Combining PAKEs for hybrid security

Merged talk based on concurrent works by

PAKE Combiners and Efficient Post-Quantum Instantiations

Julia Hesse[1] **Michael Rosenberg**[2]

[1] IBM Research Europe — Zurich [2] Cloudflare

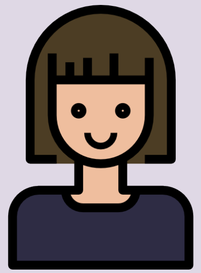
Hybrid Password Authentication Key Exchange in the UC Framework

You Lyu **Shengli Liu**

Shanghai Jiao Tong University, China

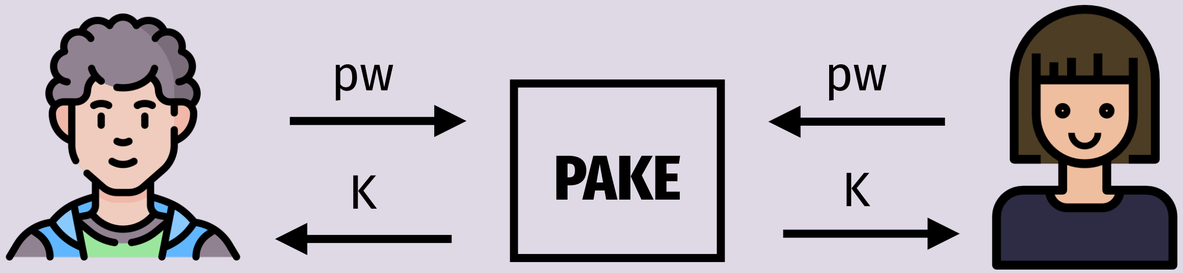
ia.cr/2024/{1621,1630}

What is a PAKE



What is a PAKE

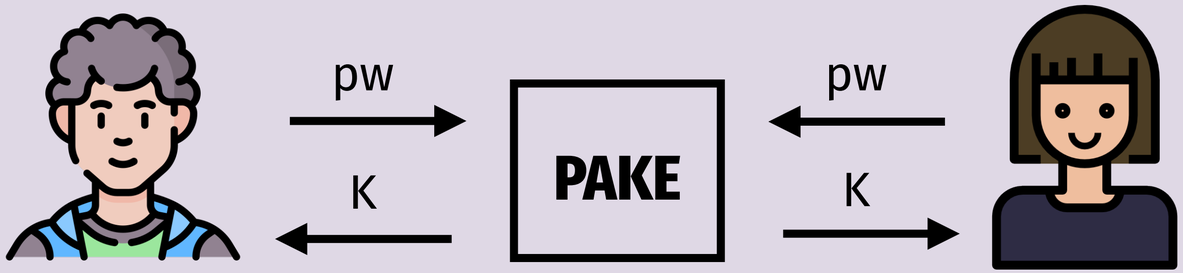
Two parties use a password to establish a secure shared secret

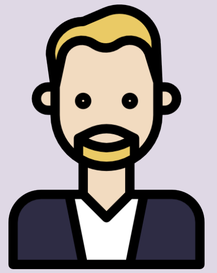


What is a PAKE

Two parties use a password to
establish a secure shared secret

1. A passive adversary cannot derive K

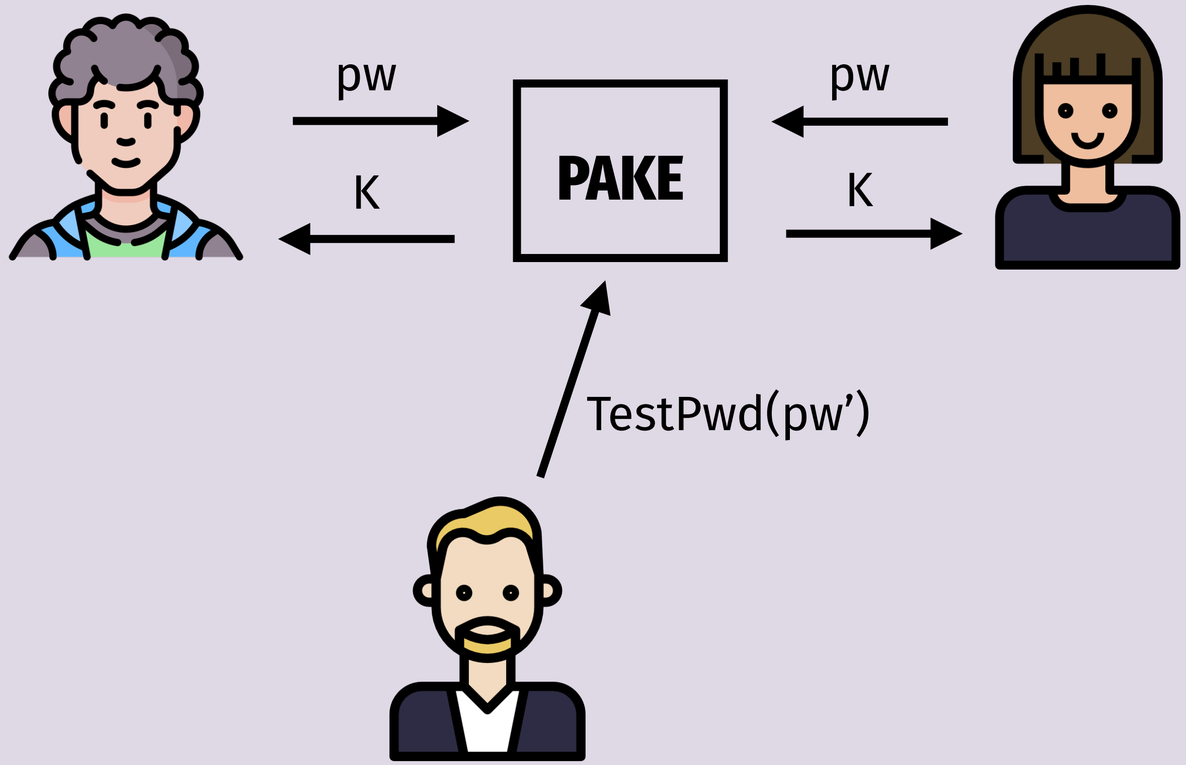




What is a PAKE

Two parties use a password to establish a secure shared secret

1. A passive adversary cannot derive K
2. An active adversary has only 1 pw guess per session



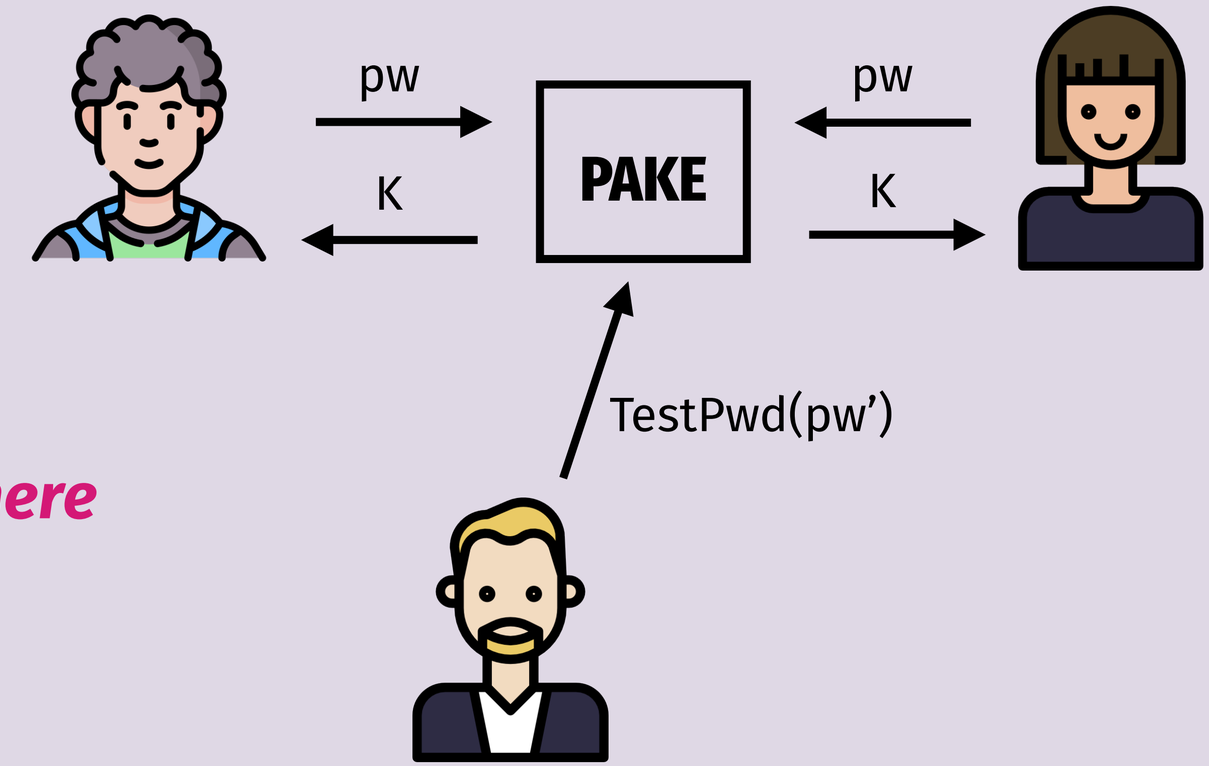
What is a PAKE

Two parties use a password to establish a secure shared secret

1. A passive adversary cannot derive K
2. An active adversary has only 1 pw guess per session

A PAKE combiner takes 2 PAKEs and produces a new PAKE

Weird! Cryptographic statements where nothing is high entropy!



Problem Statement

Problem Statement

We have classical and PQ PAKEs

Problem Statement

We have classical and PQ PAKEs

But the PQ ones are new

Problem Statement

We have classical and PQ PAKEs

But the PQ ones are new

note
proof flawed (Appx. A.1)
result incorrect (Sects. 3.2 and 3.3, Appx. A.4)
result incorrect (Sect. 3.2 and Appx. A.4)
result ambiguous, unclear if OEKE-PRF or OEKE-RO; security proof incorrect (Sect. 3.1); either way result also incorrect (Appx. A.3)
result incorrect (Sect. 3.3, Appxs. A.1 and A.3)
result incorrect (Appx. A.3)

Problem Statement

We have classical and PQ PAKEs

But the PQ ones are new

We can **hedge with hybrid**

note
proof flawed (Appx. A.1)
result incorrect (Sects. 3.2 and 3.3, Appx. A.4)
result incorrect (Sect. 3.2 and Appx. A.4)
result ambiguous, unclear if OEKE-PRF or OEKE-RO; security proof incorrect (Sect. 3.1); either way result also incorrect (Appx. A.3)
result incorrect (Sect. 3.3, Appxs. A.1 and A.3)
result incorrect (Appx. A.3)

Problem Statement

We have classical and PQ PAKEs

But the PQ ones are new

We can **hedge with hybrid**

Recommended by French ANSSI and German BSI

note
proof flawed (Appx. A.1)
result incorrect (Sects. 3.2 and 3.3, Appx. A.4)
result incorrect (Sect. 3.2 and Appx. A.4)
result ambiguous, unclear if OEKE-PRF or OEKE-RO; security proof incorrect (Sect. 3.1); either way result also incorrect (Appx. A.3)
result incorrect (Sect. 3.3, Appxs. A.1 and A.3)
result incorrect (Appx. A.3)

Problem Statement

We have classical and PQ PAKEs

But the PQ ones are new

We can **hedge with hybrid**

Recommended by French ANSSI and German BSI

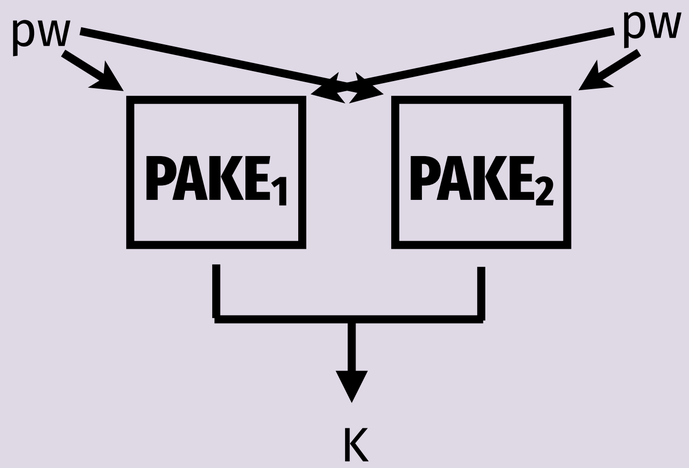
How do you make a hybrid PAKE?

note
proof flawed (Appx. A.1)
result incorrect (Sects. 3.2 and 3.3, Appx. A.4)
result incorrect (Sect. 3.2 and Appx. A.4)
result ambiguous, unclear if OEKE-PRF or OEKE-RO; security proof incorrect (Sect. 3.1); either way result also incorrect (Appx. A.3)
result incorrect (Sect. 3.3, Appxs. A.1 and A.3)
result incorrect (Appx. A.3)

Our Contributions

Our Contributions

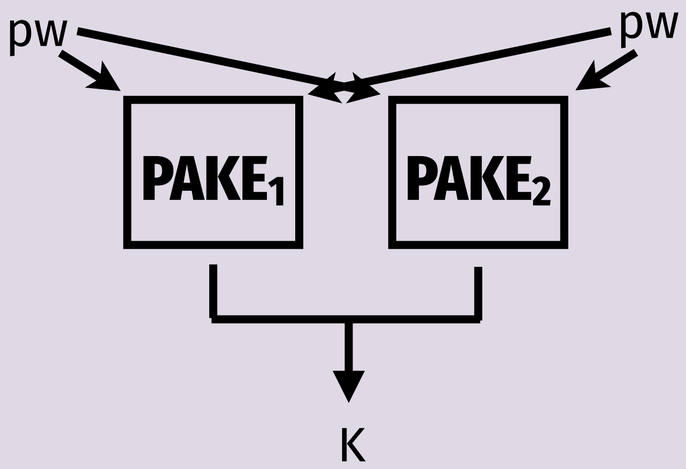
Parallel combiner (ParComb)



1-round PAKE + 1-round PAKE ⇒ 1-round PAKE

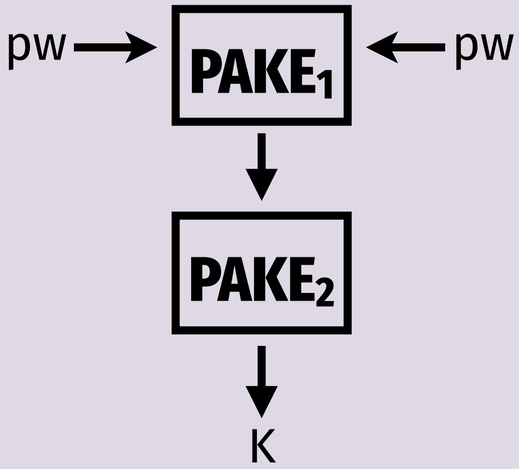
Our Contributions

Parallel combiner (ParComb)



1-round PAKE + 1-round PAKE ⇒ 1-round PAKE

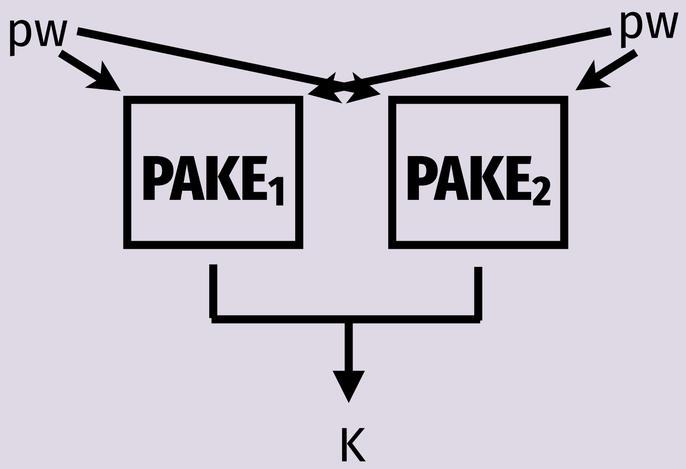
Sequential combiner (SeqComb)



