CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

When is an AKE Protocol Secure against State Reveals?

Revisiting Models, Protocols, and Transforms

Subham Das[1]

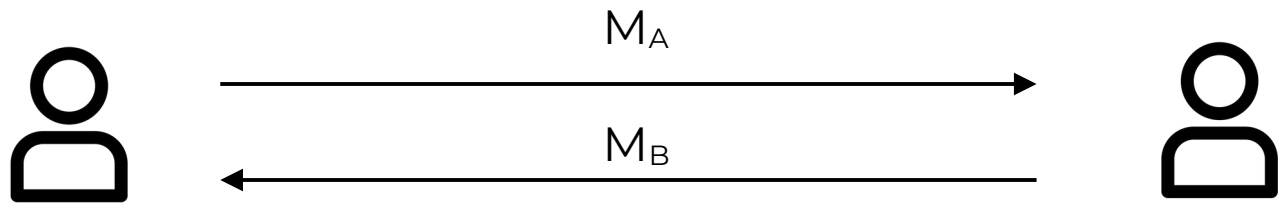
joint work with Hans Heum[2], Xiangyu Liu[1] and Doreen Riepel[1]

[1] CISPA, [2] Simula UiB

ProTeCS 2026

Authenticated Key Exchange

Authenticated Key Exchange



Authenticated Key Exchange

Authenticated Key Exchange

Security Goals:

- Key Indistinguishability
- Authentication between users

Authenticated Key Exchange

Security Goals:

- Key Indistinguishability
- Authentication between users

Considered Protocols:

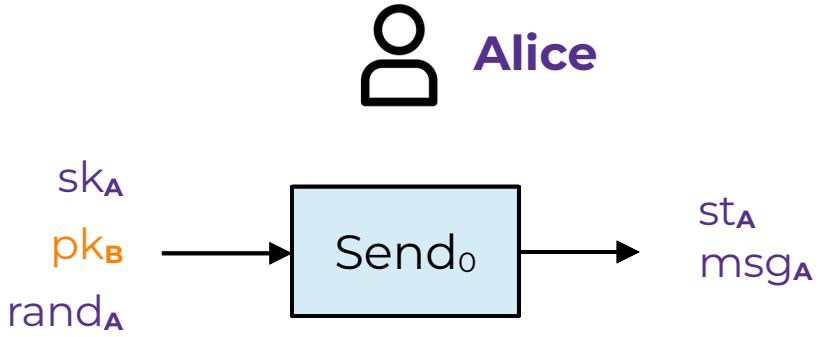
- 2-round protocols
- Implicitly authenticated protocols

