$_1$|VK$_2$)
H(VK$_3$|VK$_4$)
VK$_1$
VK$_2$
VK$_3$
VK$_4$


SPB
h'
hk'
NIWI
H(VK$_1$|VK$_2$)
H(VK$_3$|VK$_4$)
VK$_1$
VK$_2$
VK$_3$
VK$_4$

NIWI

Prove Membership using an OR

π: NIWI proof that

- ct encrypts a valid VRF proof p under some vk_i and
- The proof p verifies for v under vk_i
- $\exists$ an SPB opening showing $vk_i$is consistent with h, hk

NIWI

Prove Membership using an OR

π: NIWI proof that

- ct encrypts a valid VRF proof p under some vk_i and
- The proof p verifies for v under vk_i
- $\exists$ an SPB opening showing $vk_i$is consistent with h, hk

Same proof but for vk_j

NIWI

Prove Membership using an OR

π: NIWI proof that

- ct encrypts a valid VRF proof p under some vk_i and
- The proof p verifies for v under vk_i
- ∃ an SPB opening showing $vk_i$is consistent with h, hk

Same proof but for vk_j

$(v, ct, h, hk, v', ct', h', hk', \pi)$

NIWI

Prove Membership using an OR

π: NIWI proof

$$R|\, vk_i \vee R|\, vk_j$$

$(v, ct, h, hk, v', ct', h', hk', \pi)$

Anonymity Proof Swaps between Signers

- Two signers, both alike in dignity. In fair Verona.
- Swap between signer i and signer j

Anonymity Proof Swaps between Signers

- Two signers, both alike in dignity. In fair Verona.
- Swap between signer i and signer j

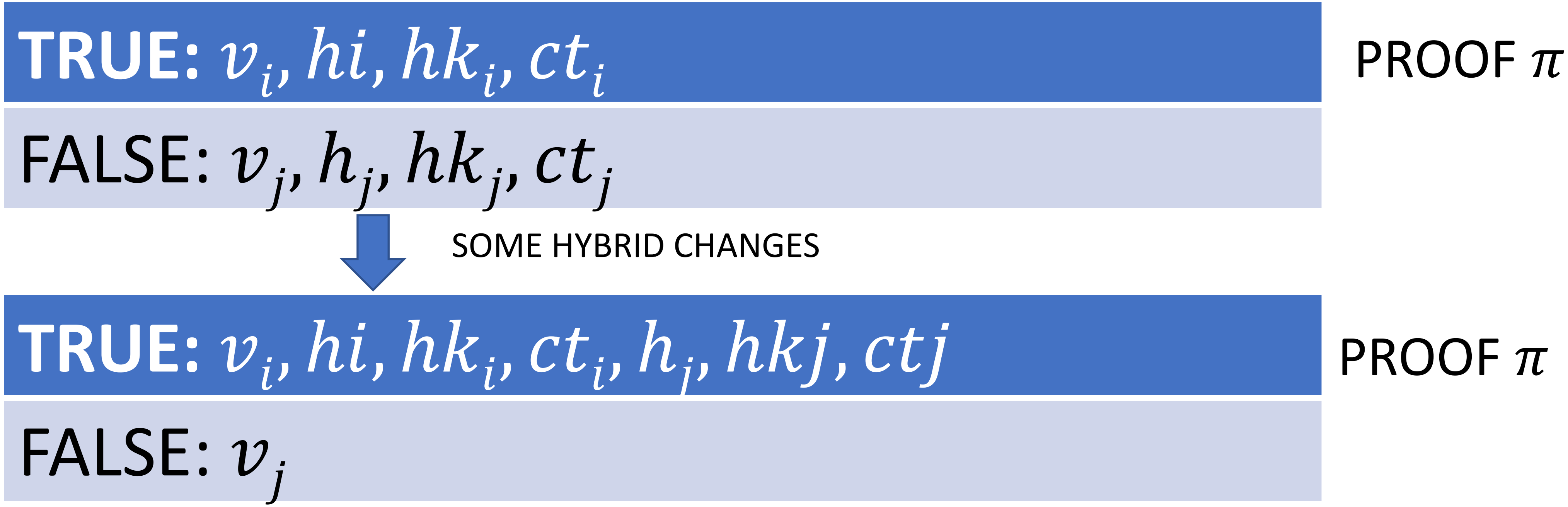
TRUE: v_i, hi, hk_i, ct_i

FALSE: v_j, h_j, hk_j, ct_j

PROOF π

Anonymity Proof Swaps between Signers

- Two signers, both alike in dignity. In fair Verona.
- Swap between signer i and signer j