1-round PAKE + N-round PAKE ⇒ (N+1)-round PAKE

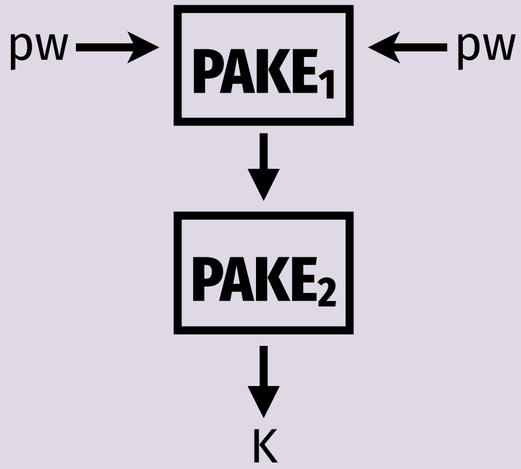
Our Contributions

Parallel combiner (ParComb)



1-round PAKE + 1-round PAKE ⇒ 1-round PAKE

Sequential combiner (SeqComb)

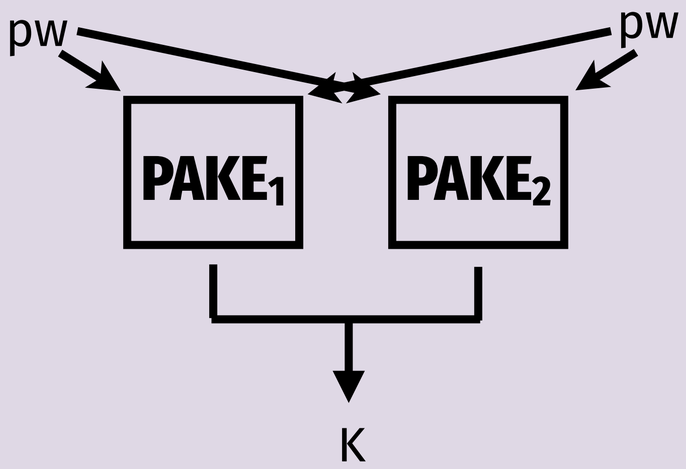


1-round PAKE + N-round PAKE ⇒ (N+1)-round PAKE

Hybrid: can plug in an existing classical and PQ PAKEs

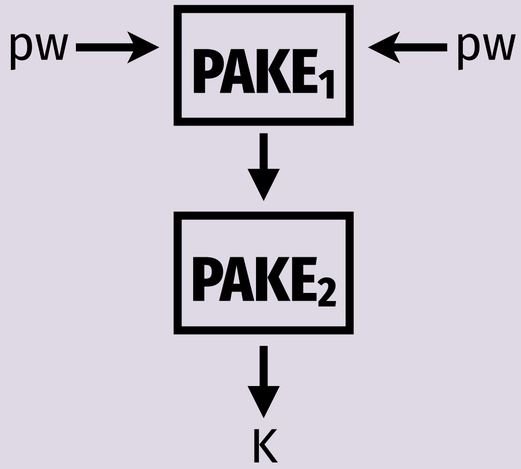
Our Contributions

Parallel combiner (ParComb)



1-round PAKE + 1-round PAKE ⇒ 1-round PAKE

Sequential combiner (SeqComb)



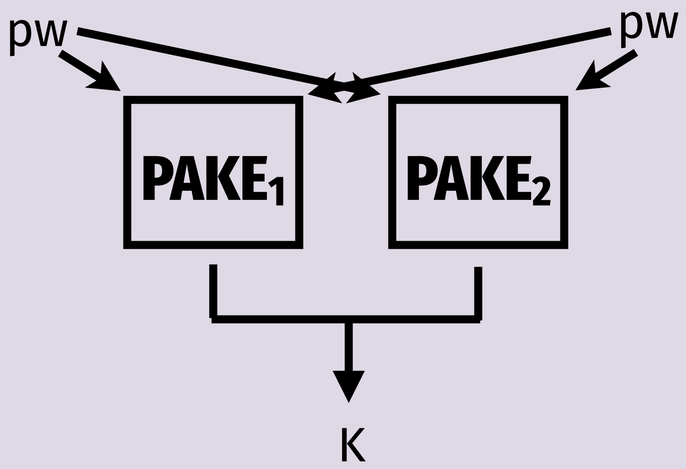
1-round PAKE + N-round PAKE ⇒ (N+1)-round PAKE

Hybrid: can plug in an existing classical and PQ PAKEs

Cheap: overhead of at most 2 hashes

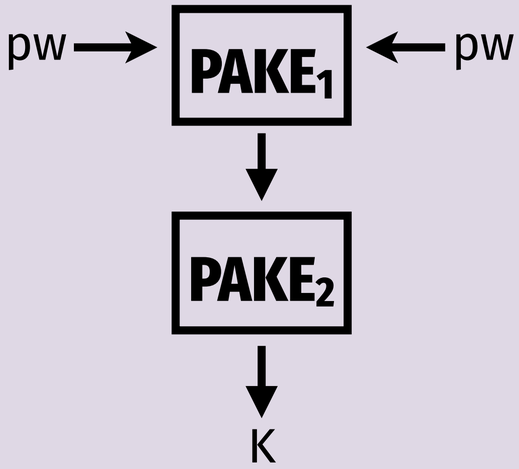
Our Contributions

Parallel combiner (ParComb)



1-round PAKE + 1-round PAKE ⇒ 1-round PAKE

Sequential combiner (SeqComb)



1-round PAKE + N-round PAKE ⇒ (N+1)-round PAKE

Hybrid: can plug in an existing classical and PQ PAKEs

Cheap: overhead of at most 2 hashes

Yields other PAKE flavors: PAKE ⇒ aPAKE, iPAKE

How not to make a hybrid PAKE

How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

EKE w/ key conf.

How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

EKE w/ key conf.

Diffie-Hellman with pw-encrypted shares

How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

EKE w/ key conf.

Diffie-Hellman with pw-encrypted shares

$r \leftarrow \mathbb{F}$

$R' := \mathrm{Enc}_{pw}(rG)$

How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

EKE w/ key conf.

Diffie-Hellman with

$r \leftarrow \mathbb{F}$

$R' := \mathrm{Enc}_{pw}(rG)$ **pw-encrypted shares**

keyed permutation

How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

EKE w/ key conf.

$r \leftarrow \mathbb{F}$

$R' := \mathrm{Enc}_{pw}(rG)$ $\xrightarrow{R'}$

How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

EKE w/ key conf.

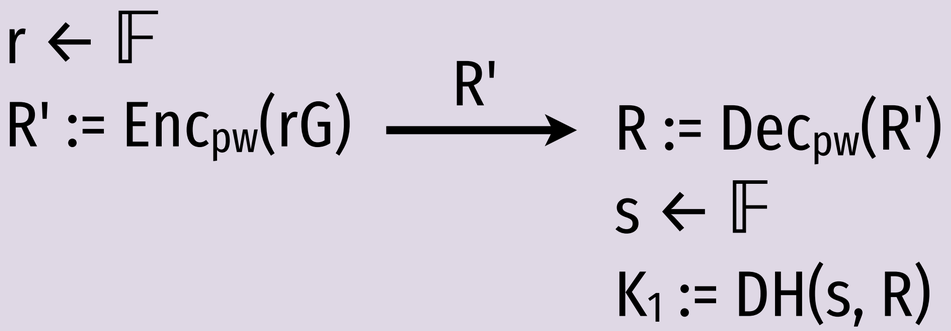
$r \leftarrow \mathbb{F}$

$R' := \mathrm{Enc}_{pw}(rG) \xrightarrow{R'} R := \mathrm{Dec}_{pw}(R')$

How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

EKE w/ key conf.



How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

EKE w/ key conf.

$r \leftarrow \mathbb{F}$

$R' := \mathrm{Enc}_{pw}(rG)$ —R'→ $R := \mathrm{Dec}_{pw}(R')$

$s \leftarrow \mathbb{F}$

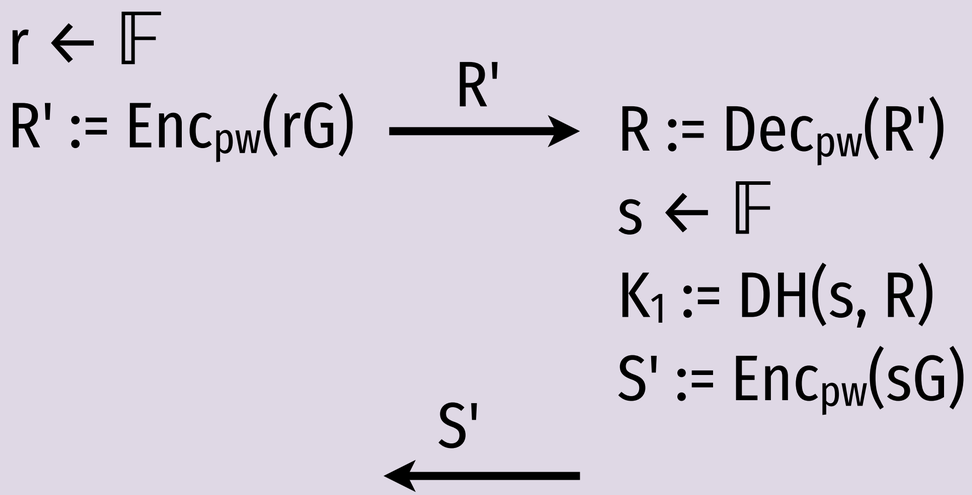
$K_1 := \mathrm{DH}(s, R)$

$S' := \mathrm{Enc}_{pw}(sG)$

How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

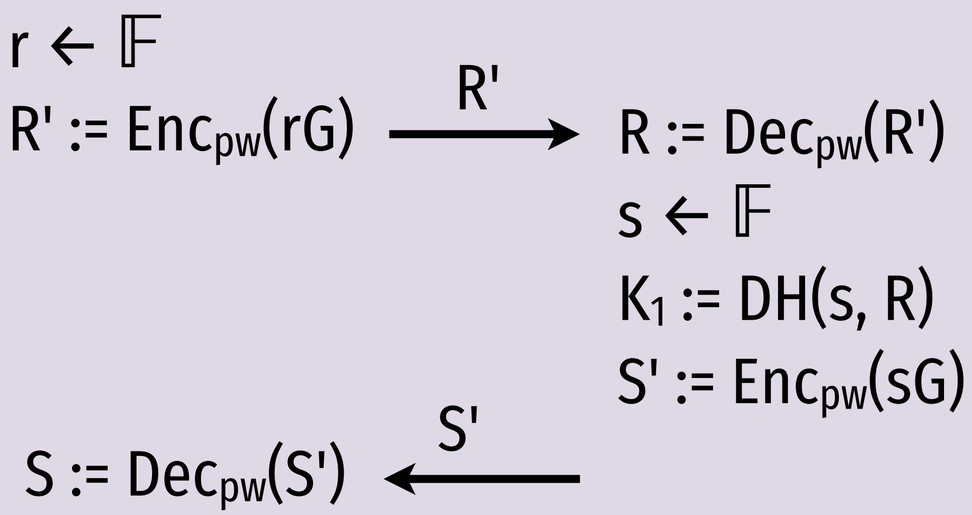
EKE w/ key conf.



How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

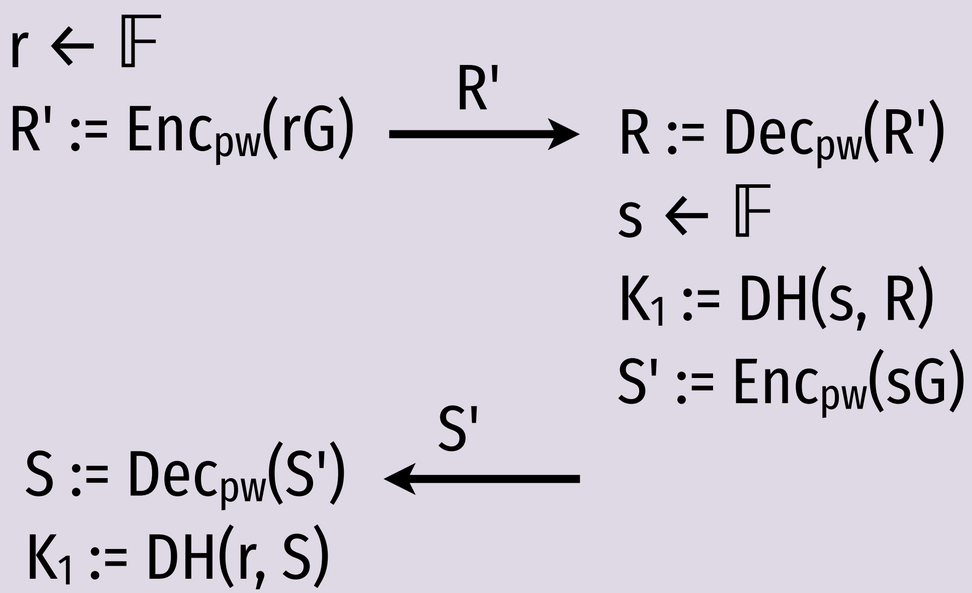
EKE w/ key conf.



How not to make a hybrid PAKE

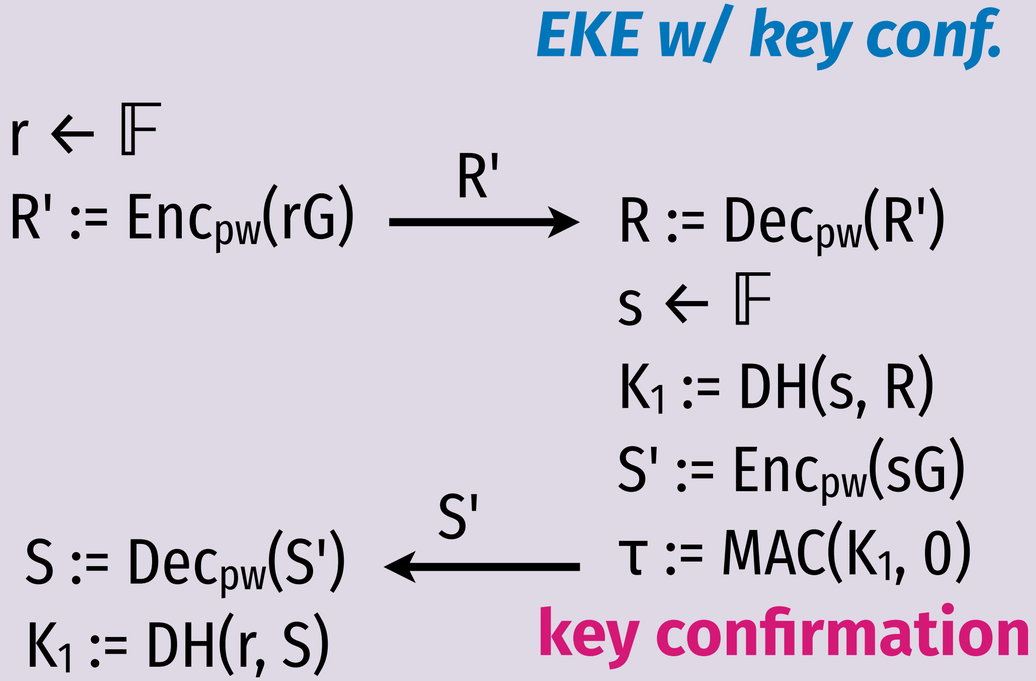
KEM combiners: just run 2 KEMs and hash the outputs

EKE w/ key conf.