AKE: A little syntax

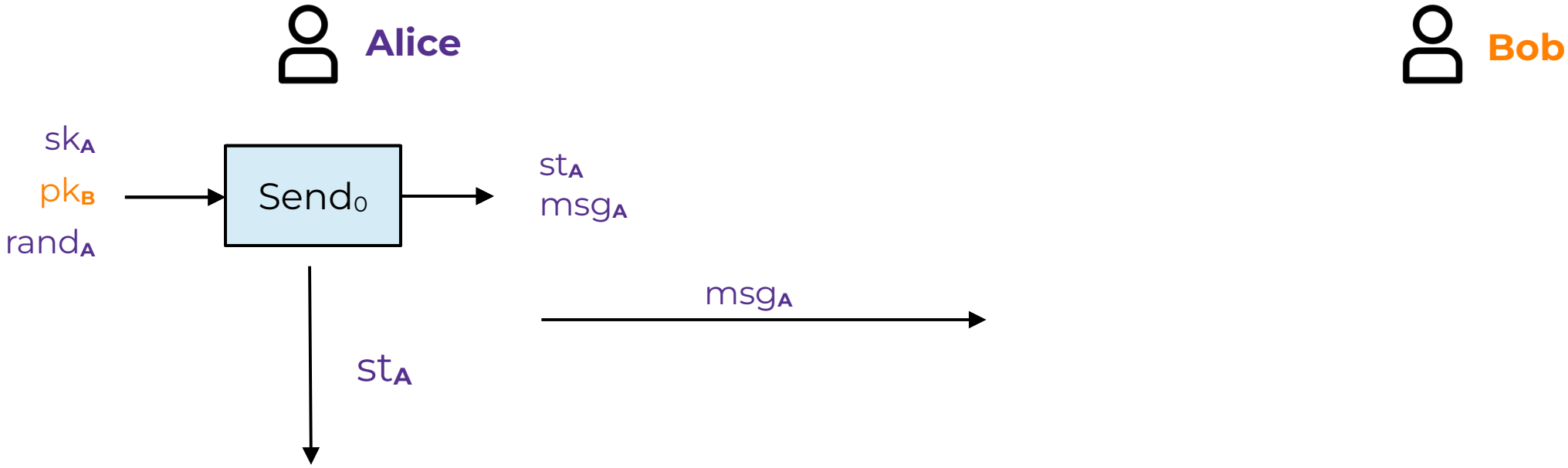
Alice

Bob

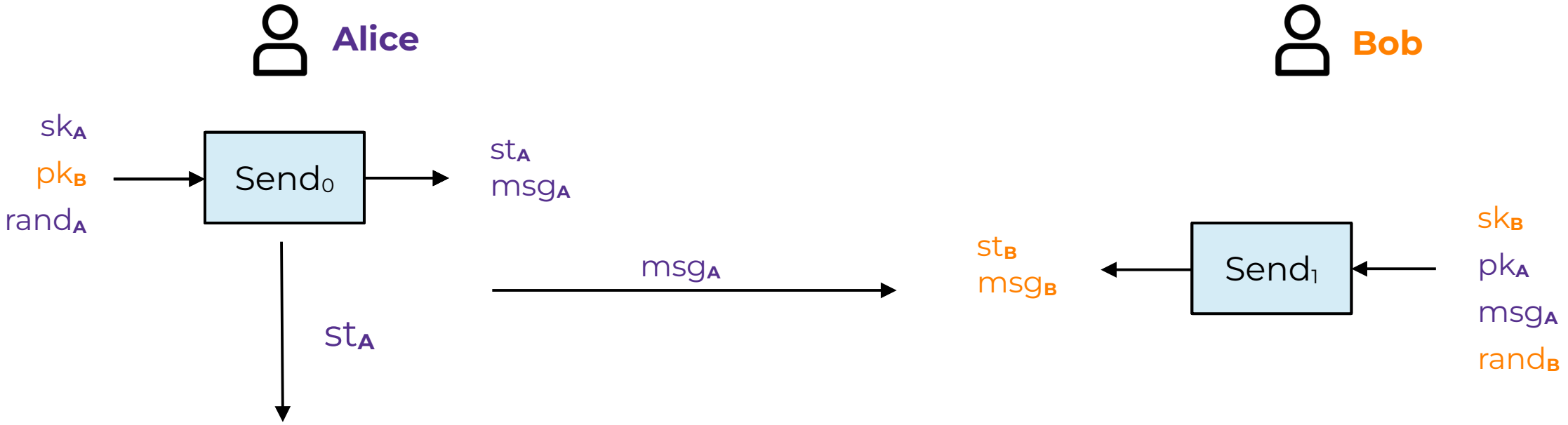
AKE: A little syntax



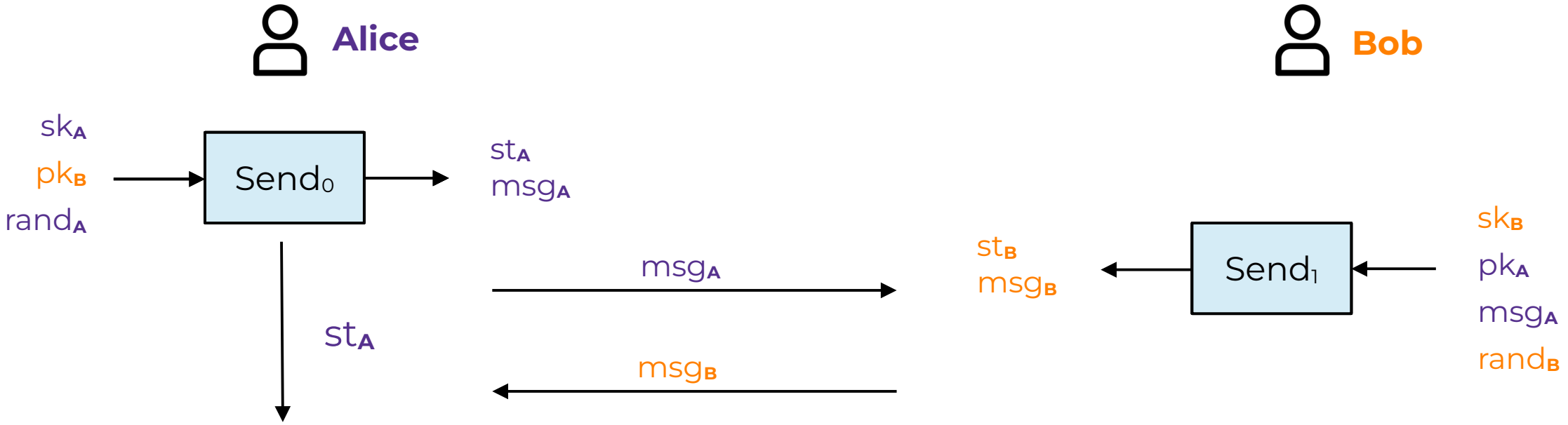
AKE: A little syntax



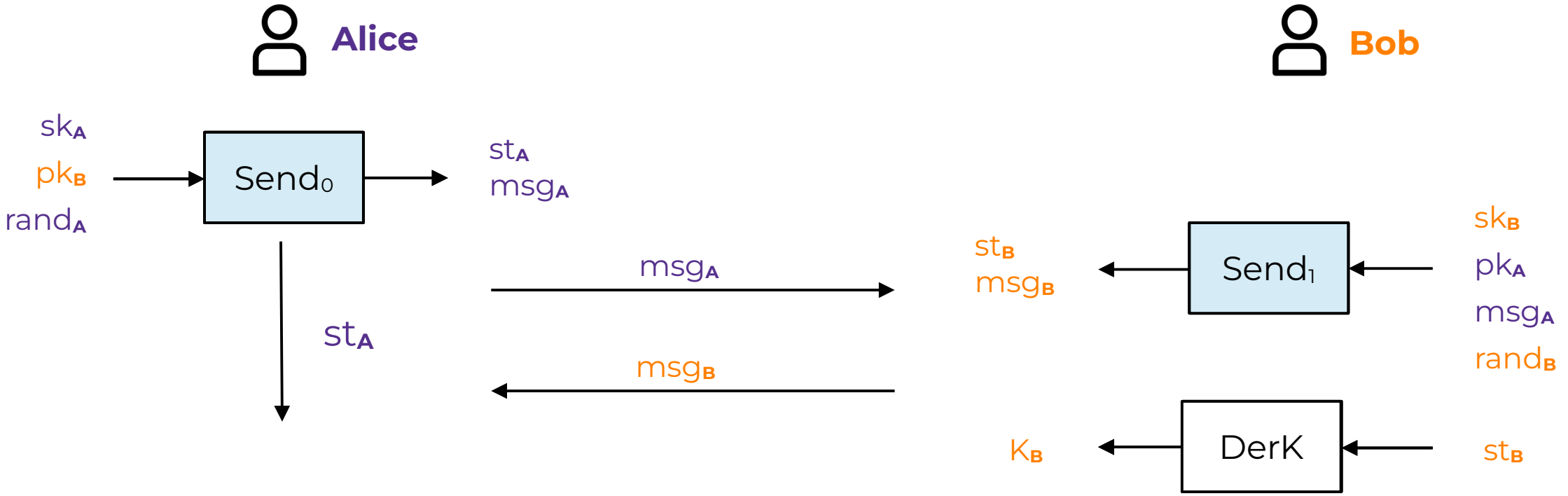
AKE: A little syntax



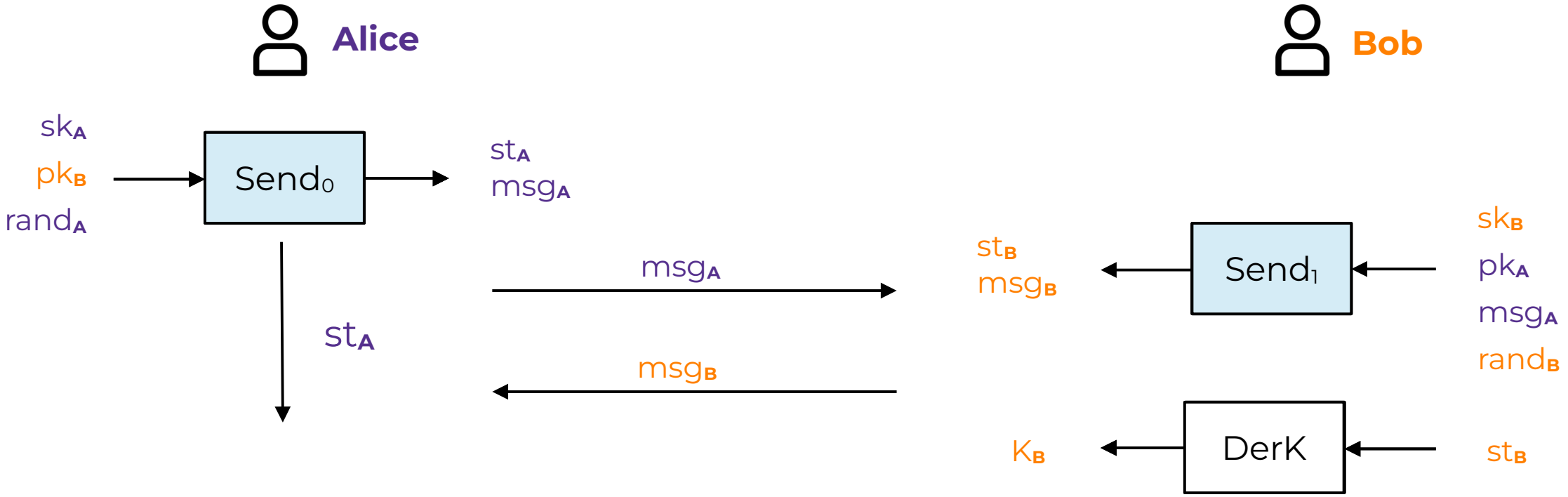
AKE: A little syntax



AKE: A little syntax

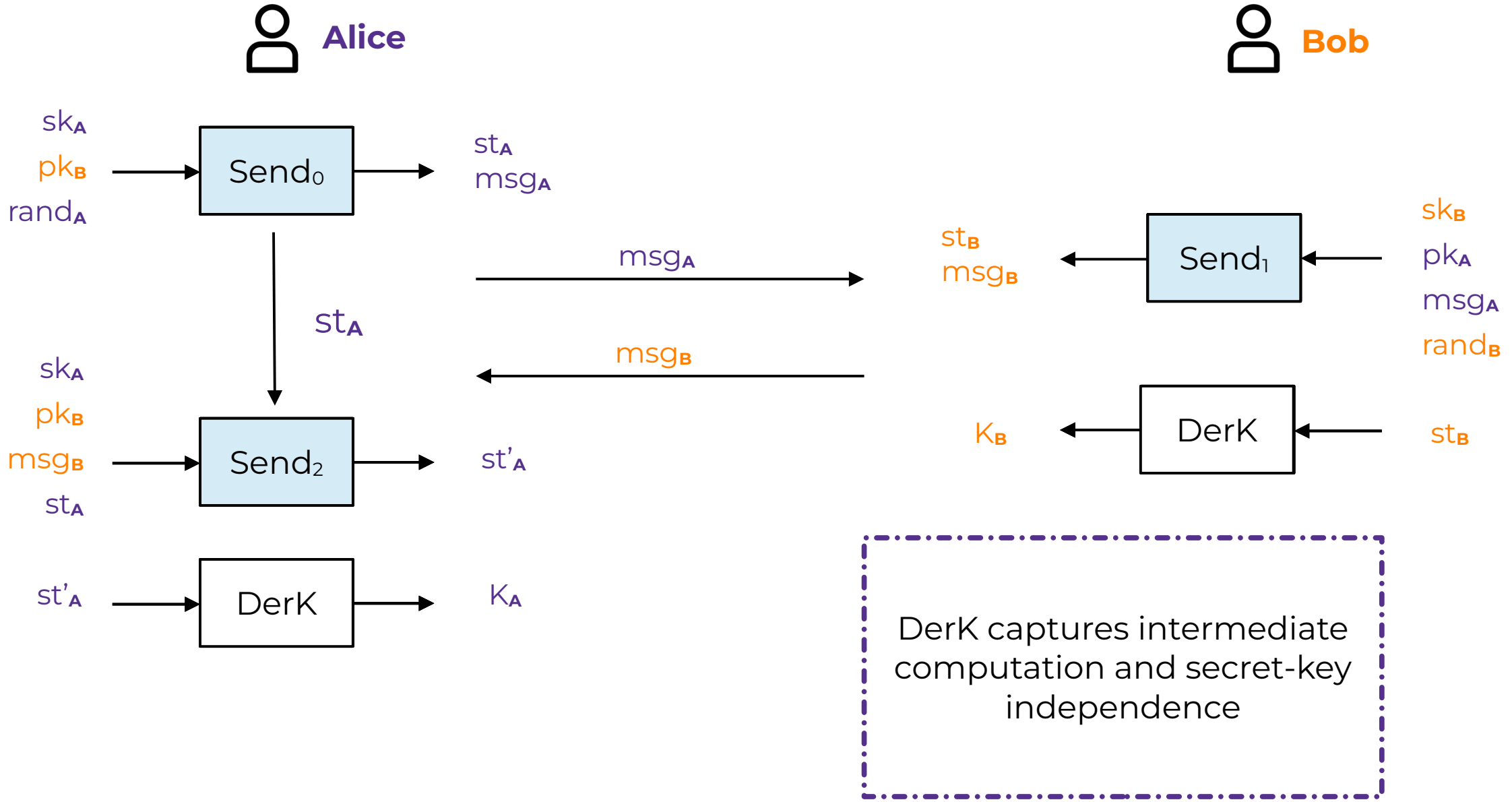


AKE: A little syntax



DerK captures intermediate computation and secret-key independence

AKE: A little syntax



DerK captures intermediate computation and secret-key independence

Security model

Security model

Security is modeled as a game between Challenger and Adversary.

The goal of the adversary is to distinguish between real or random keys.

Security model

Security is modeled as a game between Challenger and Adversary.
The goal of the adversary is to distinguish between real or random keys.

Adversarial Capabilities

Security model

Security is modeled as a game between Challenger and Adversary.

The goal of the adversary is to distinguish between real or random keys.