TRUE: v_i, hi, hk_i, ct_i PROOF π

FALSE: v_j, h_j, hk_j, ct_j

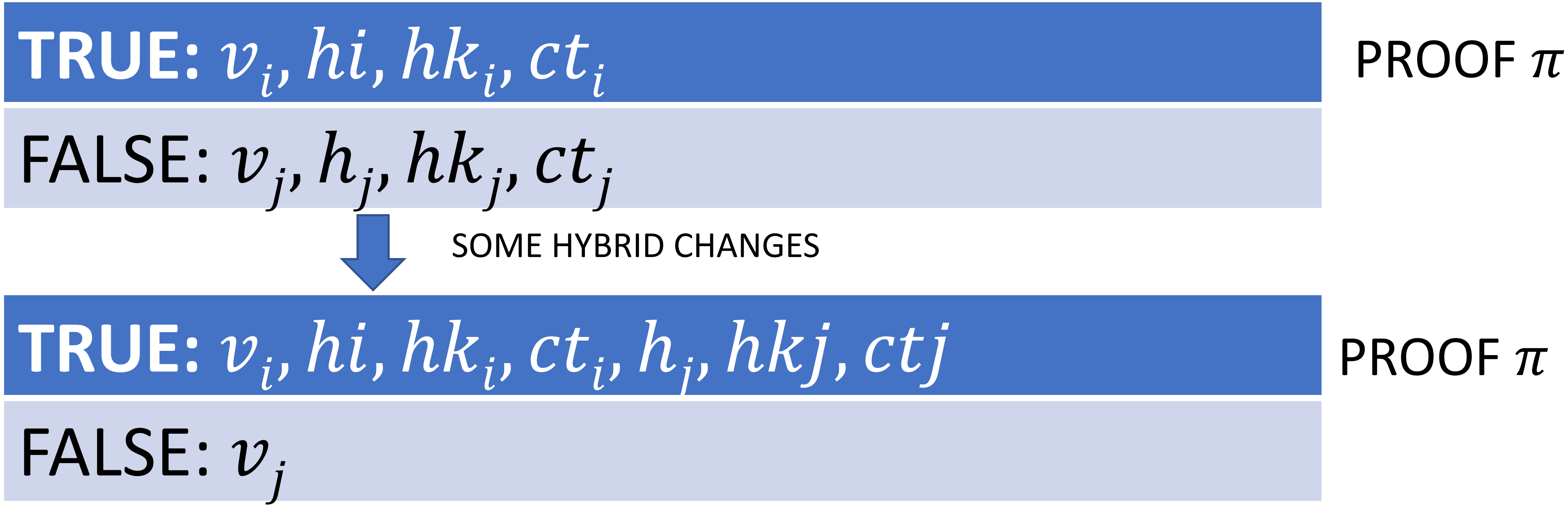
SOME HYBRID CHANGES

TRUE: $v_i, hi, hk_i, ct_i, h_j, hkj, ctj$ PROOF π

FALSE: v_j

Anonymity Proof Swaps between Signers

- Two signers, both alike in dignity. In fair Verona.
- Swap between signer i and signer j