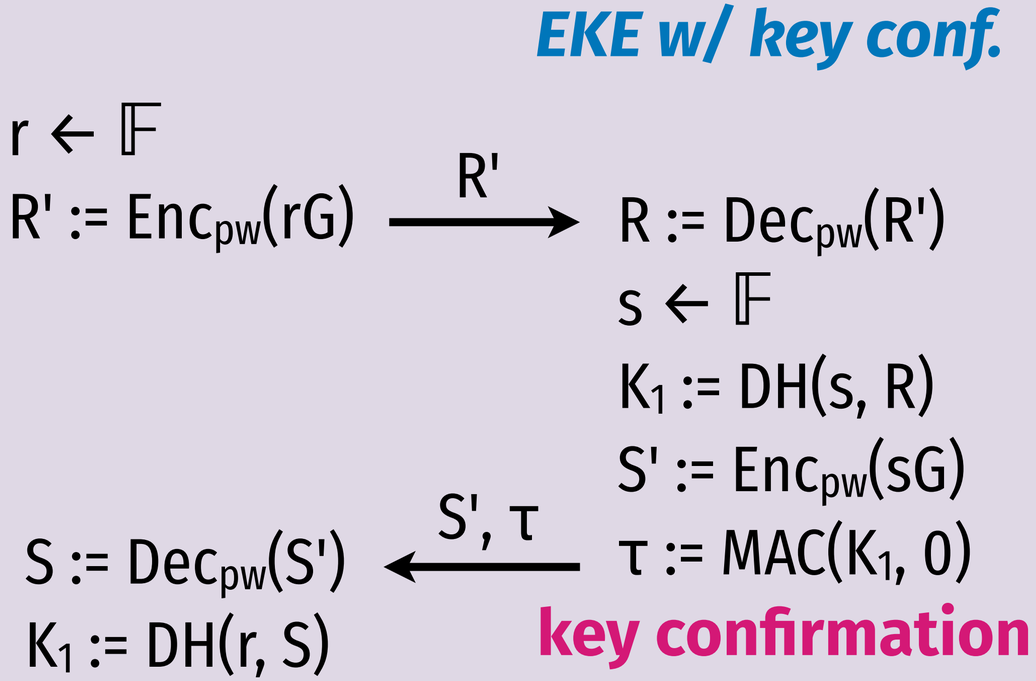
How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs



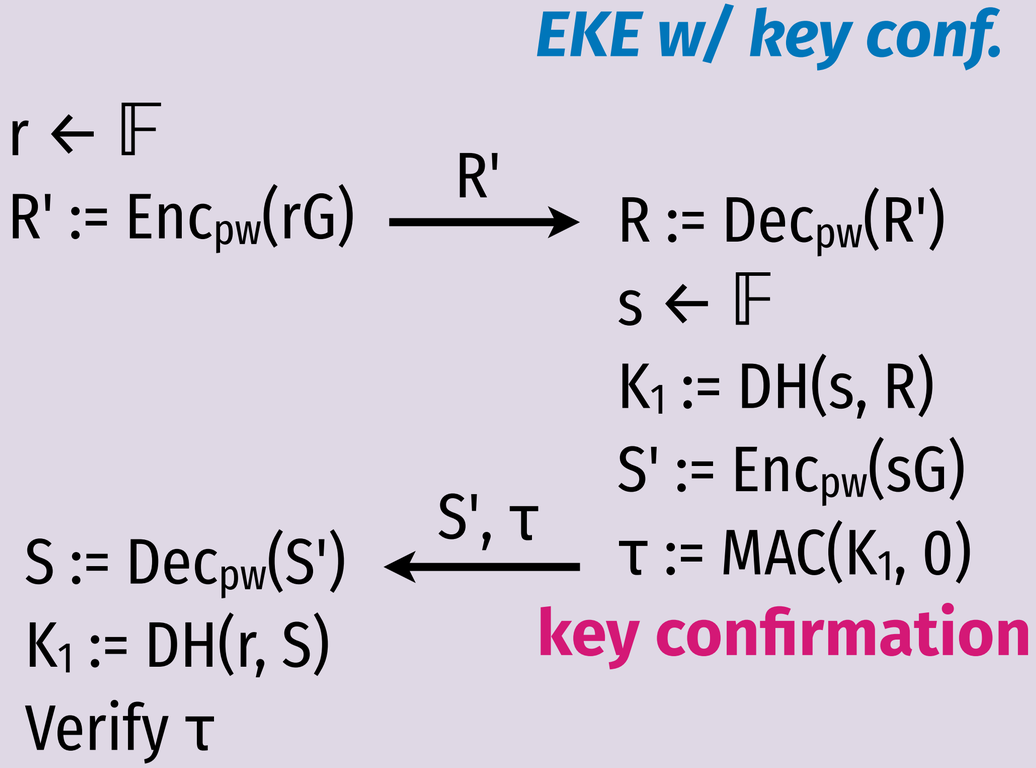
How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs



How not to make a hybrid PAKE

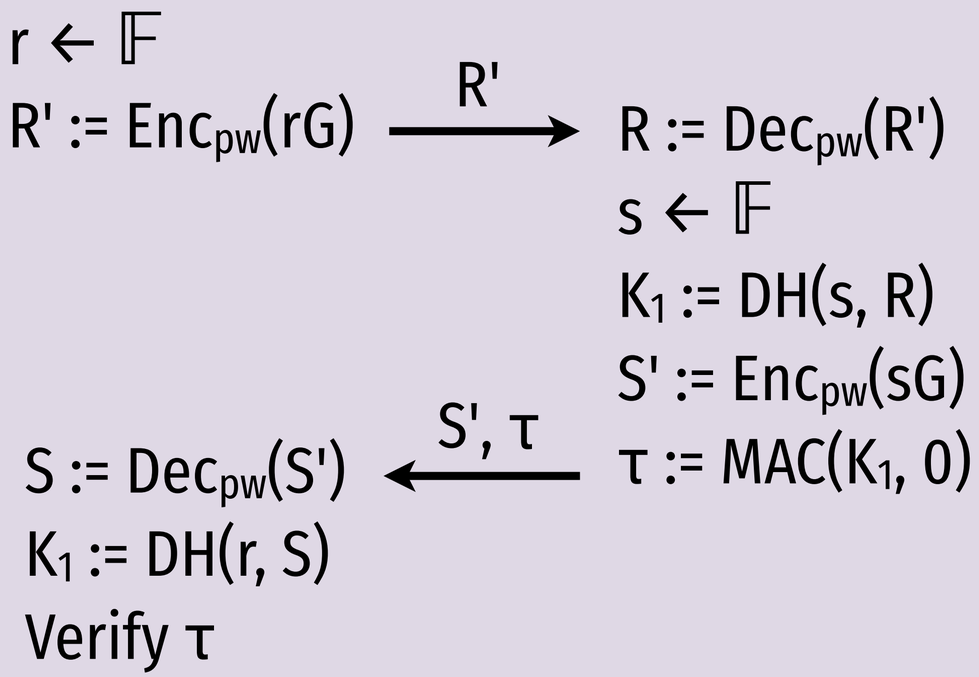
KEM combiners: just run 2 KEMs and hash the outputs



How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

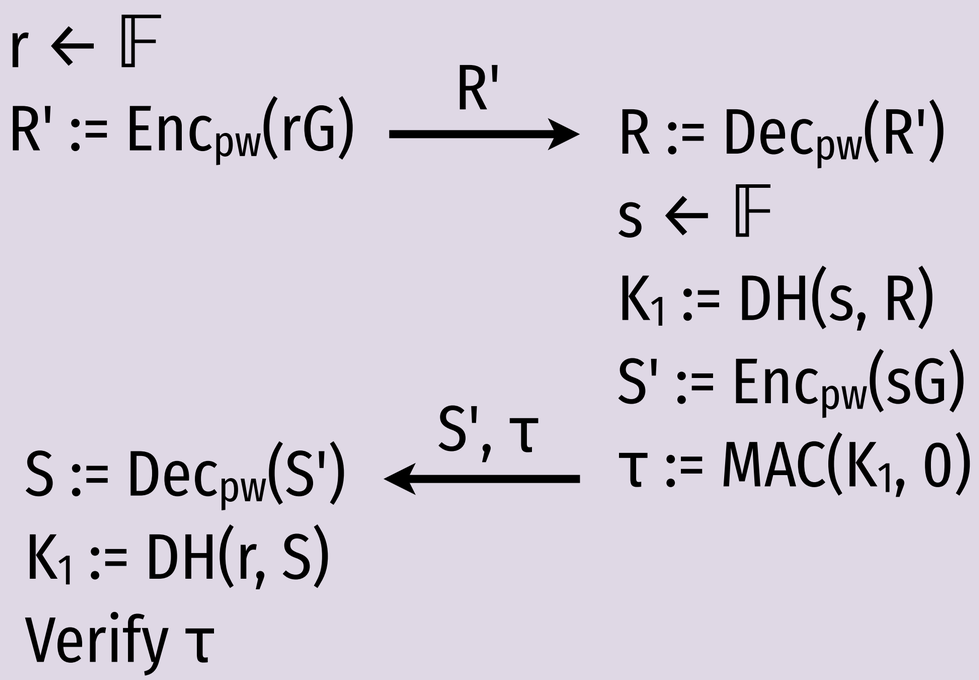
EKE w/ key conf.



How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs

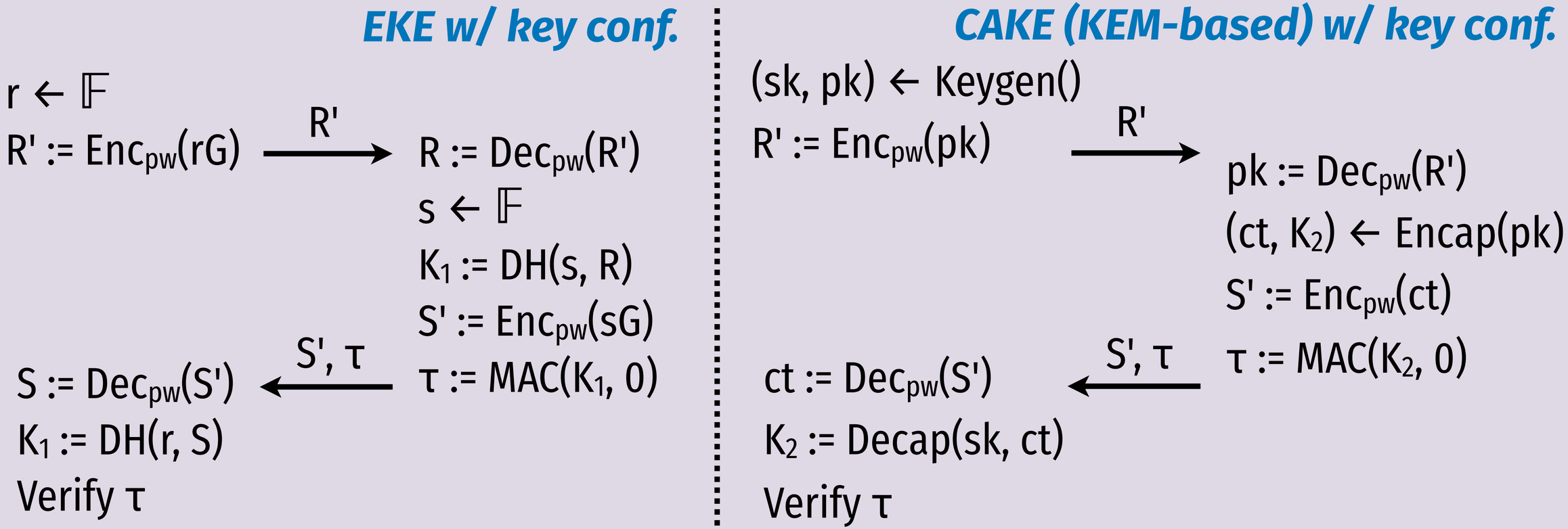
EKE w/ key conf.



CAKE (KEM-based) w/ key conf.

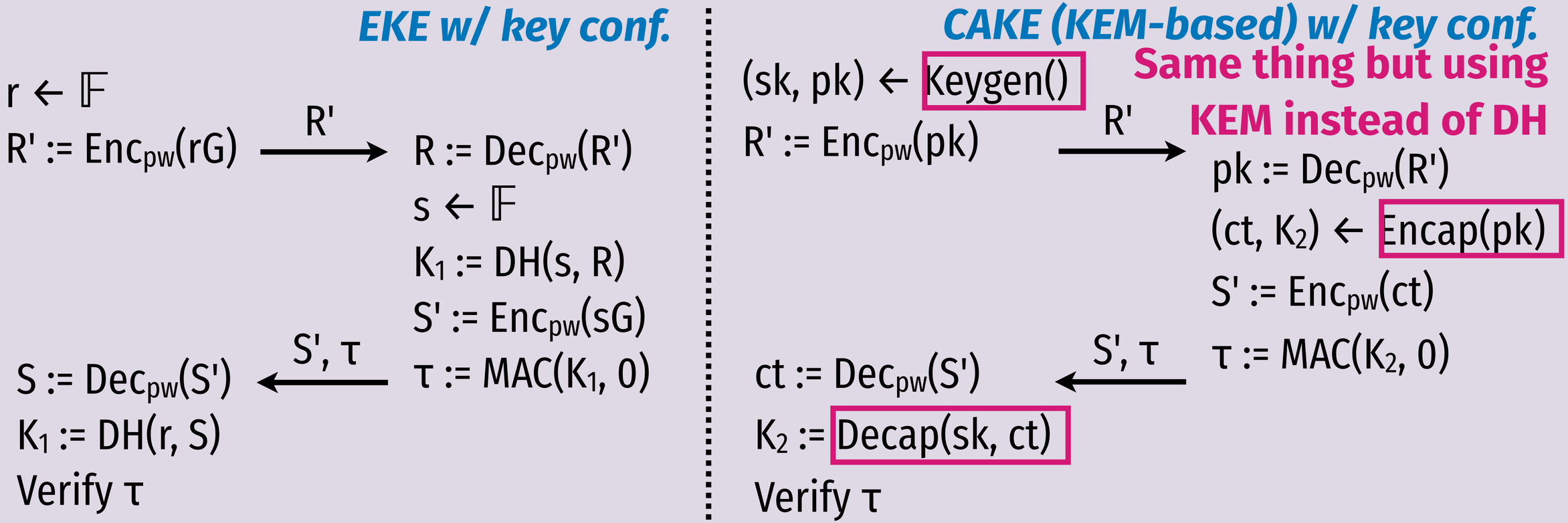
How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs



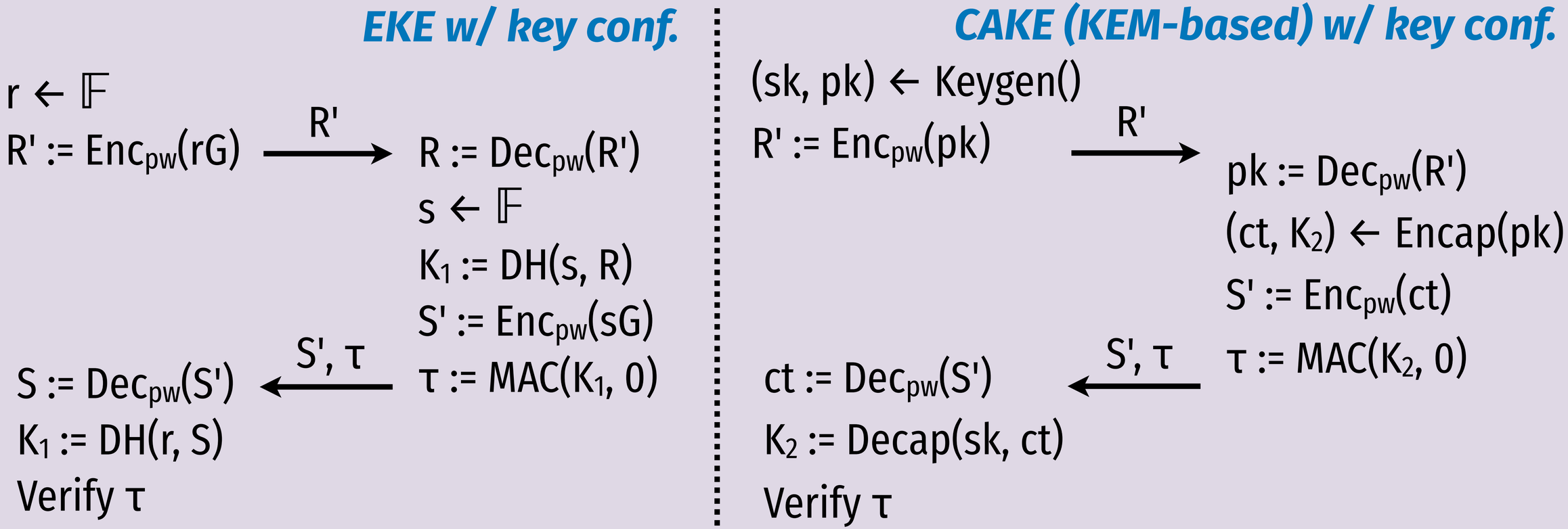
How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs



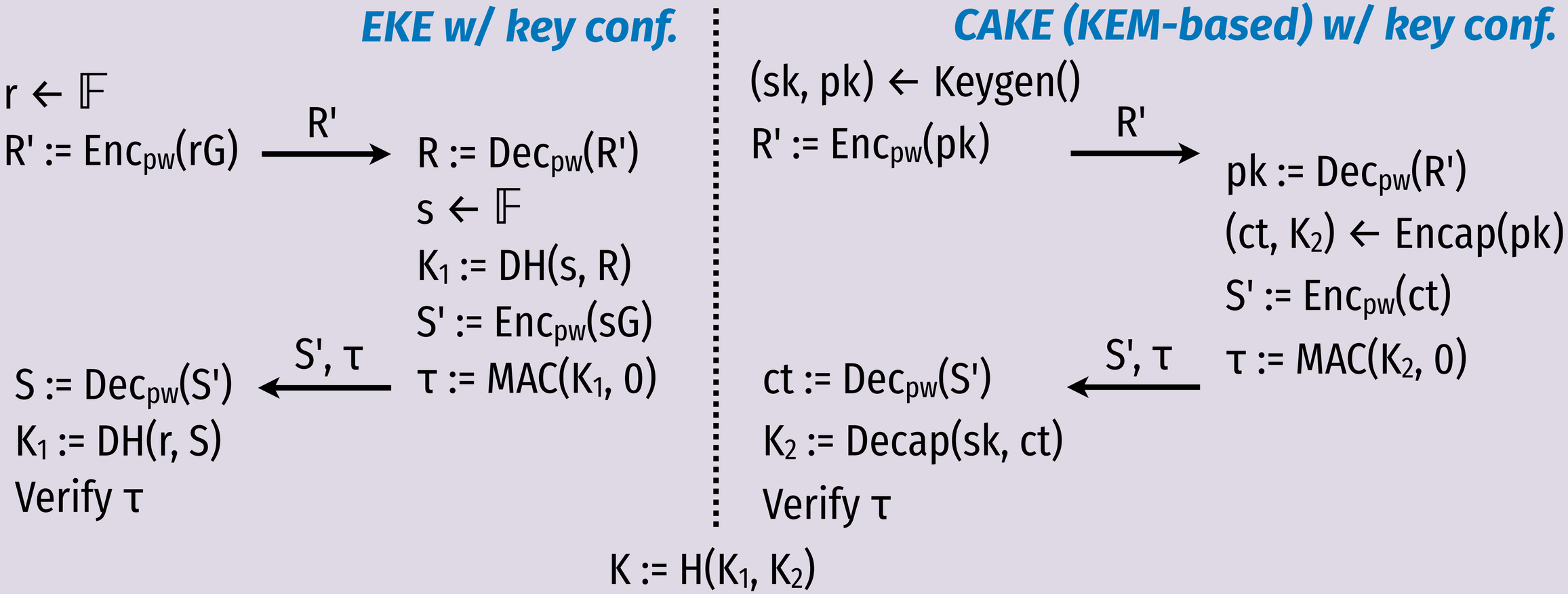
How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs



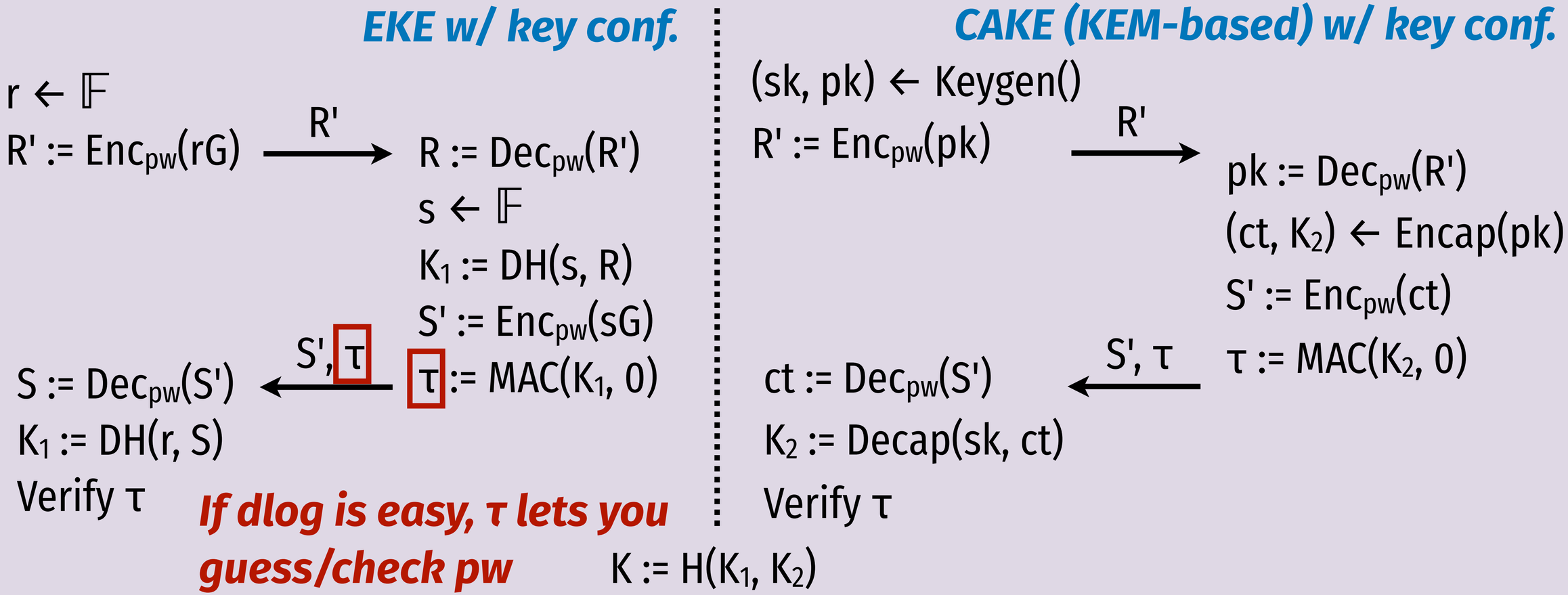
How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs



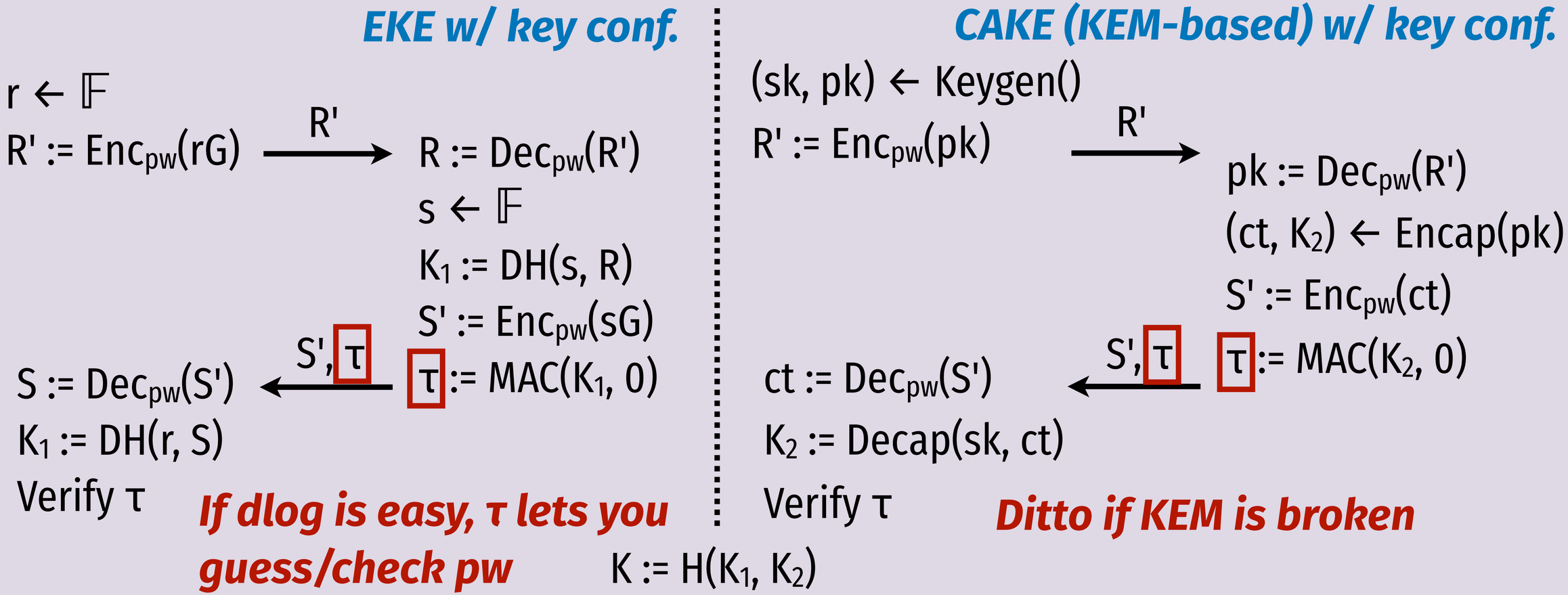
How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs



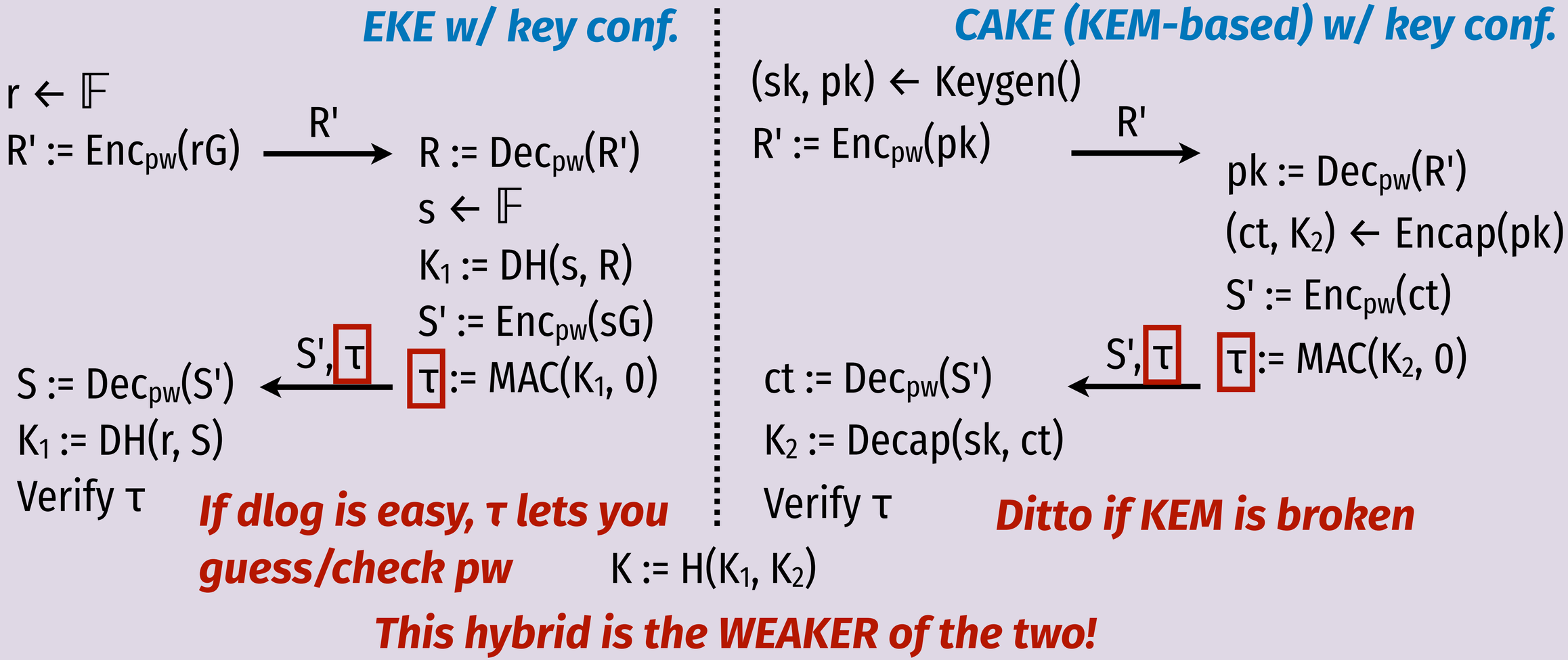
How not to make a hybrid PAKE

KEM combiners: just run 2 KEMs and hash the outputs



How not to make a hybrid PAKE

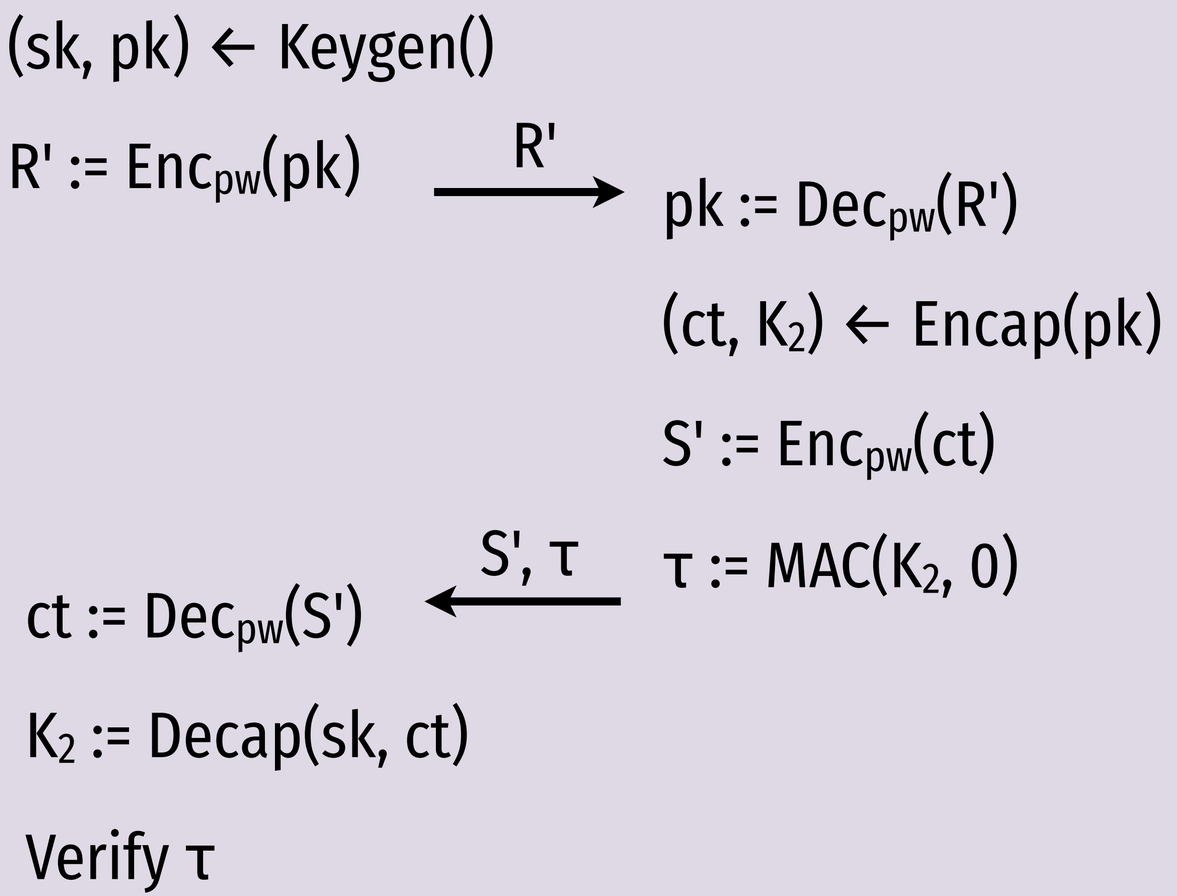
KEM combiners: just run 2 KEMs and hash the outputs