Adversarial Capabilities

- Controls the network and can modify/drop messages

Security model

Security is modeled as a game between Challenger and Adversary.

The goal of the adversary is to distinguish between real or random keys.

Adversarial Capabilities

- Controls the network and can modify/drop messages
- Adaptively corrupts long term keys

Security model

Security is modeled as a game between Challenger and Adversary.

The goal of the adversary is to distinguish between real or random keys.

Adversarial Capabilities

- Controls the network and can modify/drop messages
- Adaptively corrupts long term keys
- Registers corrupted parties

Security model

Security is modeled as a game between Challenger and Adversary.

The goal of the adversary is to distinguish between real or random keys.

Adversarial Capabilities

- Controls the network and can modify/drop messages
- Adaptively corrupts long term keys
- Registers corrupted parties
- Reveals session keys

Security model

Security is modeled as a game between Challenger and Adversary.

The goal of the adversary is to distinguish between real or random keys.

Adversarial Capabilities

- Controls the network and can modify/drop messages
- Adaptively corrupts long term keys
- Registers corrupted parties
- Reveals session keys
- Access to **state-reveal oracles** depending on allowed types of state-reveals

Security model

Security is modeled as a game between Challenger and Adversary.

The goal of the adversary is to distinguish between real or random keys.

Adversarial Capabilities

- Controls the network and can modify/drop messages
- Adaptively corrupts long term keys
- Registers corrupted parties
- Reveals session keys
- Access to **state-reveal oracles** depending on allowed types of state-reveals

Partnering

Security model

Security is modeled as a game between Challenger and Adversary.

The goal of the adversary is to distinguish between real or random keys.

Adversarial Capabilities

- Controls the network and can modify/drop messages
- Adaptively corrupts long term keys
- Registers corrupted parties
- Reveals session keys
- Access to **state-reveal oracles** depending on allowed types of state-reveals

Partnering

- Two sessions are fully partnered if they have “matching conversations”

Security model

Security is modeled as a game between Challenger and Adversary.

The goal of the adversary is to distinguish between real or random keys.

Adversarial Capabilities

- Controls the network and can modify/drop messages
- Adaptively corrupts long term keys
- Registers corrupted parties
- Reveals session keys
- Access to **state-reveal oracles** depending on allowed types of state-reveals

Partnering

- Two sessions are fully partnered if they have “matching conversations”
- **Independent sessions** are those sessions which are not partnered to the other

Security model: State Reveals

Security model: State Reveals

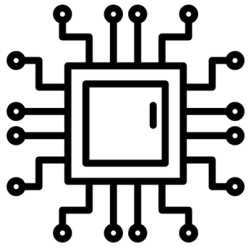
State: Everything which needs to be stored by a session to complete the protocol execution.

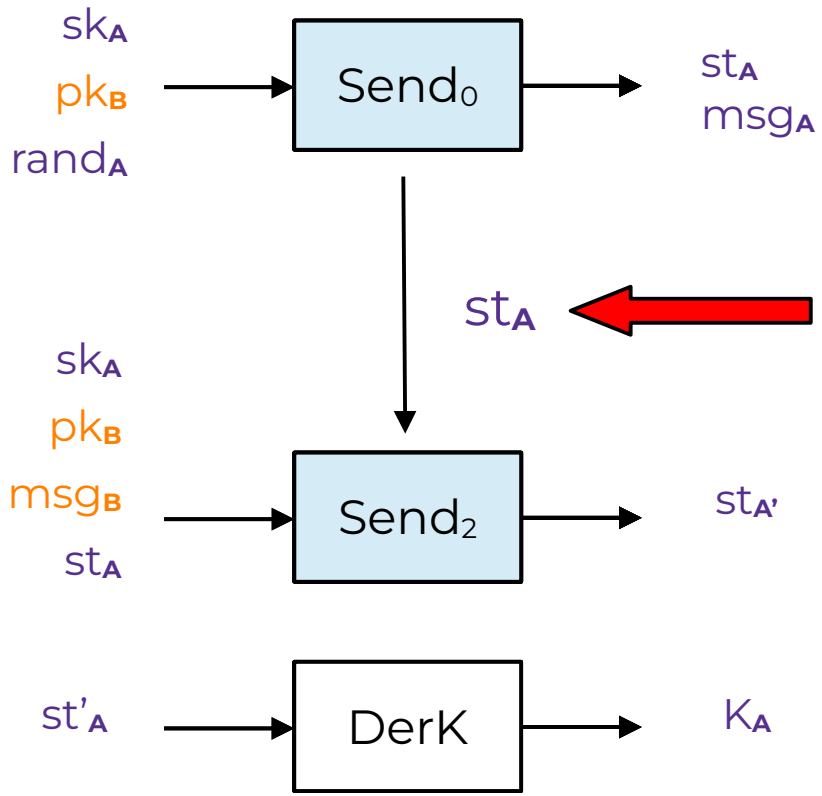
Security model: State Reveals

State: Everything which needs to be stored by a session to complete the protocol execution.

Ordinary State Reveal

- Outputs the entire state of a given session
- CK model [CK01] HMQV proven secure



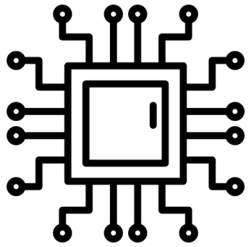


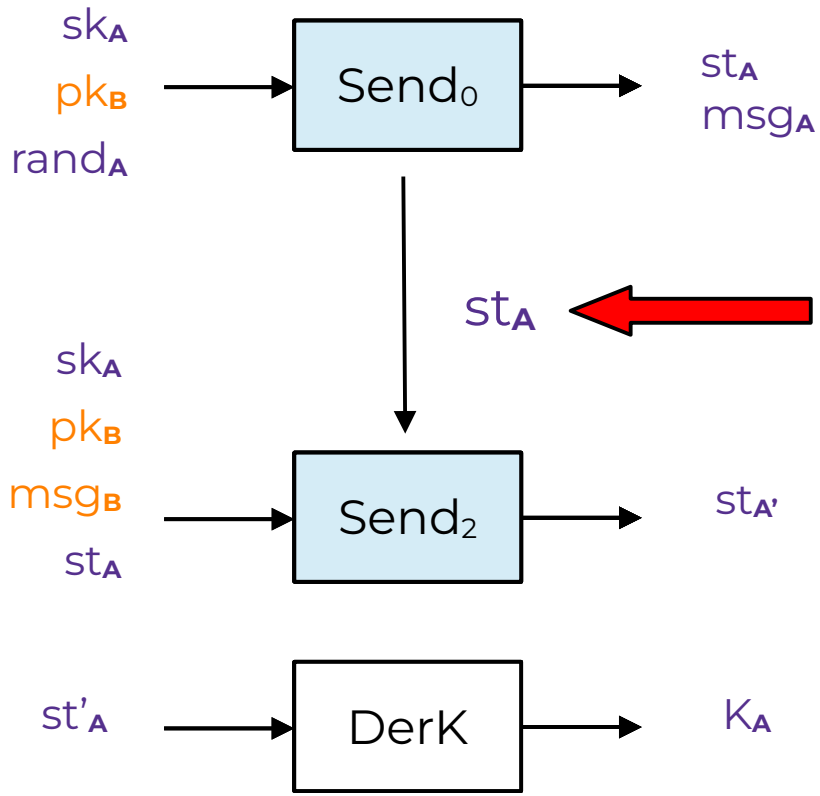
Security model: State Reveals

State: Everything which needs to be stored by a session to complete the protocol execution.

Ordinary State Reveal

- Outputs the entire state of a given session
- CK model [CK01] HMQV proven secure





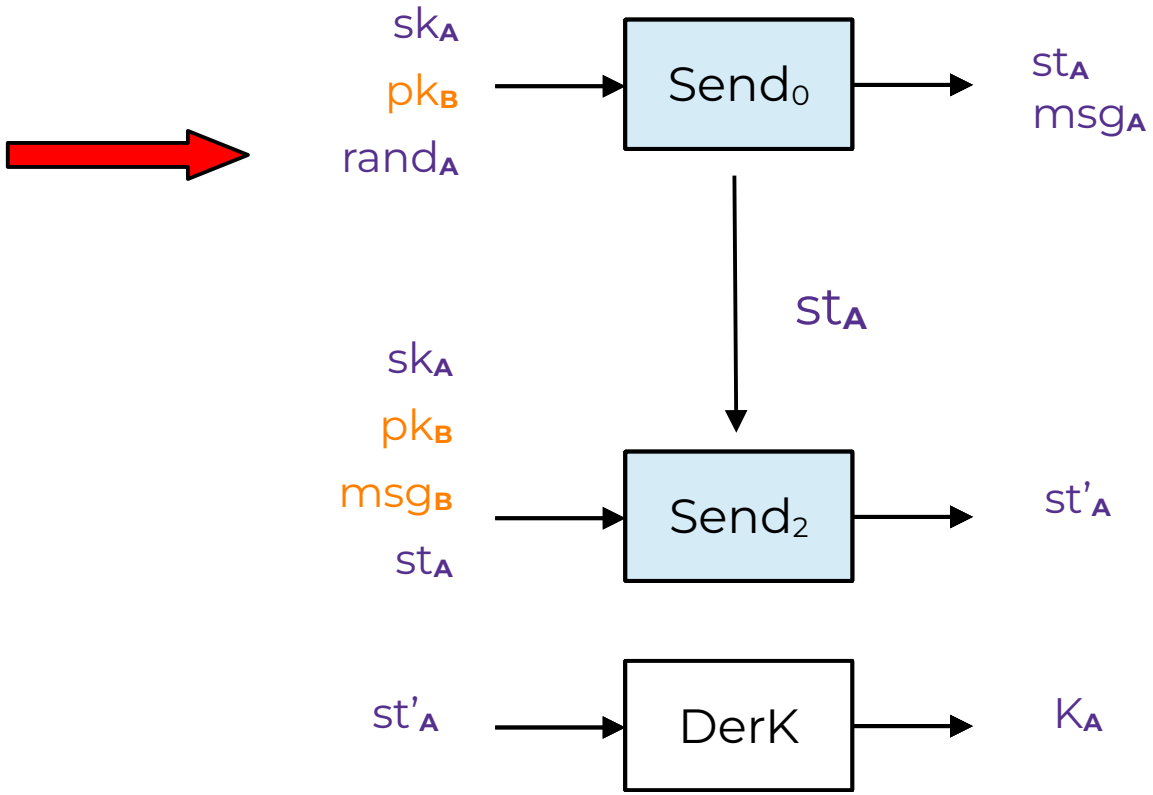
(allowed only for Alice)

Security model: State Reveals

State: Everything which needs to be stored by a session to complete the protocol execution.