TRUE: v_i, hi, hk_i, ct_i

FALSE: v_j, h_j, hk_j, ct_j

PROOF π

SOME HYBRID CHANGES

TRUE: $v_i, hi, hk_i, ct_i, h_j, hkj, ctj$

FALSE: v_j

PROOF π

If both branches are true…

TRUE: v_i, hi, hk_i, ct_i

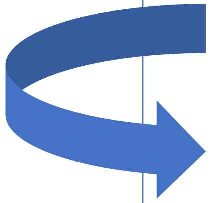
FALSE: v_j, h_j, hk_j, ct_j

$R \mid vk_i$	$R \mid vk_j$	$R \mid vk_i \vee R \mid vk_j$
T	F	T
T	T	T

If both branches are true…

TRUE: v_i, hi, hk_i, ct_i

FALSE: v_j, h_j, hk_j, ct_j



SOME HYBRID CHANGES

$R\|\ vk_i$	$R\|\ vk_j$	$R\|\ vk_i \vee R\|\ vk_j$
T	F	T
T	T	T

If both branches are true…

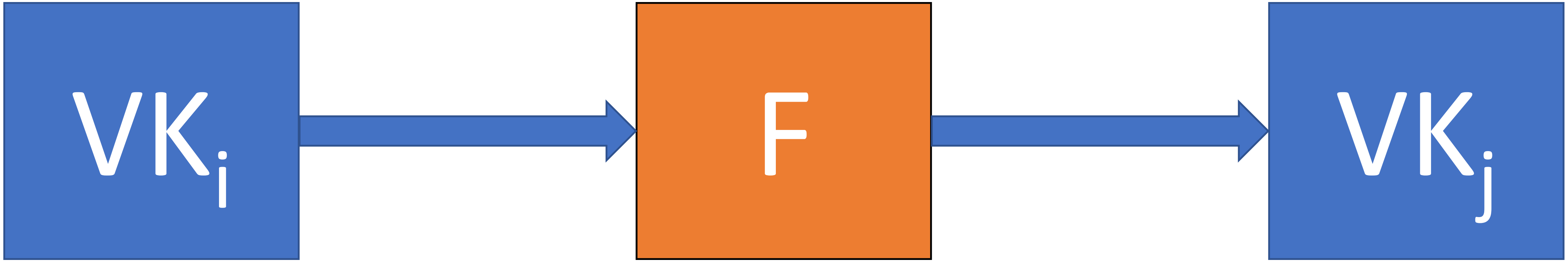
TRUE: v_i, hi, hk_i, ct_i

FALSE: v_j, h_j, hk_j, ct_j

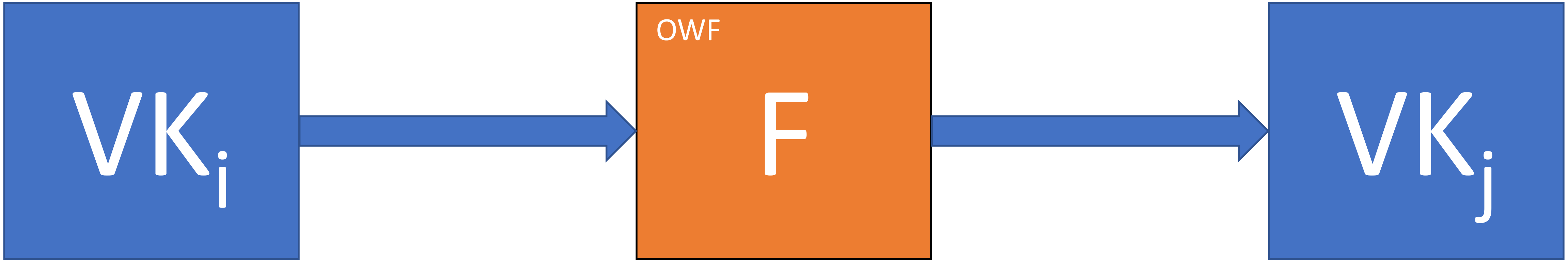
$R\mid vk_i$	$R\mid vk_j$	$R\mid vk_i \vee R\mid vk_j$
T	F	T
T	T	T
F	F	F