How else not to make a hybrid PAKE

How else not to make a hybrid PAKE

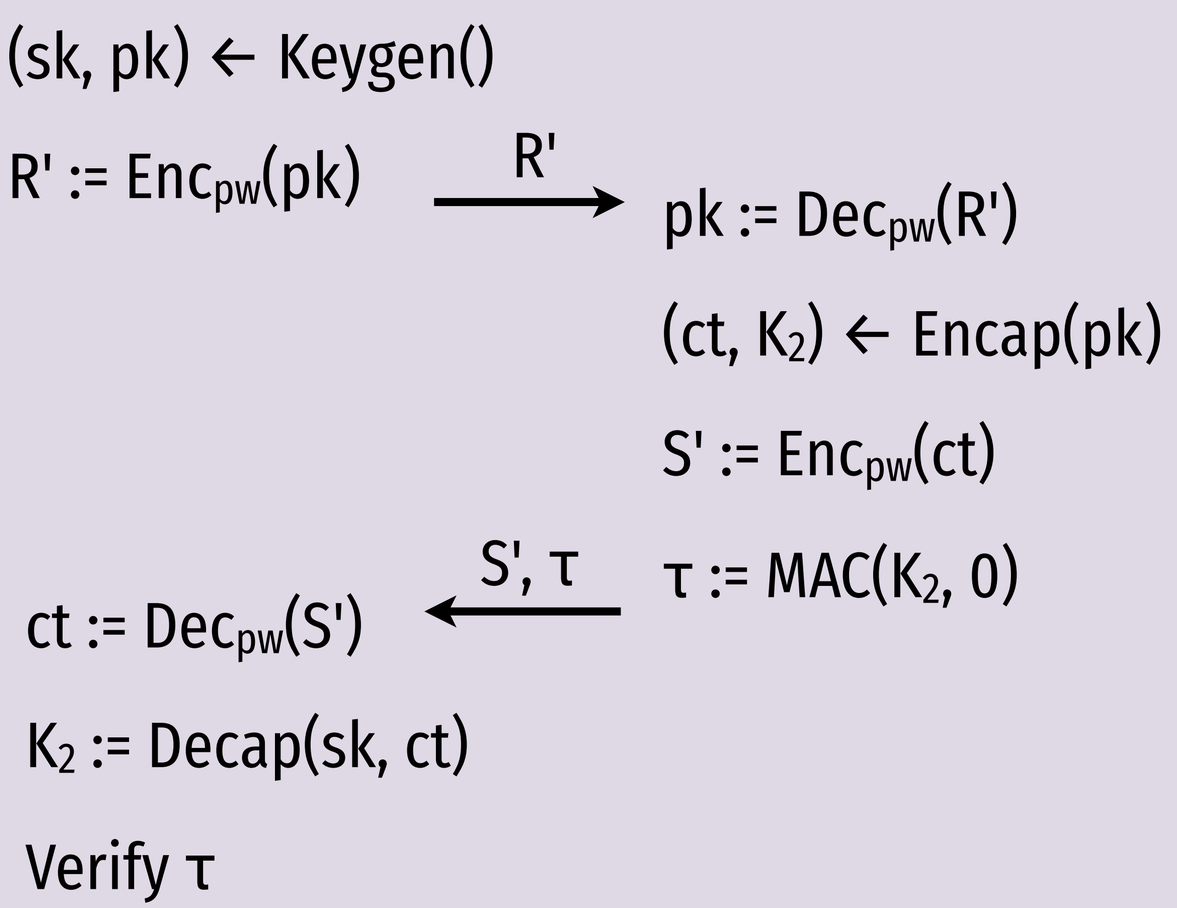
CAKE (KEM-based) with key conf.



How else not to make a hybrid PAKE

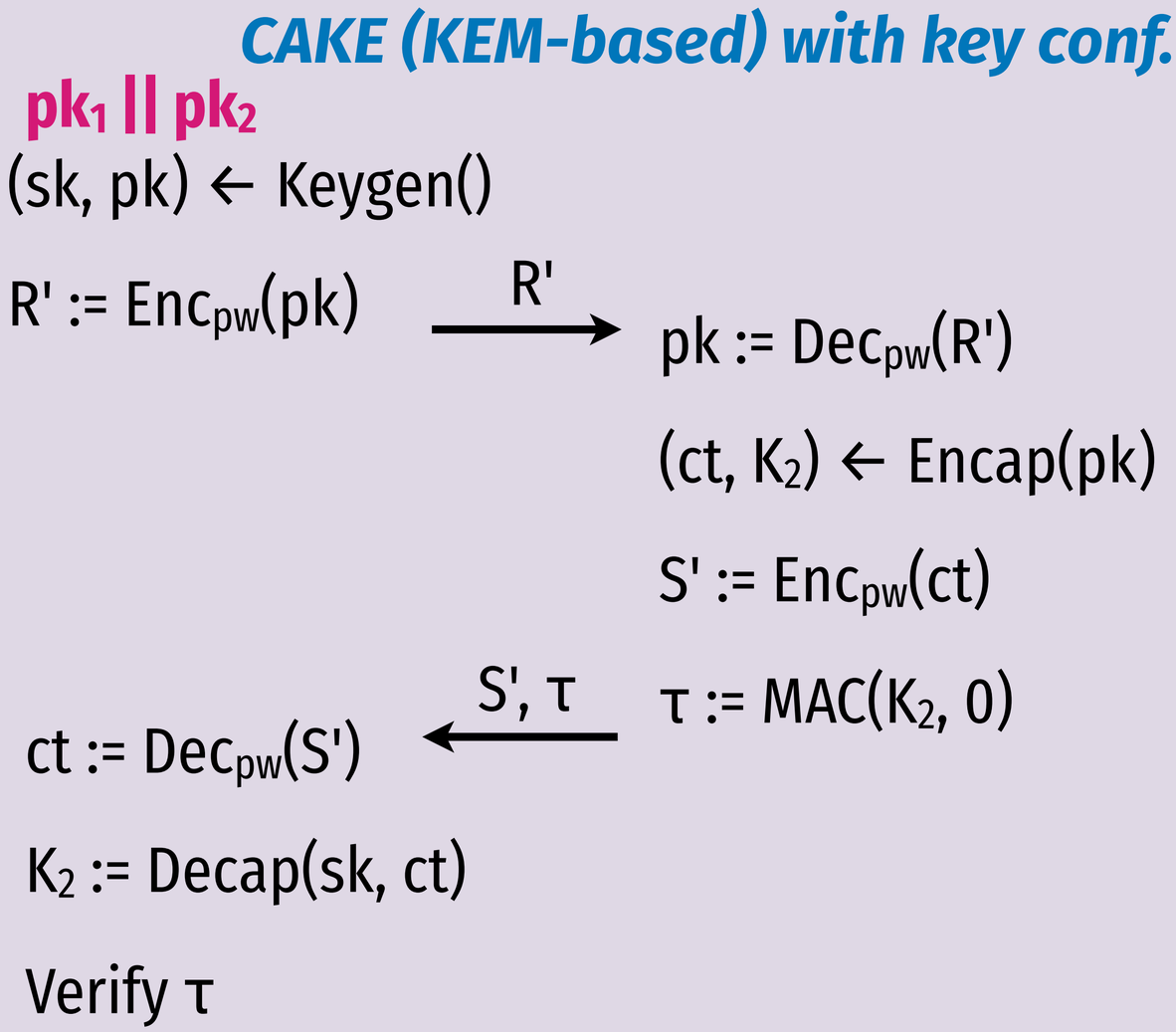
CAKE is KEM-based. Just use a **hybrid KEM**

CAKE (KEM-based) with key conf.



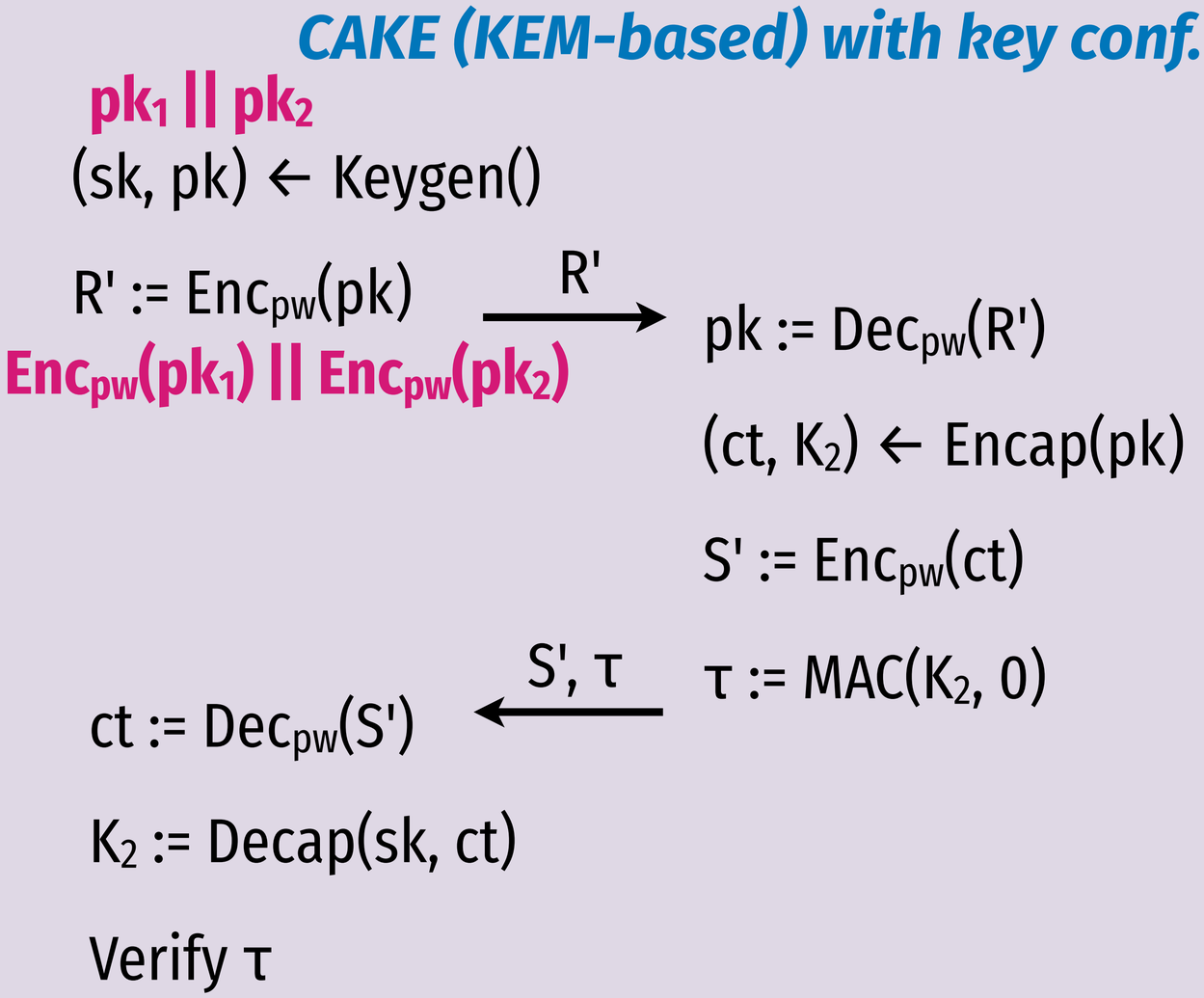
How else not to make a hybrid PAKE

CAKE is KEM-based. Just use a **hybrid KEM**



How else not to make a hybrid PAKE

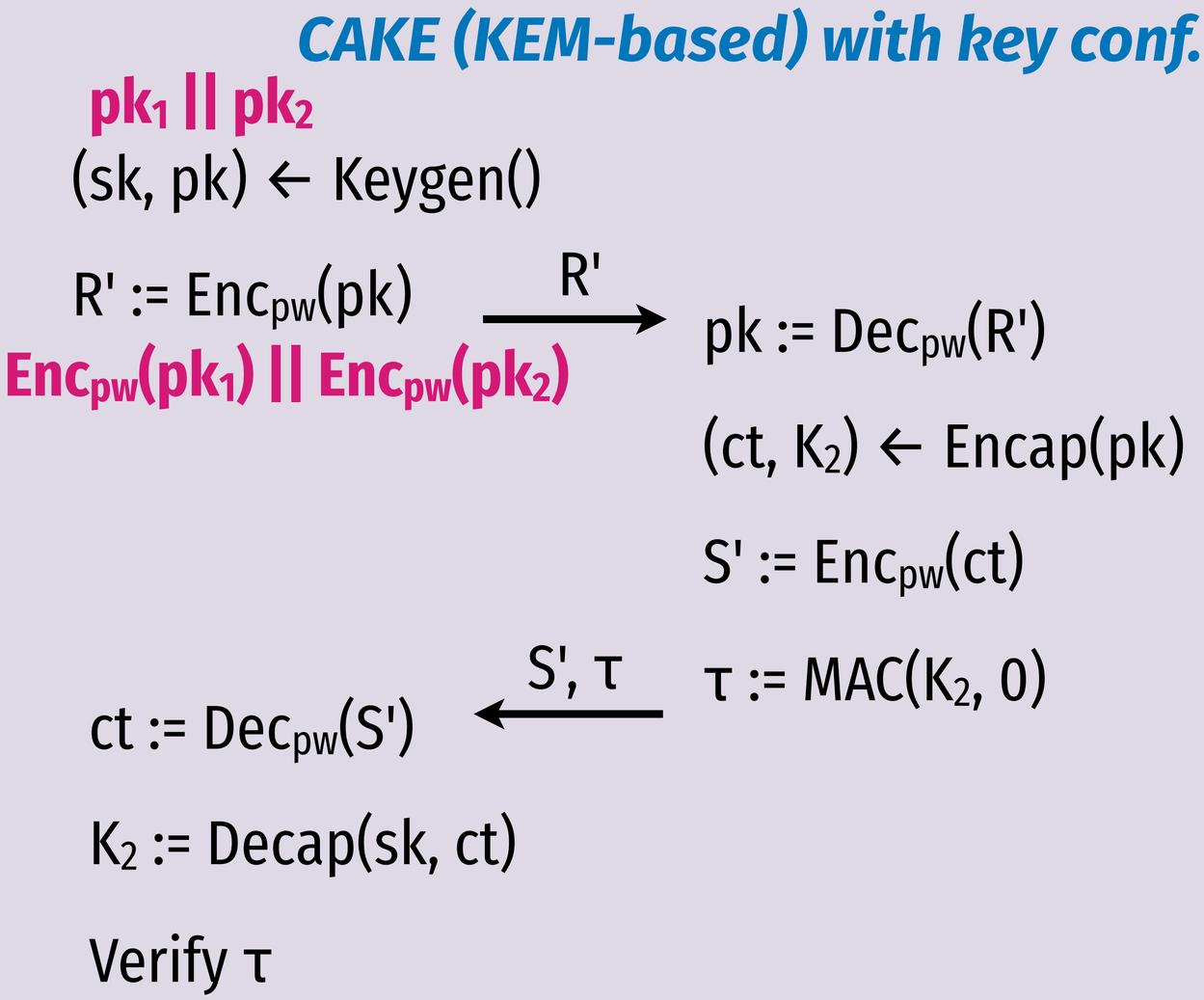
CAKE is KEM-based. Just use a **hybrid KEM**