Randomness Reveal

- Outputs the randomness used in the session
- eCK model [LL07] NAXOS proven secure

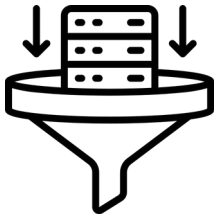


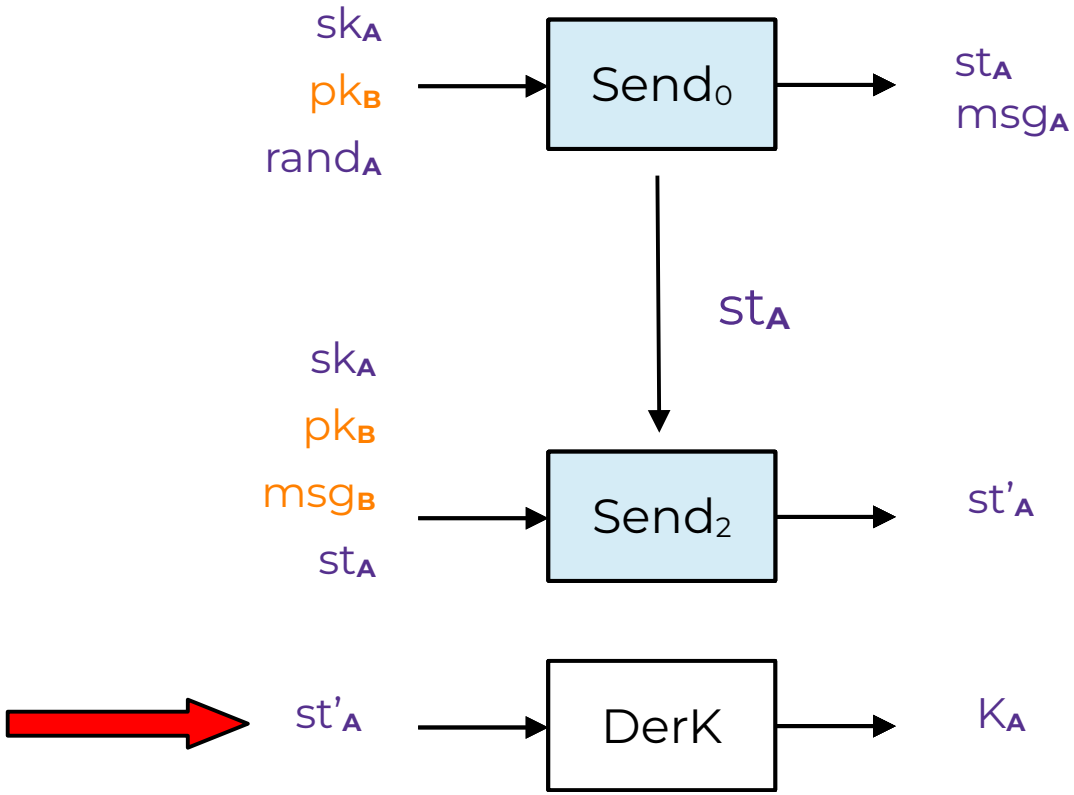
Security model: State Reveals

State: Everything which needs to be stored by a session to complete the protocol execution.

Preimage Reveal

- Outputs the input to the DerK function
- Only allowed for independent sessions
- seCK model [SEB10]





Desiderata

What do we desire?

Desiderata

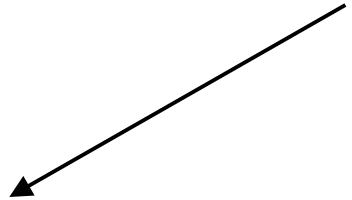
What do we desire?

AKE

Desiderata

What do we desire?

AKE



Secure against

Ordinary state reveal and
Randomness reveal on test
sessions

Desiderata

What do we desire?

AKE

Secure against

Ordinary state reveal and Randomness reveal on test sessions

Secure against

Preimage reveal **and** state-reveals of sessions independent of test sessions

Desiderata

What do we desire?

Implicit authentication

High efficiency

AKE

Secure against

Ordinary state reveal and Randomness reveal on test sessions

Secure against

Preimage reveal **and** state-reveals of sessions independent of test sessions

Desiderata

What do we desire?

Implicit authentication

High efficiency

AKE

Secure against

Ordinary state reveal and Randomness reveal on test sessions

Secure against

Preimage reveal **and** state-reveals of sessions independent of test sessions

Desiderata

What do we desire?

Implicit authentication

High efficiency

AKE

Secure against

Ordinary state reveal and Randomness reveal on test sessions

2R transform

Secure against

Preimage reveal **and** state-reveals of sessions independent of test sessions

The Randomness-Reinforcement (2R) transform

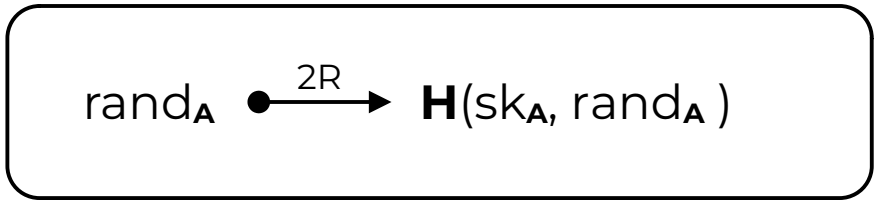
What does it do?

The Randomness-Reinforcement (2R) transform

What does it do?

Binds long term secret key with randomness, “reinforcing” it against randomness reveals

(for both Alice and Bob)



The Randomness-Reinforcement (2R) transform

What does it do?

Binds long term secret key with randomness, “reinforcing” it against randomness reveals

(for both Alice and Bob)

$\text{rand}_A \xrightarrow{2R} \mathbf{H}(\text{sk}_A, \text{rand}_A)$

“Hides” the initiator’s intermediate state by redefining it as the “plain” randomness .

(for only Alice)

$\text{st}_A \xrightarrow{2R} \text{st}_A := \text{rand}_A$

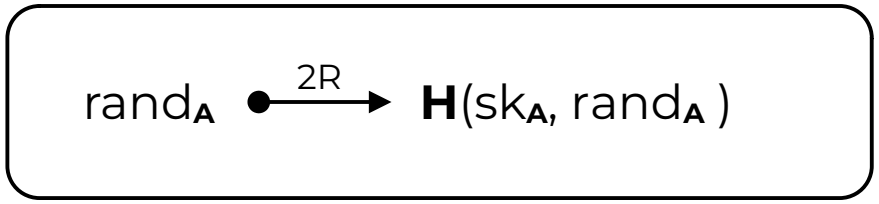
This makes state-reveal equivalent to a randomness reveal

The Randomness-Reinforcement (2R) transform

What does it do?

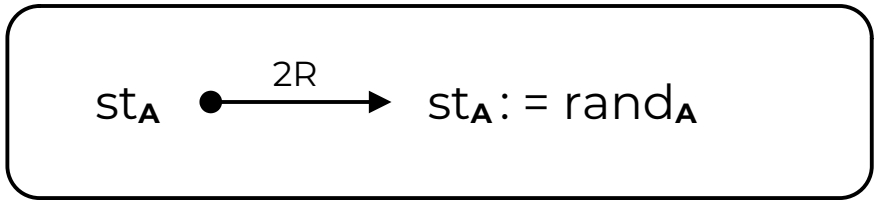
Binds long term secret key with randomness, “reinforcing” it against randomness reveals

(for both Alice and Bob)



“Hides” the initiator’s intermediate state by redefining it as the “plain” randomness .