Anonymity

Anonymity

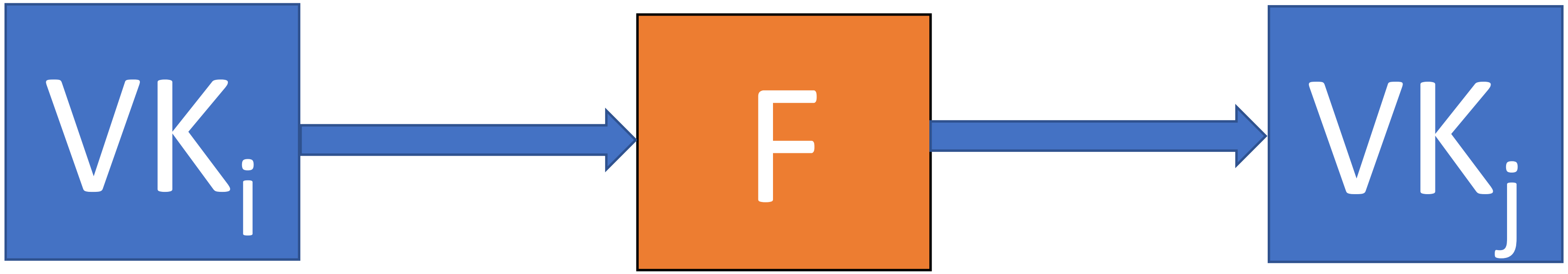


Anonymity



$$R_F:$$
$$F(sk_i) = skj$$

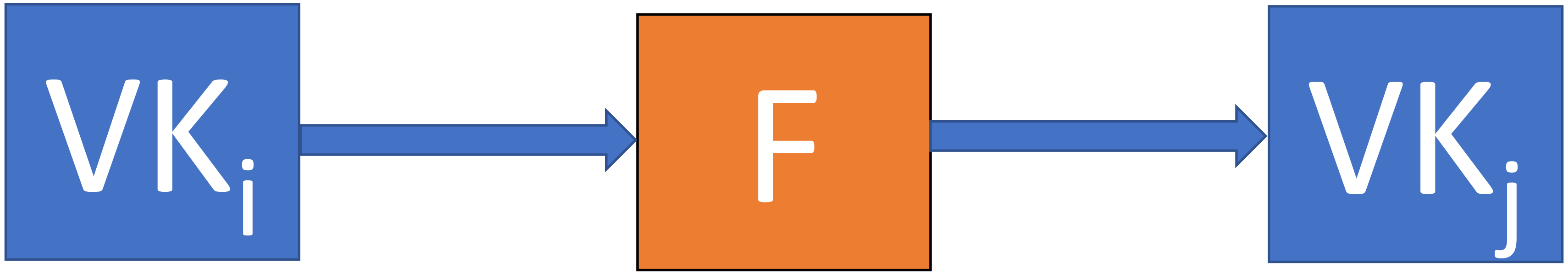
Anonymity Hybrids Via a OWF



VK_i → F → VK_j

$R\|\ vk_i$	$R\|\ vk_j$	R_F	$R\|\ vk_i \vee R\|\ vk_j \vee R_F$
T	F	F	T
F	T	F	T

Anonymity Hybrids Via a OWF



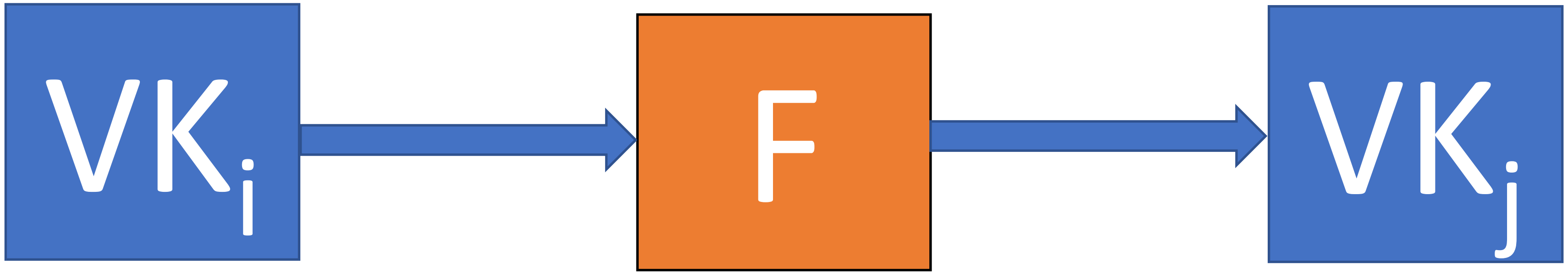
$R\vert\, vk_i$	$R\vert\, vk_j$	R_F	$R\vert\, vk_i \vee R\vert\, vk_j \vee R_F$
T	F	F	T
F	T	F	T

Real signatures from signer i

$$F(sk_i) = skj$$

Real signatures from signer j

Anonymity Hybrids Via a OWF



VK_i

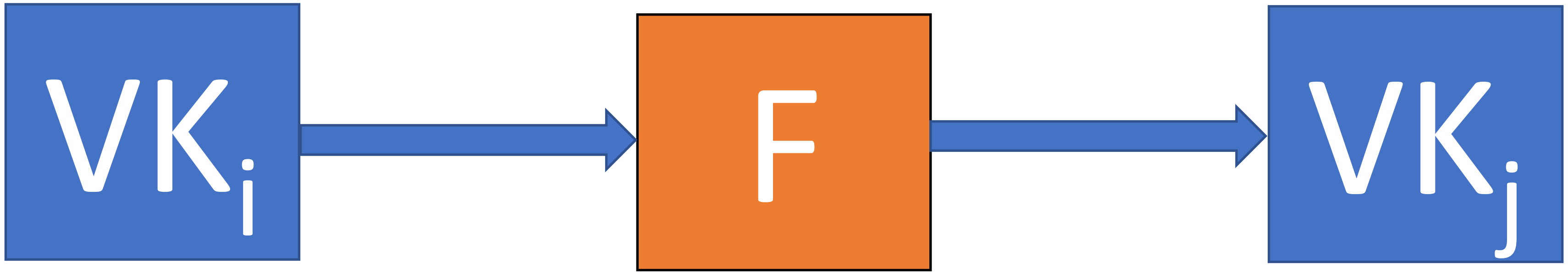
F

VK_j

$R\|\, vk_i$	$R\|\, vk_j$	R_F	$R\|\, vk_i \vee R\|\, vk_j \vee R_F$
T	F	F	T
F	F	T	T
F	T	F	T

$$F(sk_i) = skj$$

Anonymity Hybrids Via a OWF



$R\|\ vk_i$	$R\|\ vk_j$	R_F	$R\|\ vk_i \vee R\|\ vk_j \vee R_F$
T	F	F	T
F	F	T	T
F	T	T	T
F	T	F	T