How else not to make a hybrid PAKE

CAKE is KEM-based. Just use a **hybrid KEM**

Suppose pk_1 is an LWE sample

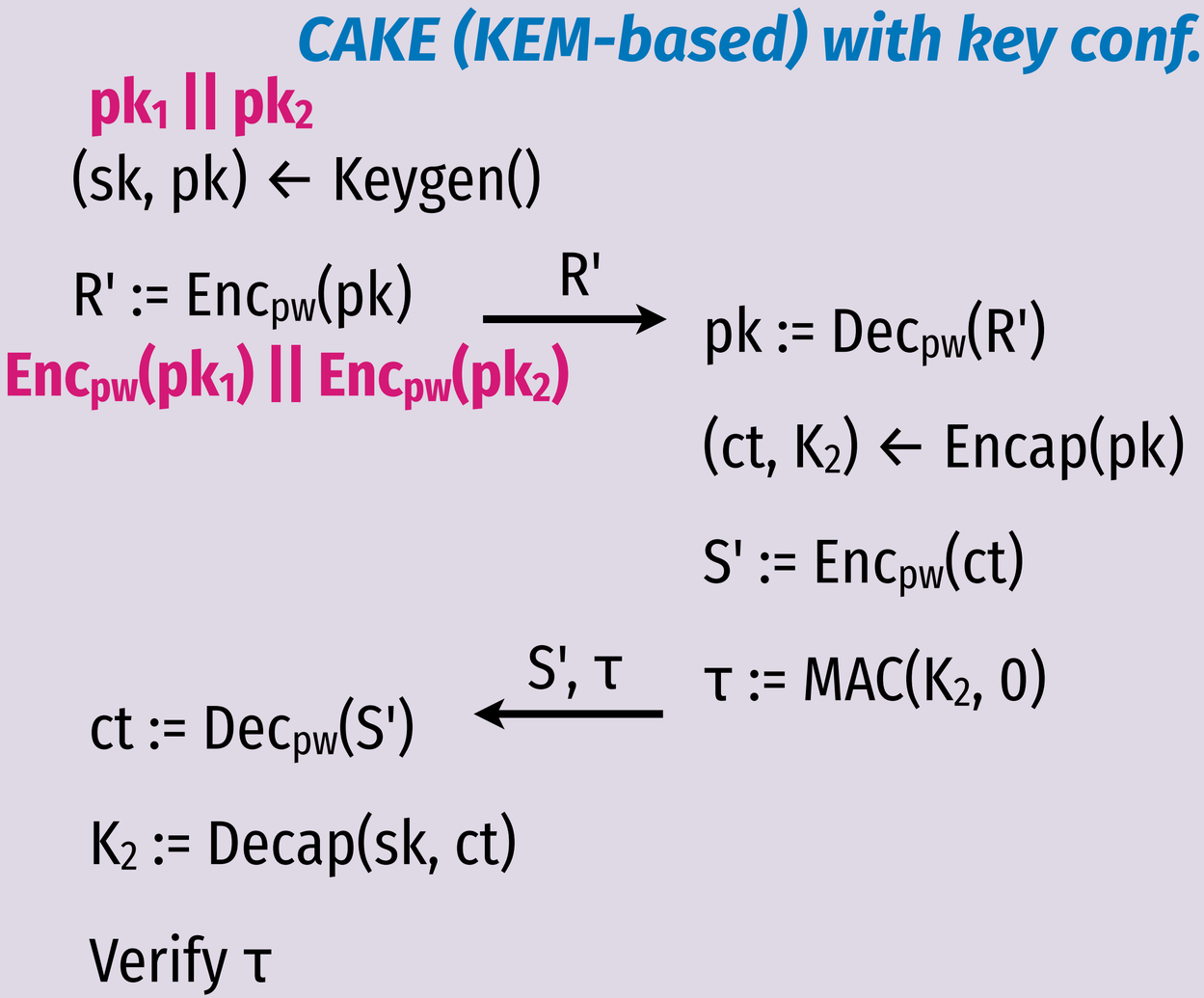


How else not to make a hybrid PAKE

CAKE is KEM-based. Just use a **hybrid KEM**

Suppose pk_1 is an LWE sample

With an LWE oracle, $Enc_{pw}(pk_1)$ lets you guess/check pw



How else not to make a hybrid PAKE

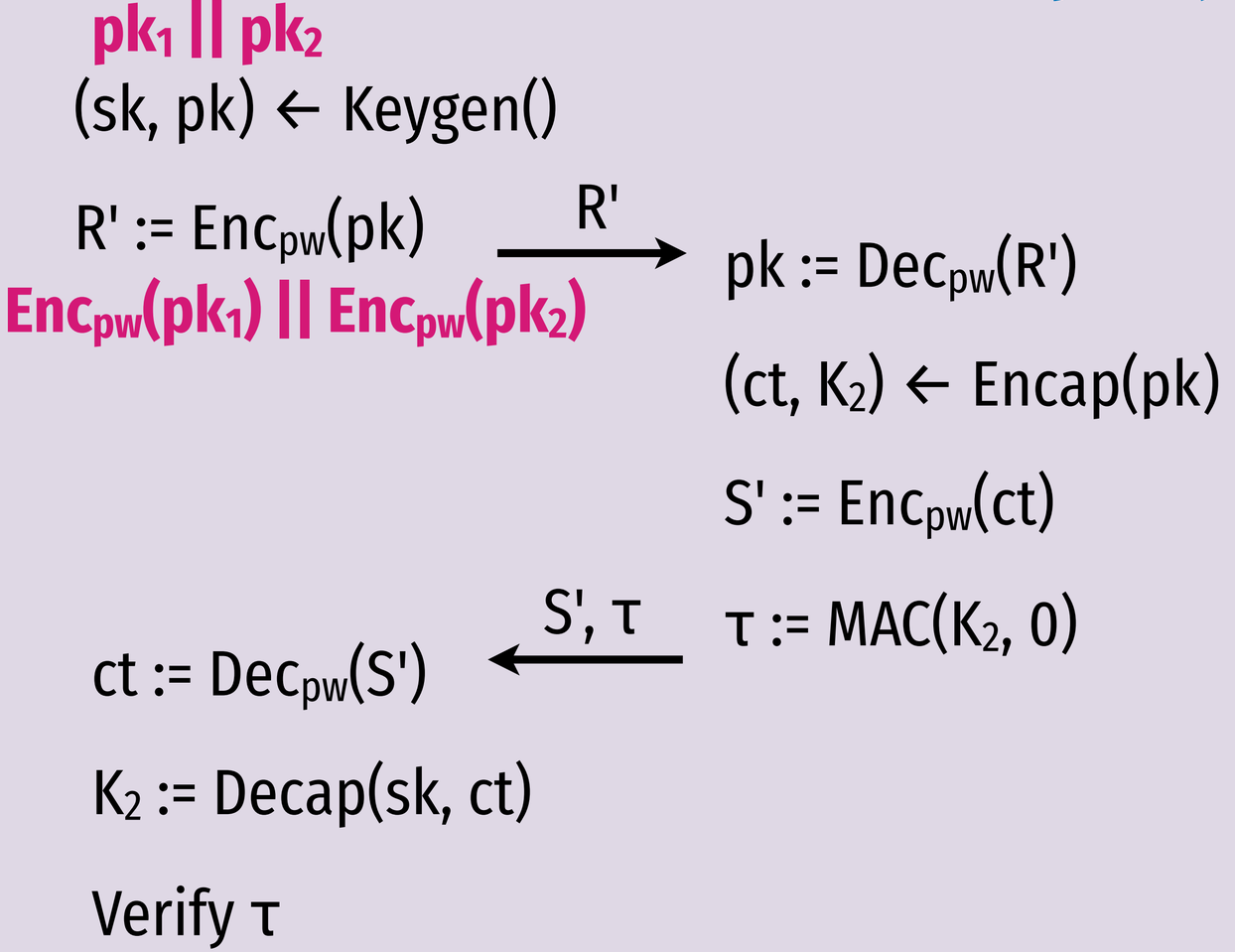
CAKE is KEM-based. Just use a **hybrid KEM**

Suppose pk_1 is an LWE sample

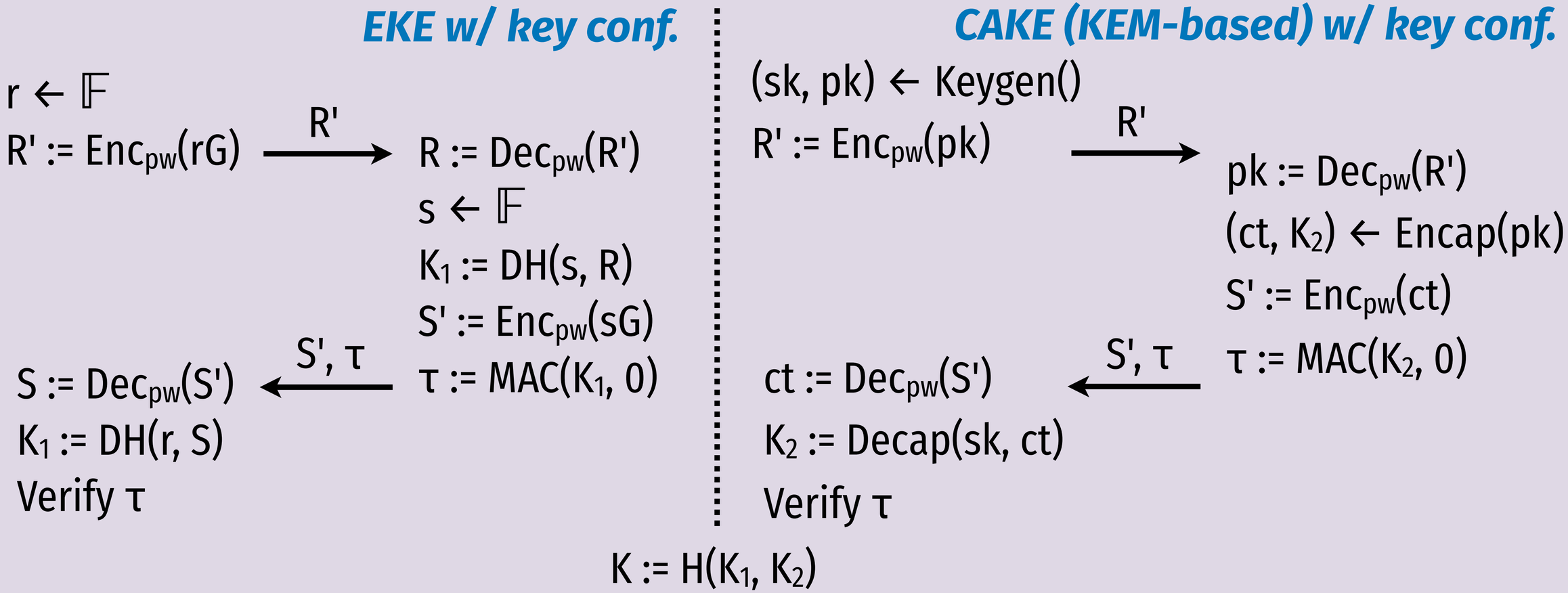
With an LWE oracle, $Enc_{pw}(pk_1)$ lets you guess/check pw

This hybrid is only as secure as LWE!

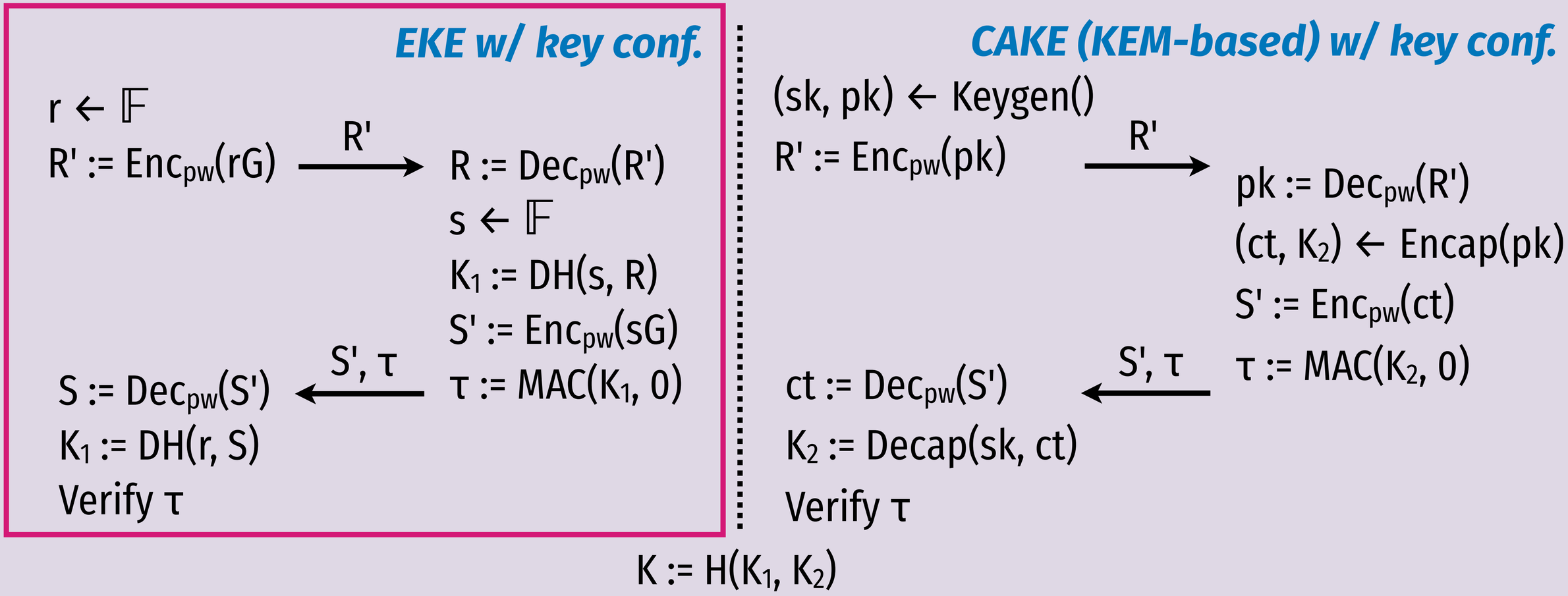
CAKE (KEM-based) with key conf.



Building ParComb

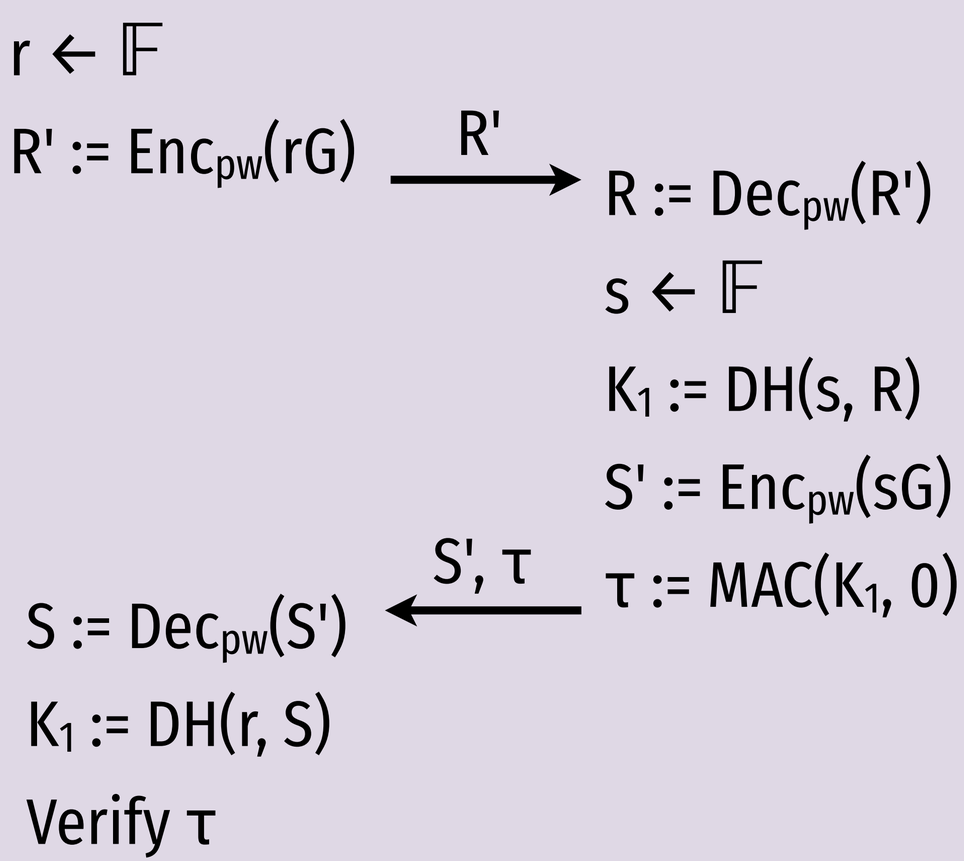


Building ParComb



Building ParComb

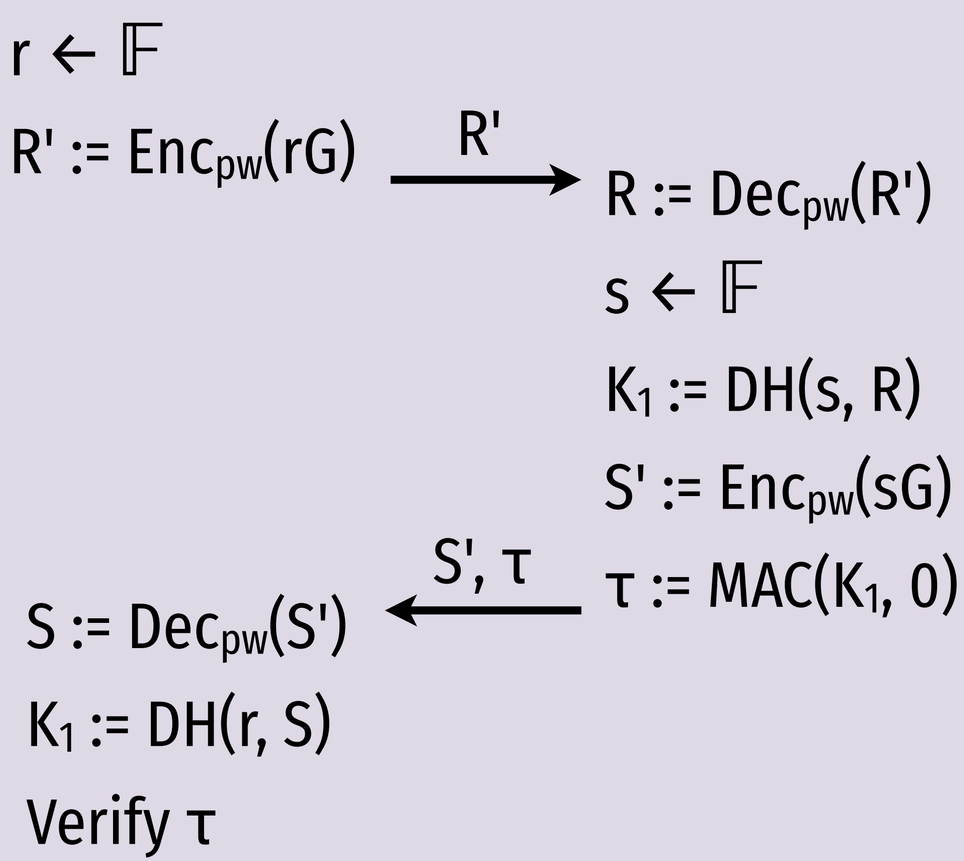
EKE w/ key conf.