(for only Alice)

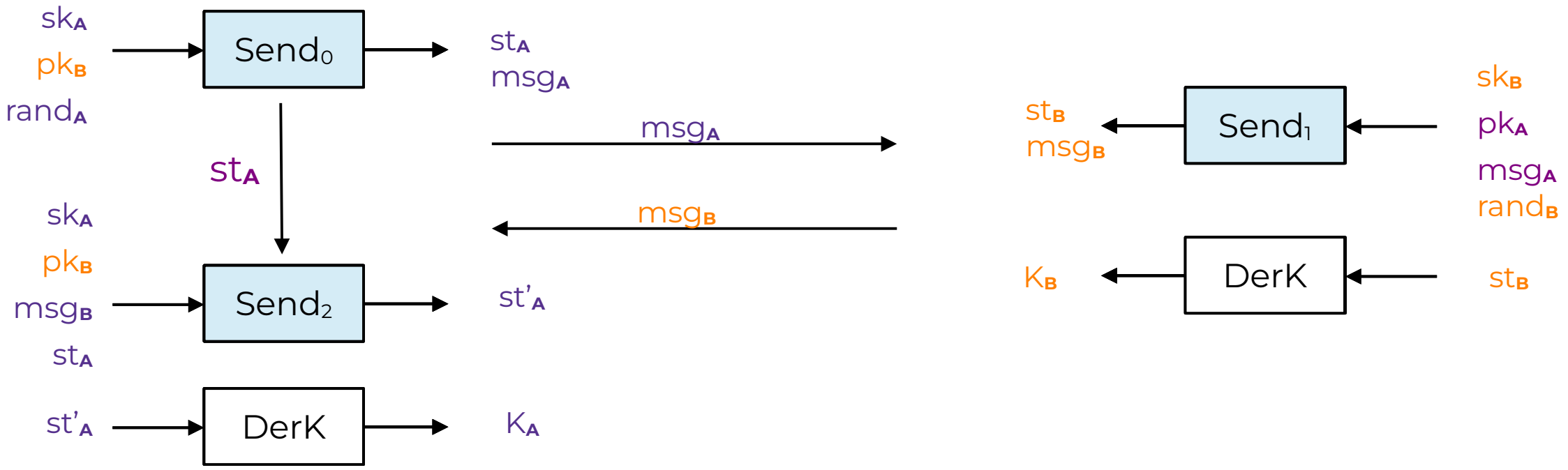


This makes state-reveal equivalent to a randomness reveal

Generalization of the NAXOS[1] trick !

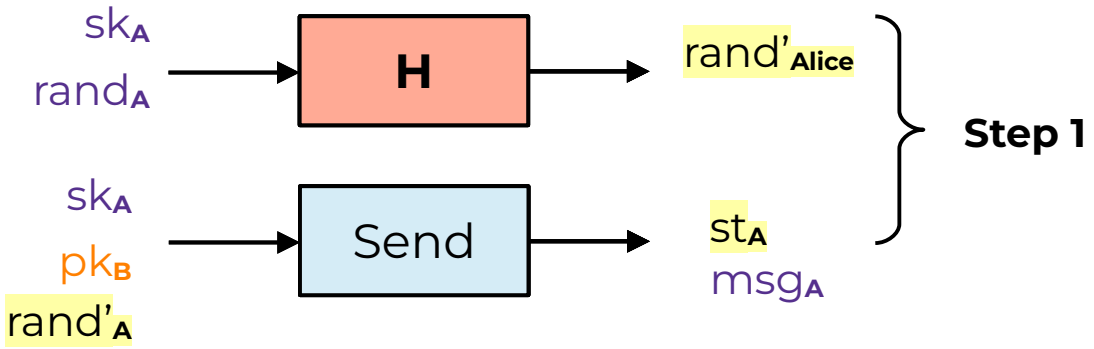
Brian A. LaMacchia, Kristin Lauter, and Anton Mityagin. Stronger security of authenticated key exchange, ProvSec 2007