But Now Unforgeability Precludes OReg!

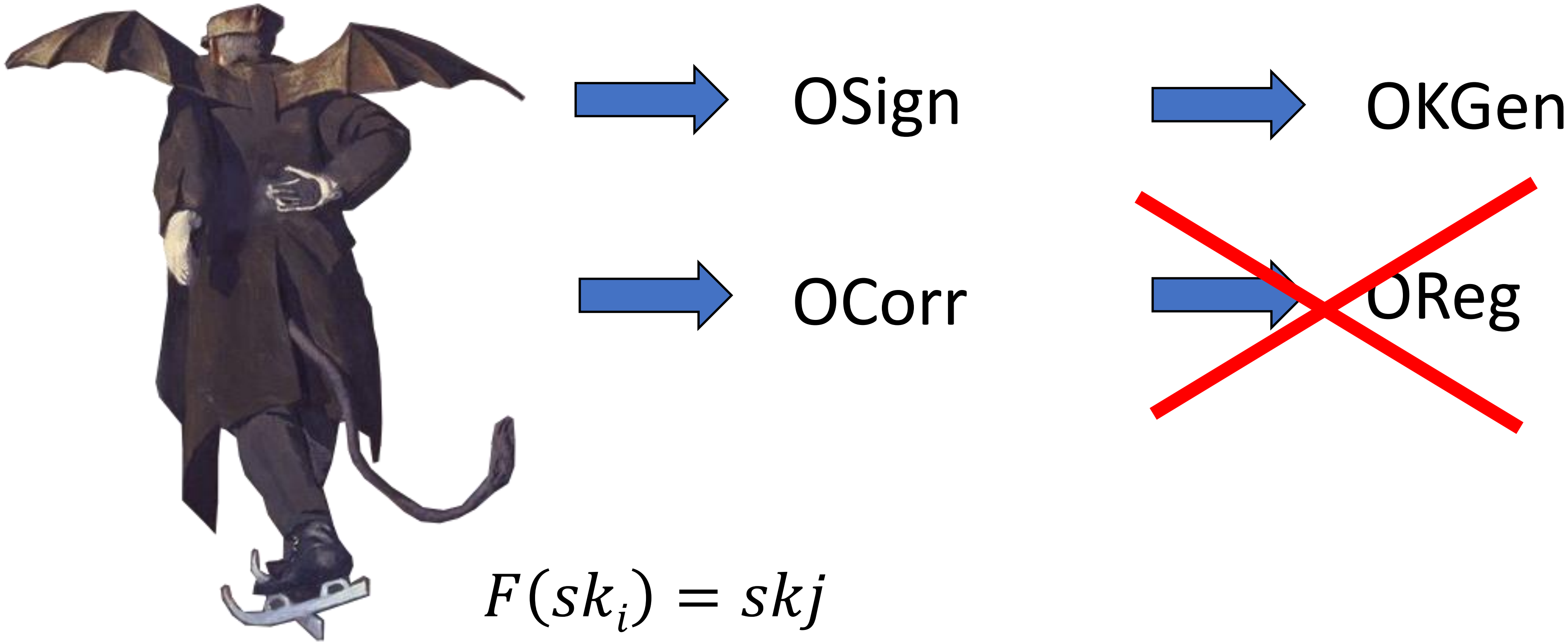
→ OSign

→ OKGen

→ OCorr

→ OReg

But Now Unforgeability Precludes OReg!



OSign

OKGen

OCorr

~~OReg~~

$F(sk_i) = skj$

Why does it matter?

- Gives feasibility, even with weakened unforgeability.
- Resultant research question:

Does there exist a **compact** thring with **malicious registration** in the **plain model**?

THANKS FOR YOUR ATTENTION!

ahaque3@ncsu.edu

IACR ePrint: 2020/683

NC STATE

Cyber Security for Europe

AIT AUSTRIAN INSTITUTE OF TECHNOLOGY