Building ParComb

Observation: it seems **τ was the only issue** with the parallel example

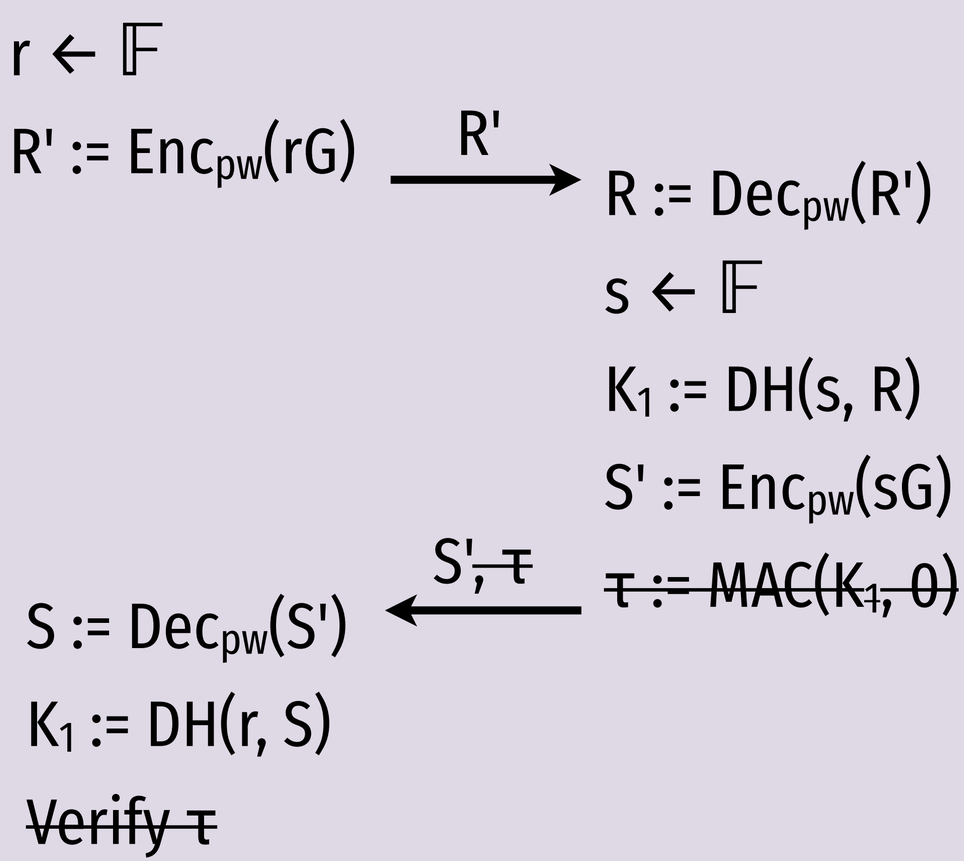
EKE w/ key conf.



Building ParComb

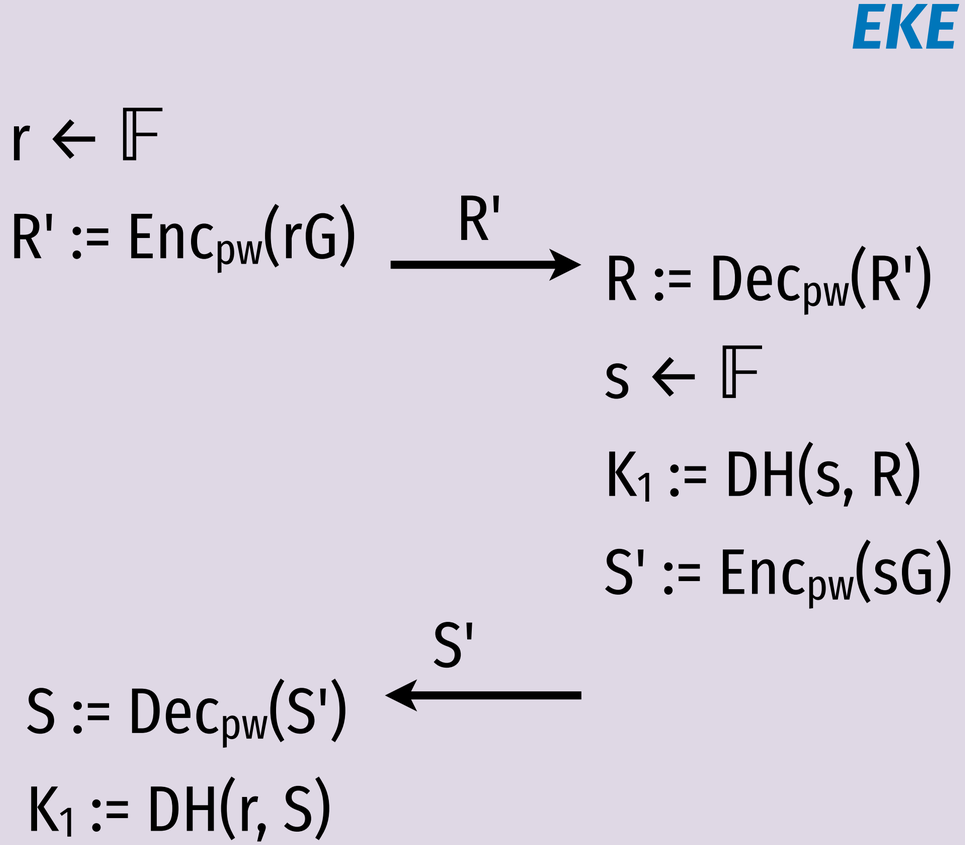
Observation: it seems **τ was the only issue** with the parallel example