The 2R transform

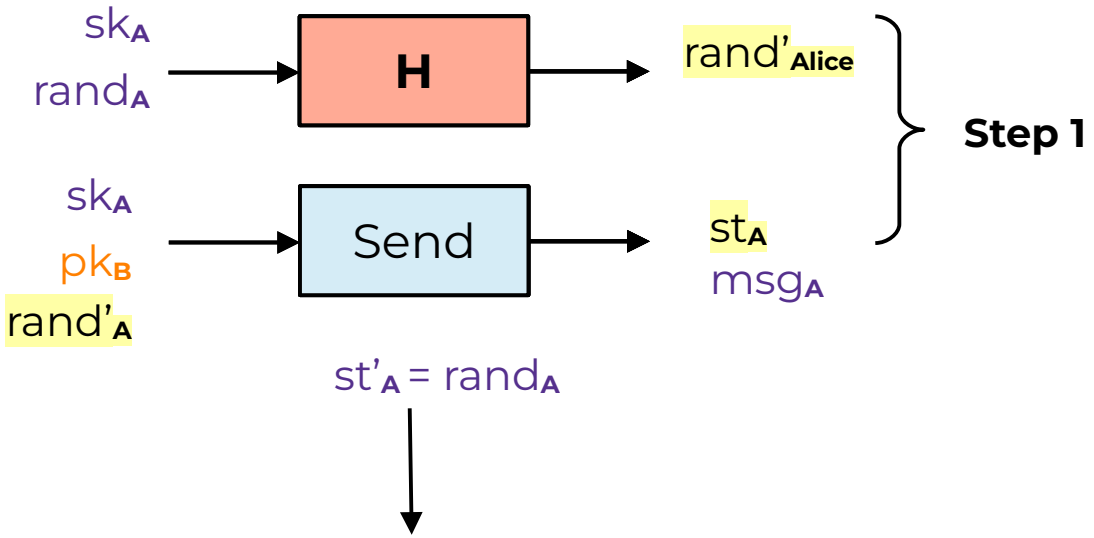


The 2R transform

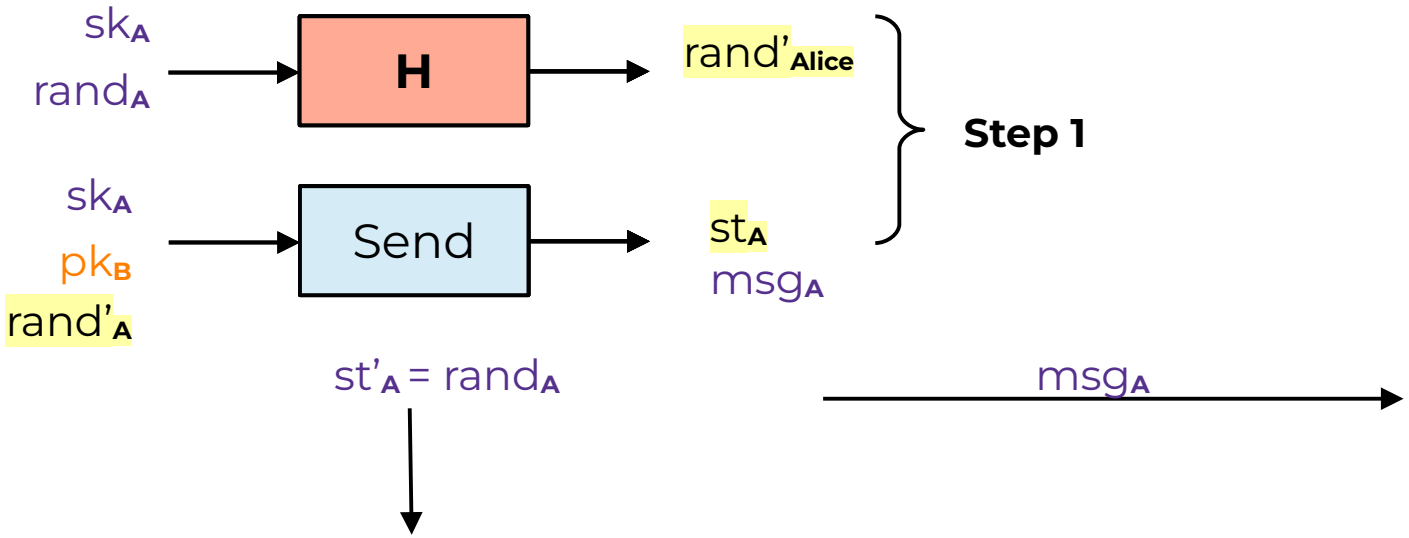
The 2R transform



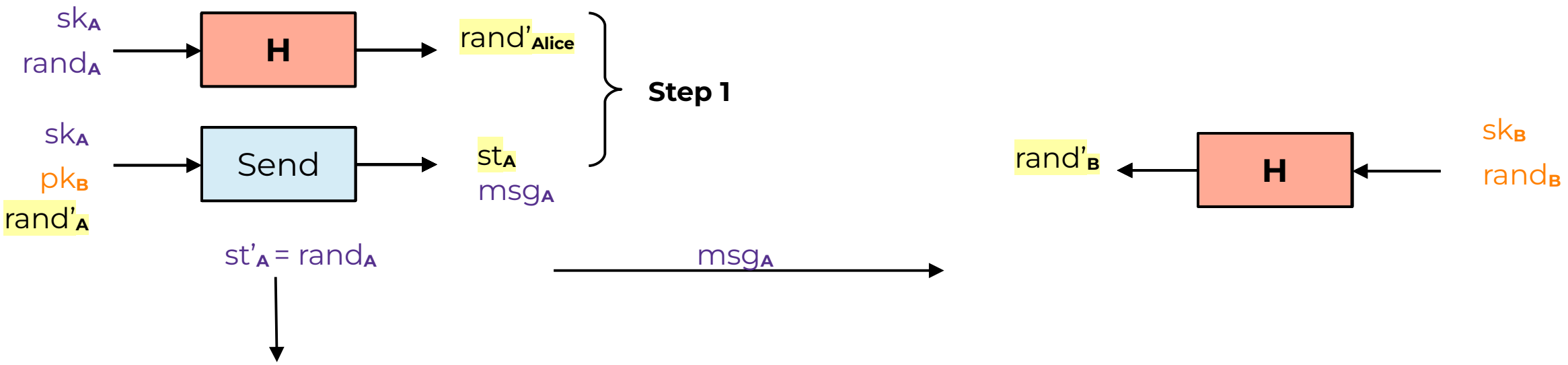
The 2R transform



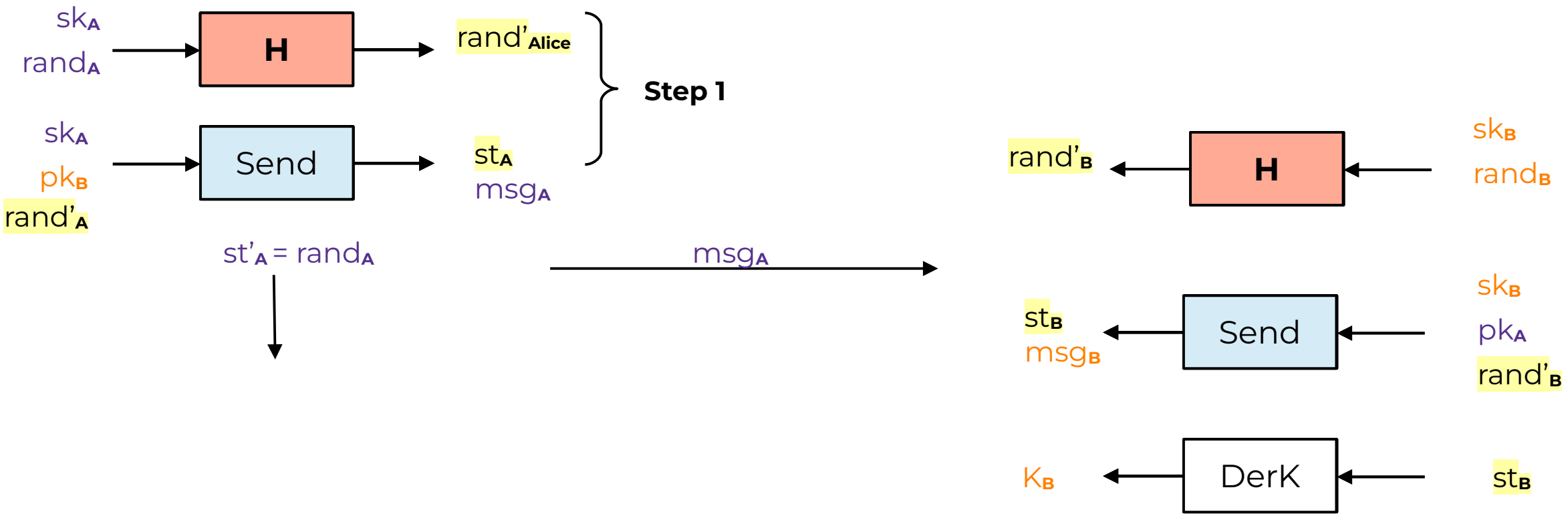
The 2R transform



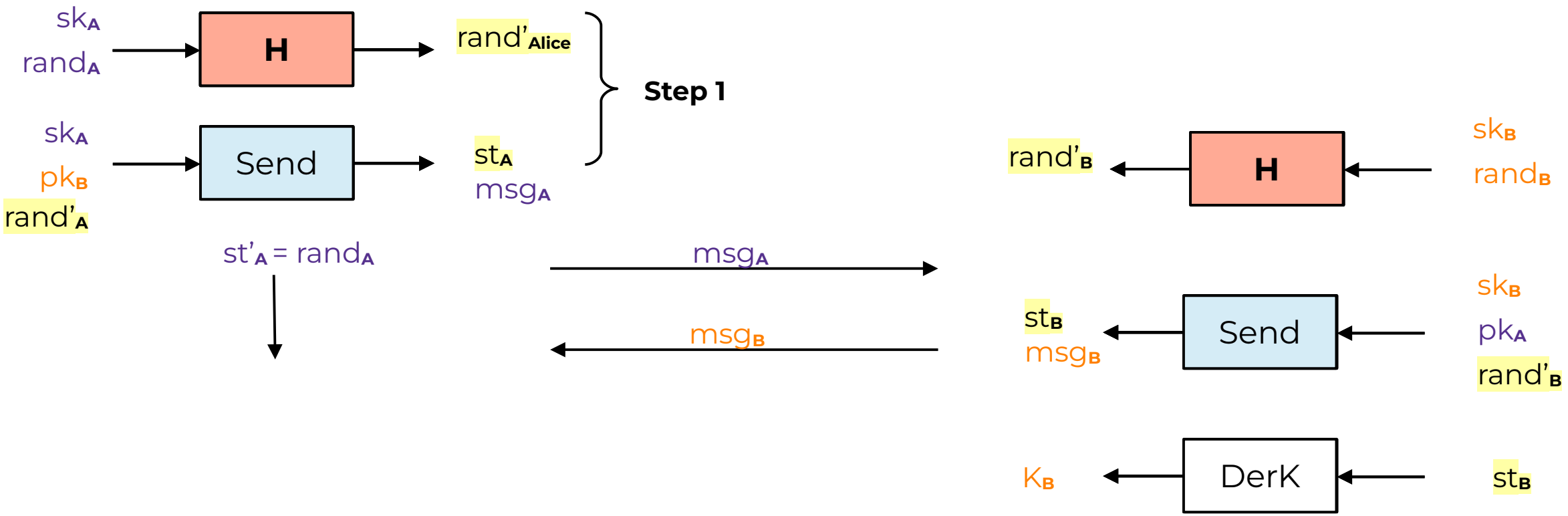
The 2R transform



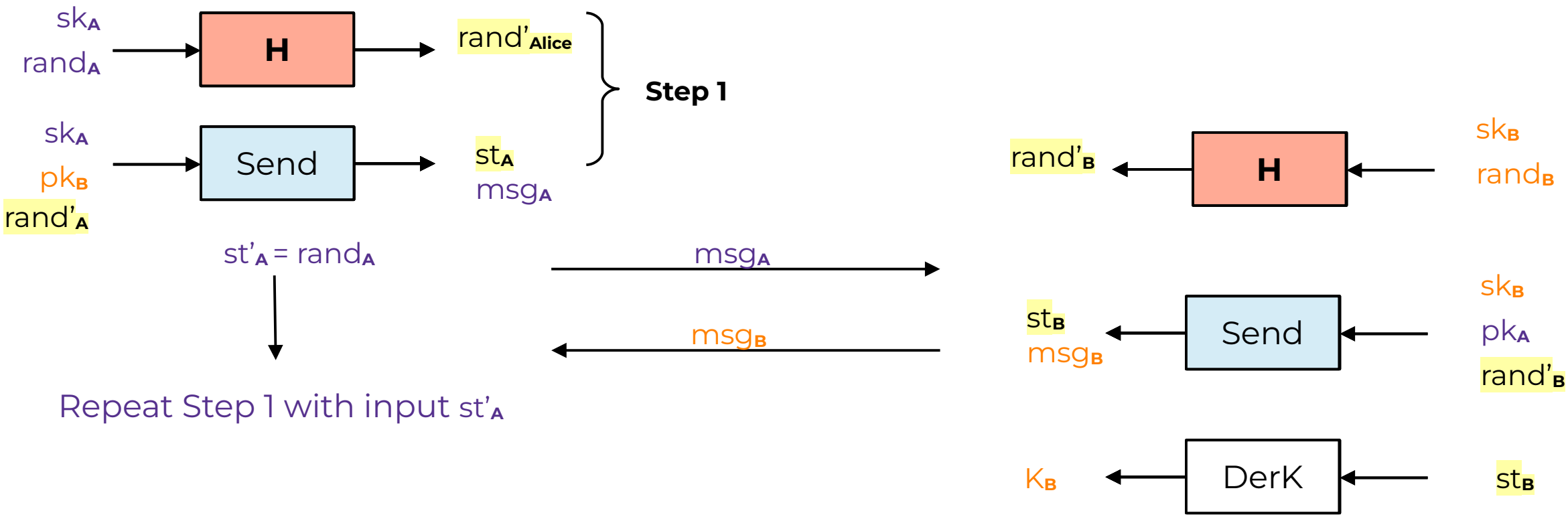
The 2R transform



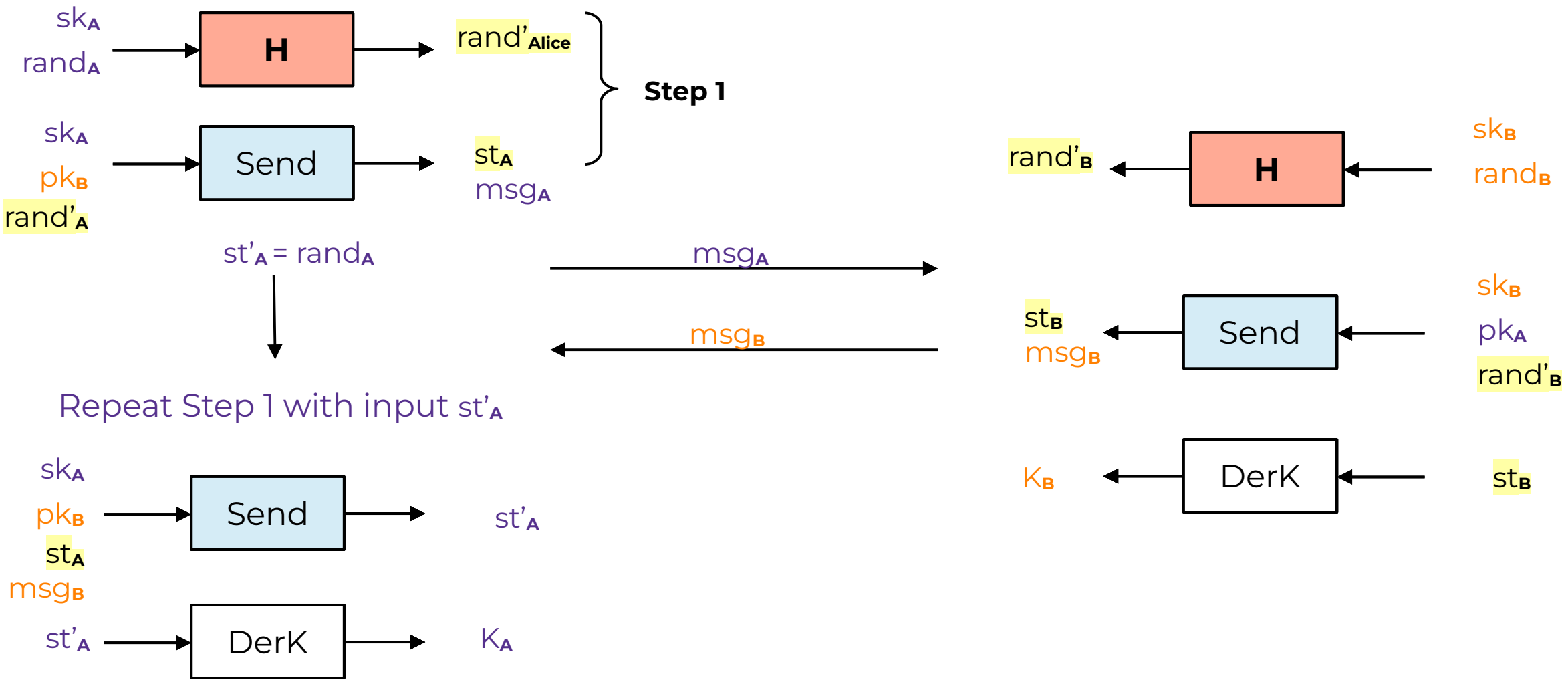
The 2R transform



The 2R transform

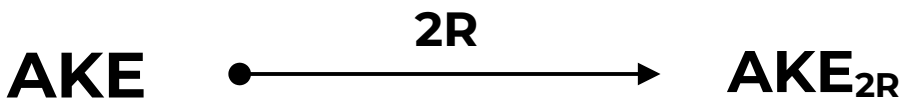


The 2R transform



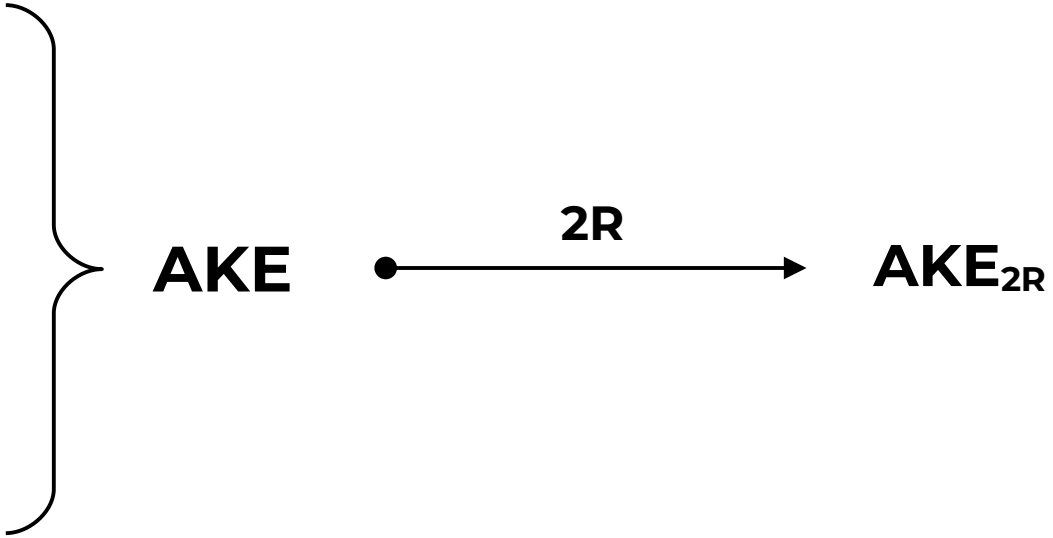
The 2R transform

The 2R transform



The 2R transform

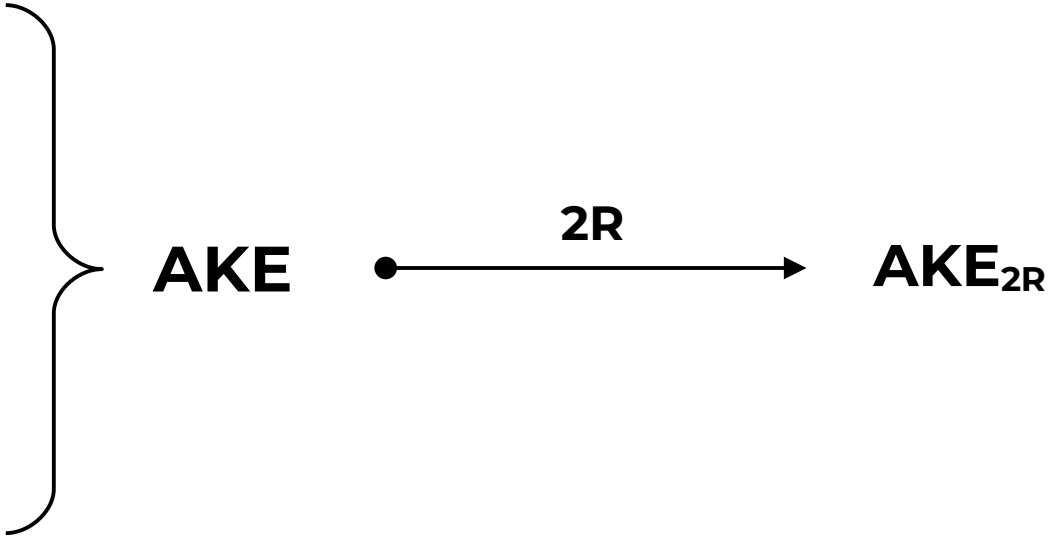
Secure in our model without any state reveals of test sessions



The 2R transform

Secure in our model without any state reveals of test sessions

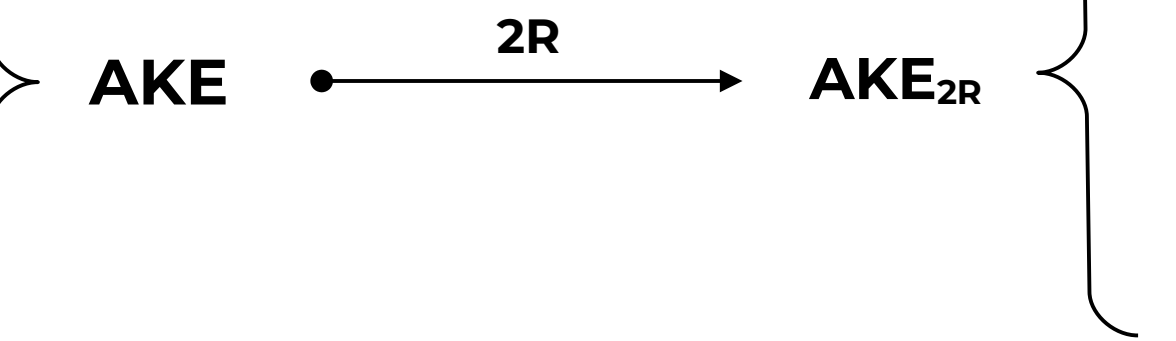
Sessions independent of test sessions are secure against preimage reveals



The 2R transform

Secure in our model without any state reveals of test sessions

Sessions independent of test sessions are secure against preimage reveals



Secure in our model against randomness reveal and ordinary state reveals of test sessions

The 2R transform

Secure in our model without any state reveals of test sessions

Sessions independent of test sessions are secure against preimage reveals

AKE —**2R**→ **AKE_{2R}**

Secure in our model against randomness reveal and ordinary state reveals of test sessions

Sessions independent of test sessions are secure against preimage **and** other state reveals

The 2R transform

Secure in our model without any state reveals of test sessions

Sessions independent of test sessions are secure against preimage reveals

AKE —**2R**→ **AKE_{2R}**

No known protocols yet

Secure in our model against randomness reveal and ordinary state reveals of test sessions

Sessions independent of test sessions are secure against preimage **and** other state reveals

The 2R transform

Secure in our model without any state reveals of test sessions

Sessions independent of test sessions are secure against preimage reveals

Candidates ?

AKE —**2R**→ **AKE_{2R}**

No known protocols yet

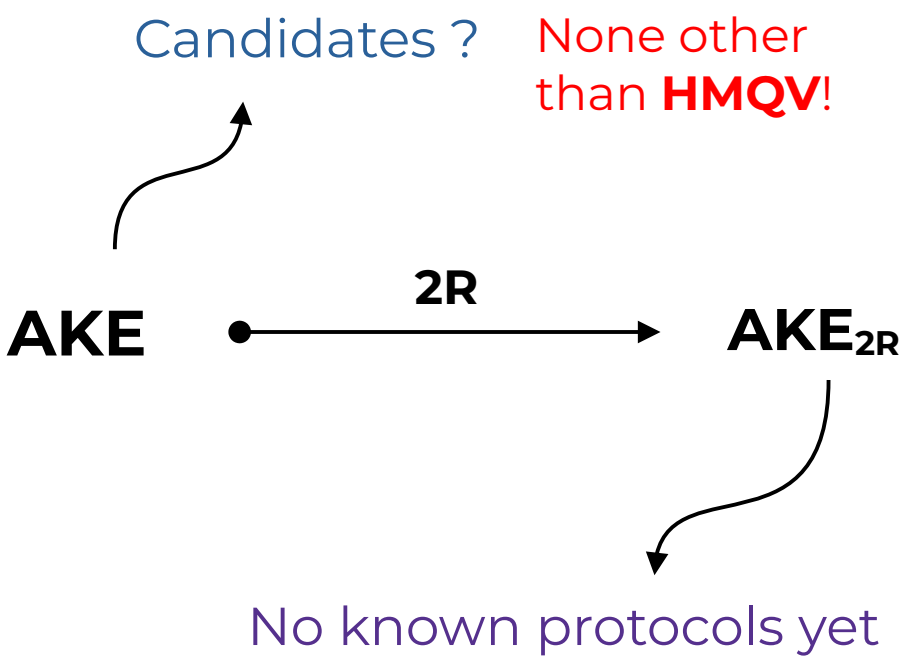
Secure in our model against randomness reveal and ordinary state reveals of test sessions

Sessions independent of test sessions are secure against preimage **and** other state reveals

The 2R transform

Secure in our model without any state reveals of test sessions

Sessions independent of test sessions are secure against preimage reveals



Secure in our model against randomness reveal and ordinary state reveals of test sessions

Sessions independent of test sessions are secure against preimage **and** other state reveals

HMQV: a curious case

Alice (a, $A = g^a$) **Bob** (b, $B = g^b$)

$x \leftarrow\$ \mathbf{Z_p}$ $X = g^x$ $y \leftarrow\$ \mathbf{Z_p}$

$Y = g^y$

$K := \mathbf{H'}(ctxt , (XA^d)^{y+be})$

$g^{(x + ad)(y + be)}$

$K := \mathbf{H'}(ctxt , (YB^e)^{x +ad})$

$g^{(y + be)(x + ad)}$

$ctxt := (A, B, X, Y)$

$d = \mathbf{H_1}(X, \mathbf{Bob})$; $e = \mathbf{H_2}(Y, \mathbf{Alice})$

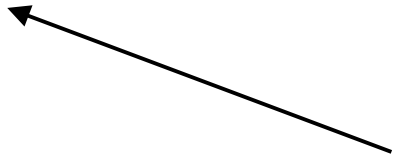
Hugo Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol, CRYPTO 2005

HMQV: Properties

HMQV

HMQV: Properties

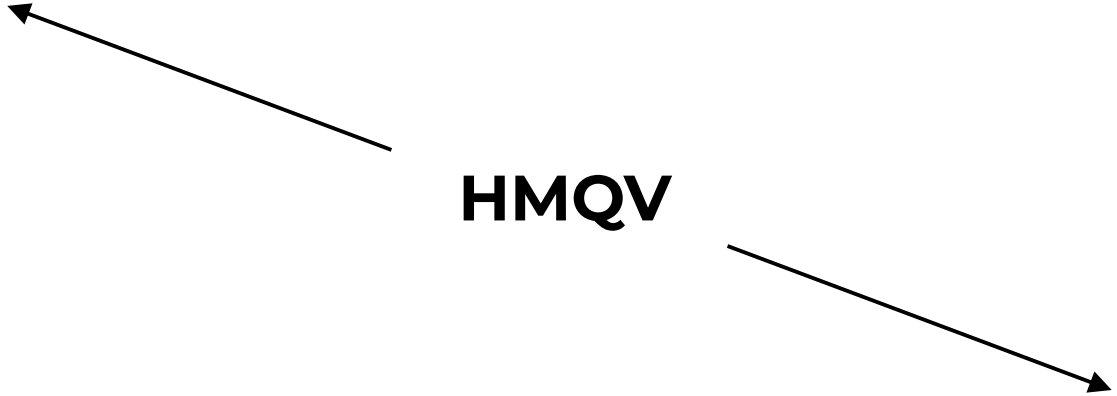
Secure against
randomness reveal and
ordinary state reveal



HMQV

HMQV: Properties

Secure against randomness reveal and ordinary state reveal



HMQV

Independent sessions are secure against preimage reveal

HMQV: Properties

Secure against randomness reveal and ordinary state reveal

HMQV

Post-processing of randomness

Independent sessions are secure against preimage reveal

HMQV: Properties

Secure against randomness reveal and ordinary state reveal

Efficient

Signal considers implementation of a variation of it

HMQV

Post-processing of randomness

Independent sessions are secure against preimage reveal

HMQV: Properties

Secure against randomness reveal and ordinary state reveal

Efficient

Signal considers implementation of a variation of it

HMQV

Post-processing of randomness

Independent sessions are secure against preimage reveal

Good candidate for 2R transform!

HMQV + 2R transform

Alice (a, $A = g^a$)

$x \leftarrow\$ \mathbf{Z_p}$

$x' \leftarrow \mathbf{H}(x, a)$

$X' = g^{x'}$ →

← $Y' = g^{y'}$

$K := \mathbf{H'}(ctxt , (Y'B^e)^{x' + ad})$

$g^{(y' + be)(x' + ad)}$

Bob (b, $B = g^b$)

$y \leftarrow\$ \mathbf{Z_p}$

$y' \leftarrow \mathbf{H}(y, b)$

$K := \mathbf{H'}(ctxt , (X'A^d)^{y'+be})$

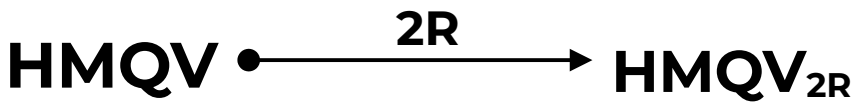
$g^{(x' + ad)(y' + be)}$

$ctxt := (A, B, X', Y')$

$d = \mathbf{H_1}(X', \mathbf{Bob})$; $e = \mathbf{H_2}(Y', \mathbf{Alice})$

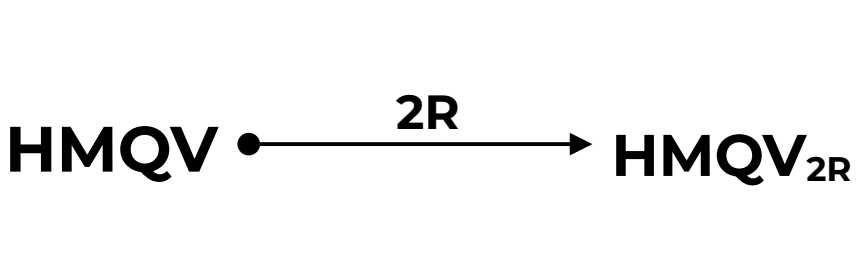
HMQV + 2R transform

HMQV + 2R transform



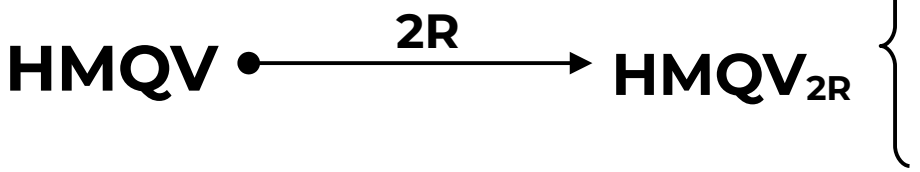
HMQV + 2R transform

Sessions independent of test sessions are secure against preimage reveals



HMQV + 2R transform

Sessions independent of test sessions are secure against preimage reveals



Sessions independent of test are secure against preimage **and** other state reveals

Desiderata

Our desires are fulfilled.

Desiderata

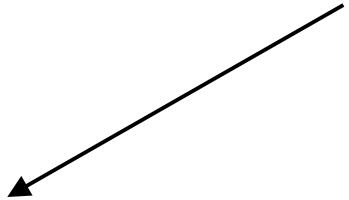
Our desires are fulfilled.

$$\mathrm{HMQV}_{\mathrm{2R}}$$

Desiderata

Our desires are fulfilled.

$\mathbf{HMQV_{2R}}$



Secure against

Ordinary state reveal and
Randomness reveal

Desiderata

Our desires are fulfilled.

$HMQV_{2R}$

Secure against

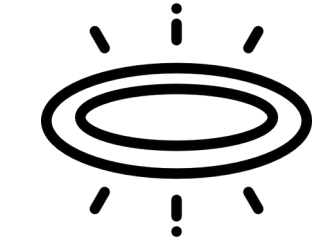
Ordinary state reveal and
Randomness reveal

Secure against

Preimage reveal **and** state-reveals
of independent sessions

Desiderata

Our desires are fulfilled.



HMQV$_{2R}$

Implicit authentication

High efficiency

Secure against

Ordinary state reveal and
Randomness reveal

Secure against

Preimage reveal **and** state-reveals
of independent sessions

Thank you for listening

Questions?