EKE ~~w/ key conf.~~



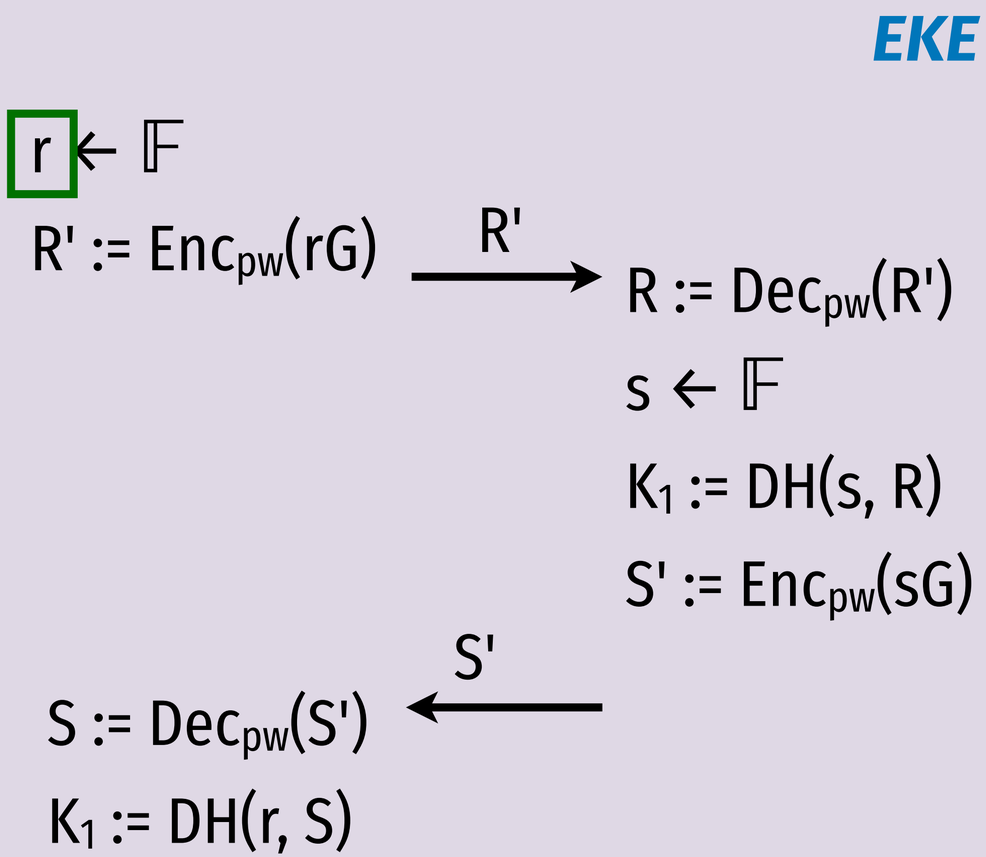
Building ParComb

Observation: it seems **τ was the only issue** with the parallel example



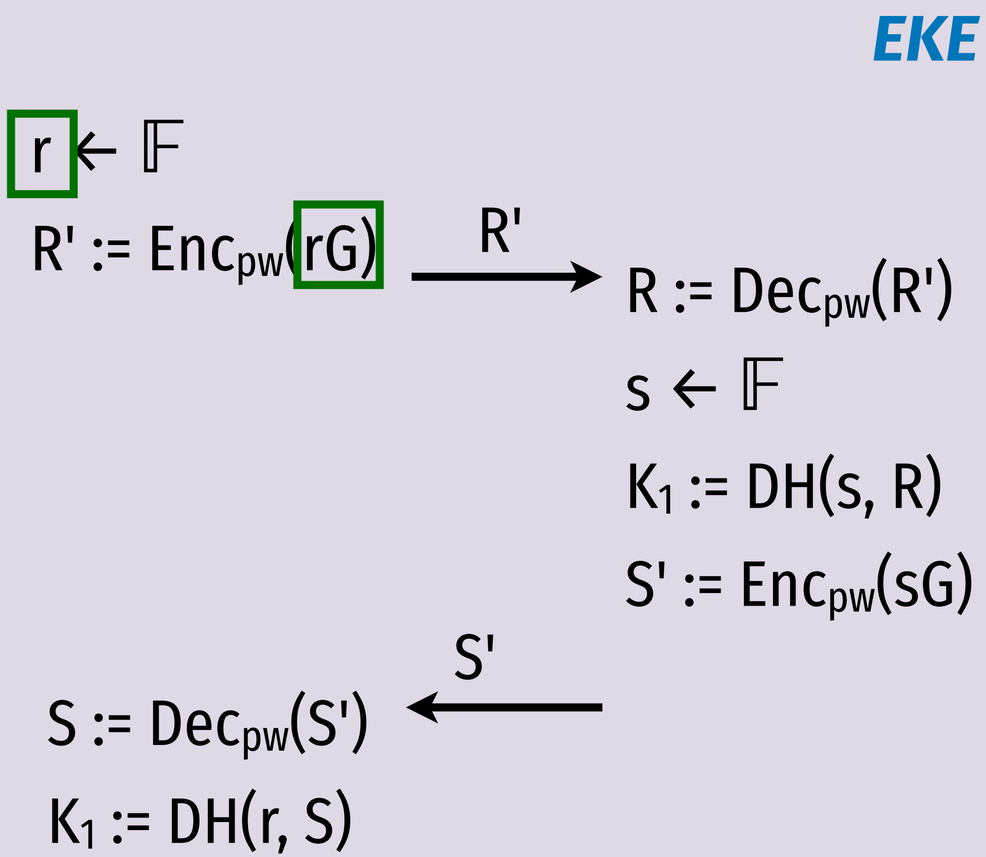
Building ParComb

Observation: it seems **τ was the only issue** with the parallel example



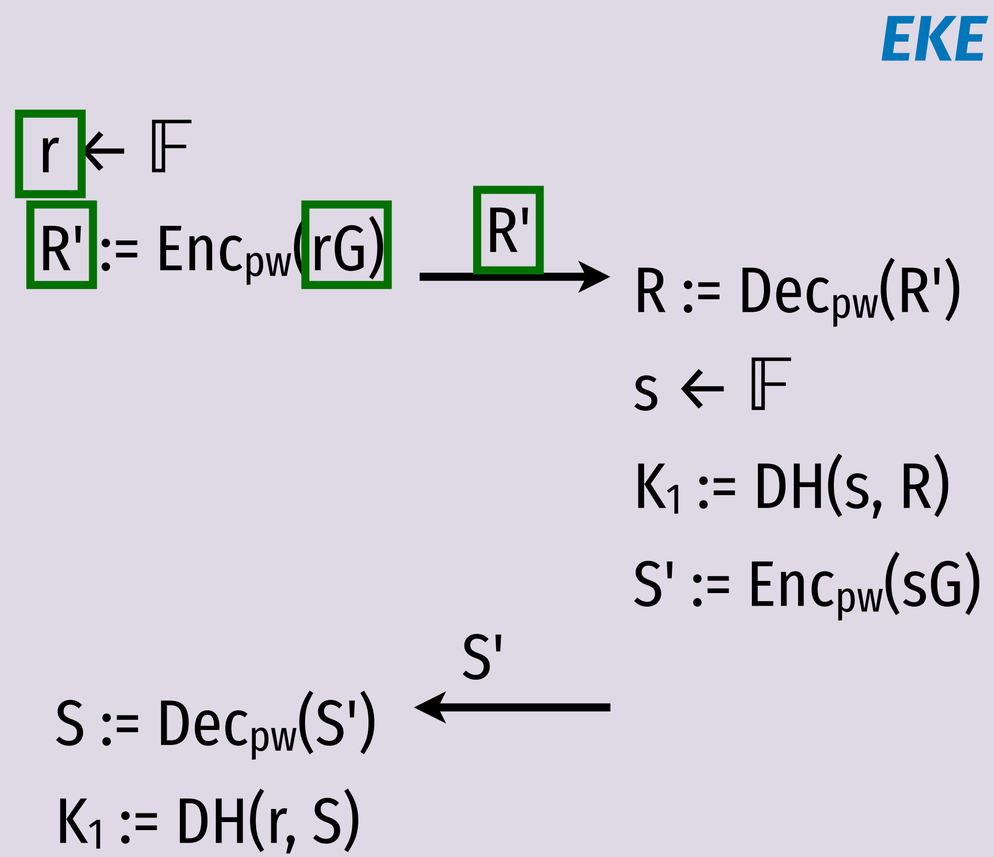
Building ParComb

Observation: it seems **τ was the only issue** with the parallel example



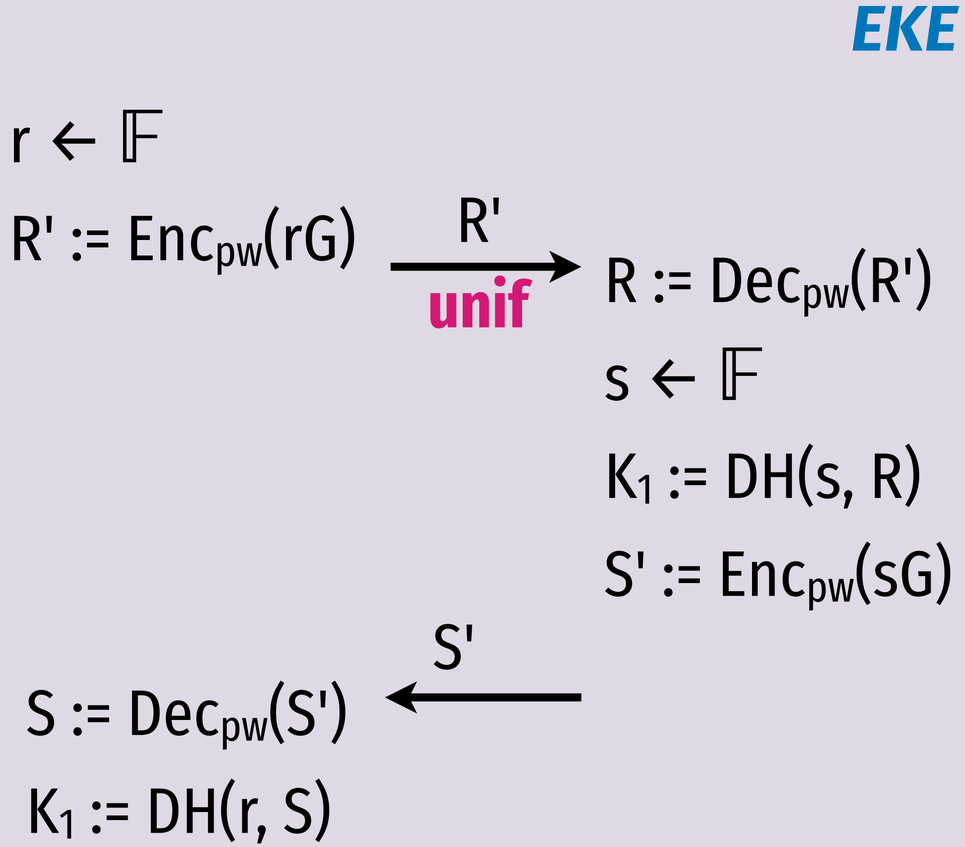
Building ParComb

Observation: it seems **τ was the only issue** with the parallel example



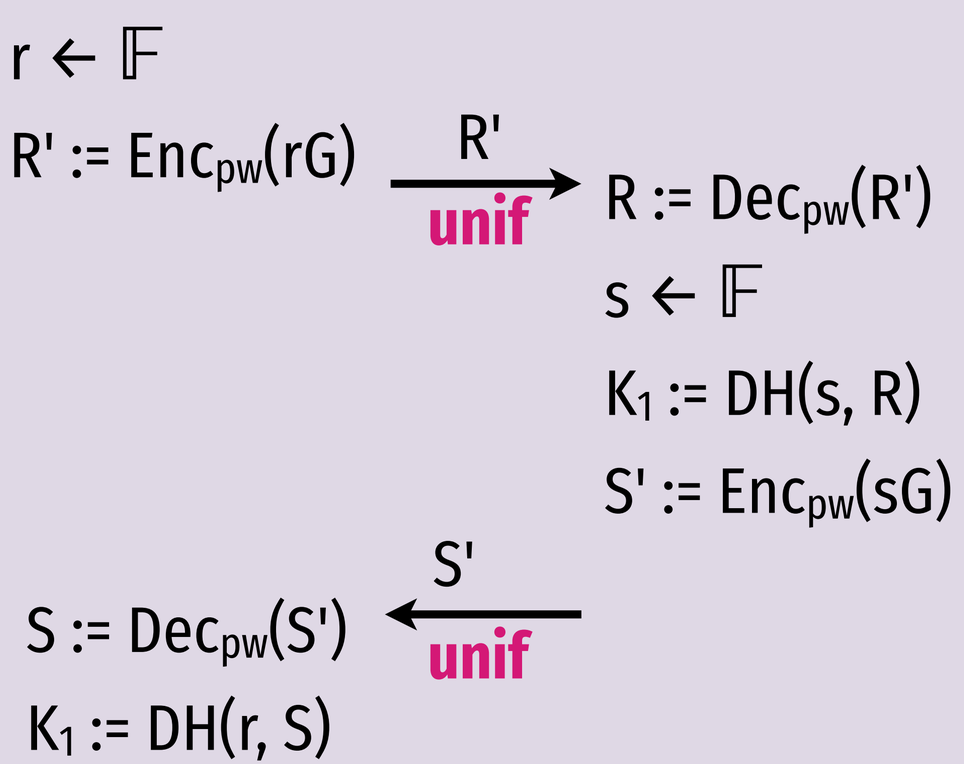
Building ParComb

Observation: it seems **τ was the only issue** with the parallel example



Building ParComb

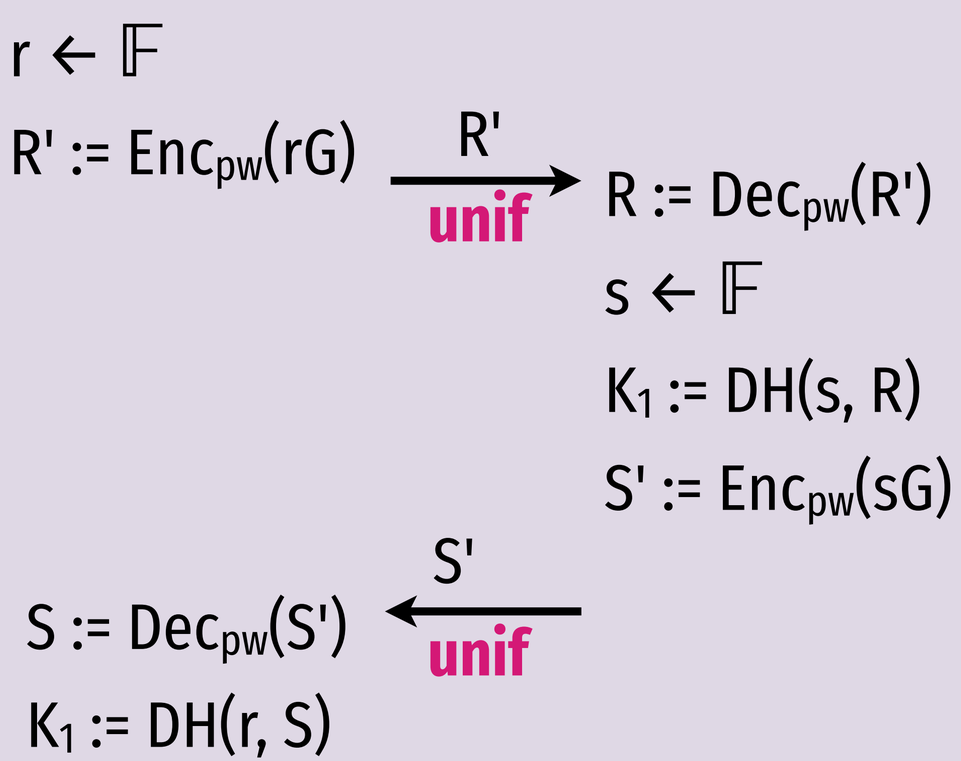
Observation: it seems **τ was the only issue** with the parallel example



Building ParComb

Observation: it seems **τ was the only issue** with the parallel example

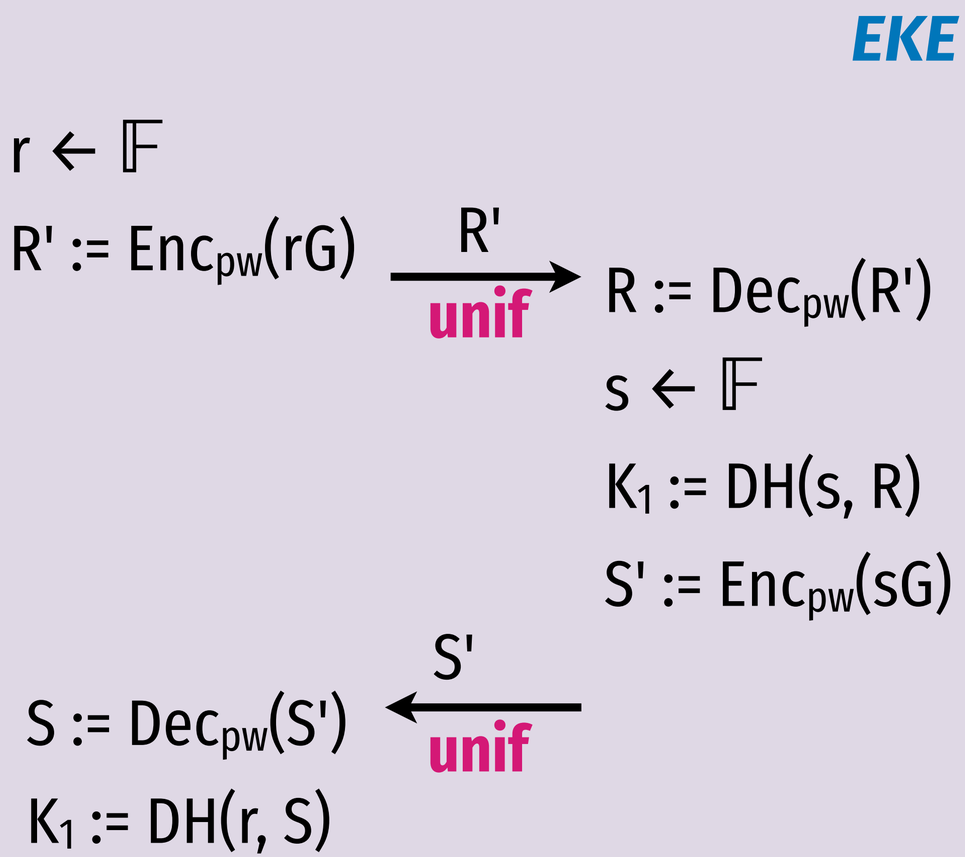
We'll call this **statistical password hiding**, or (full) DH-type PAKE



Building ParComb

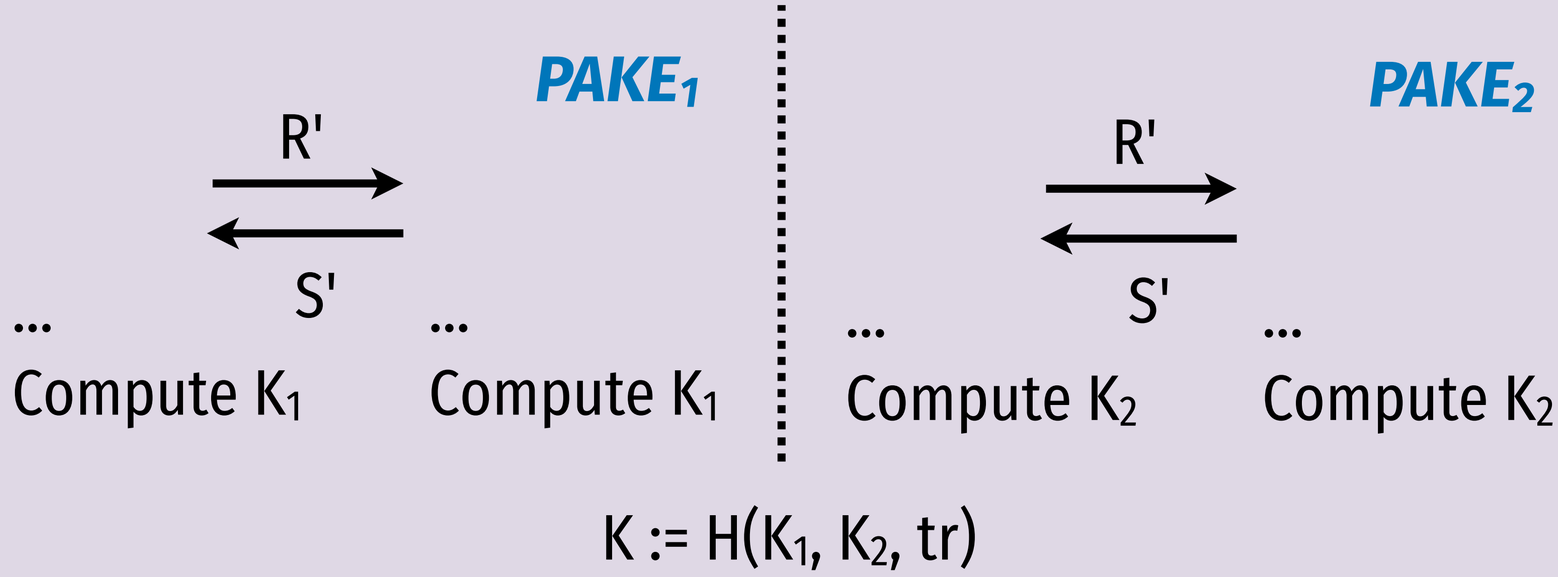
Observation: it seems **τ was the only issue** with the parallel example

We'll call this **statistical password hiding**, or (full) DH-type PAKE



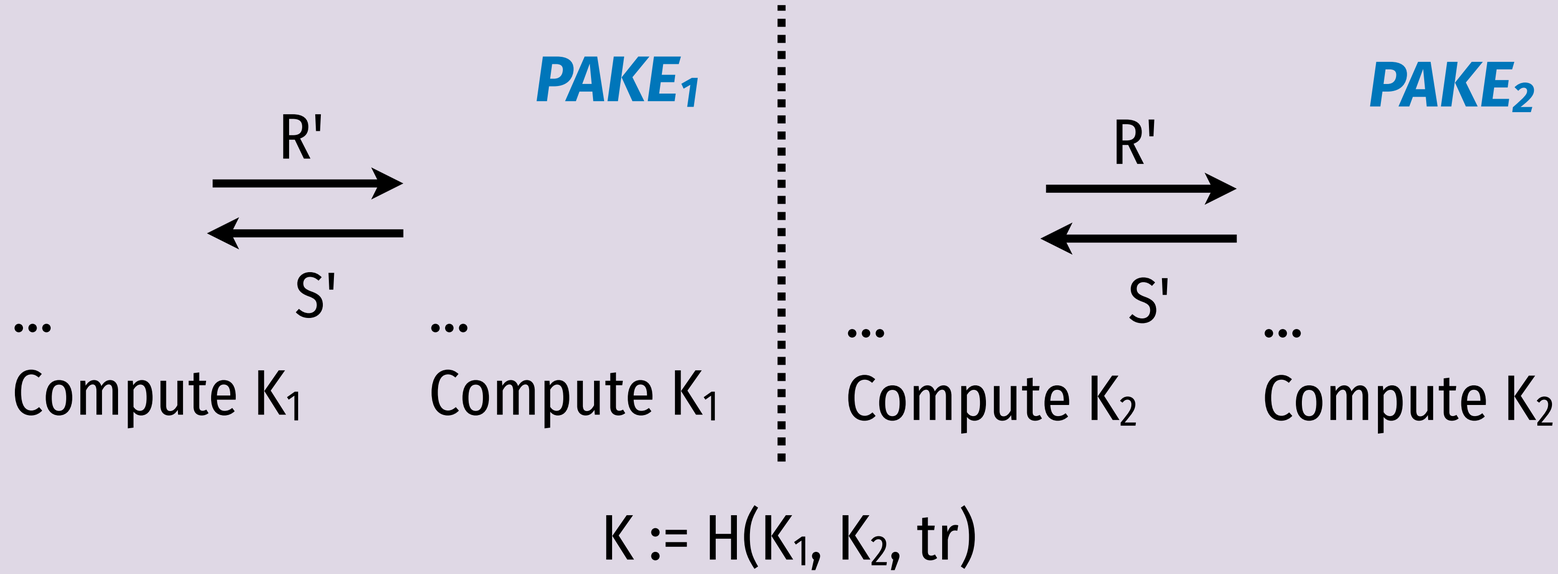
EKE, SPAKE2, CPace are stat. pw-hiding

Building ParComb



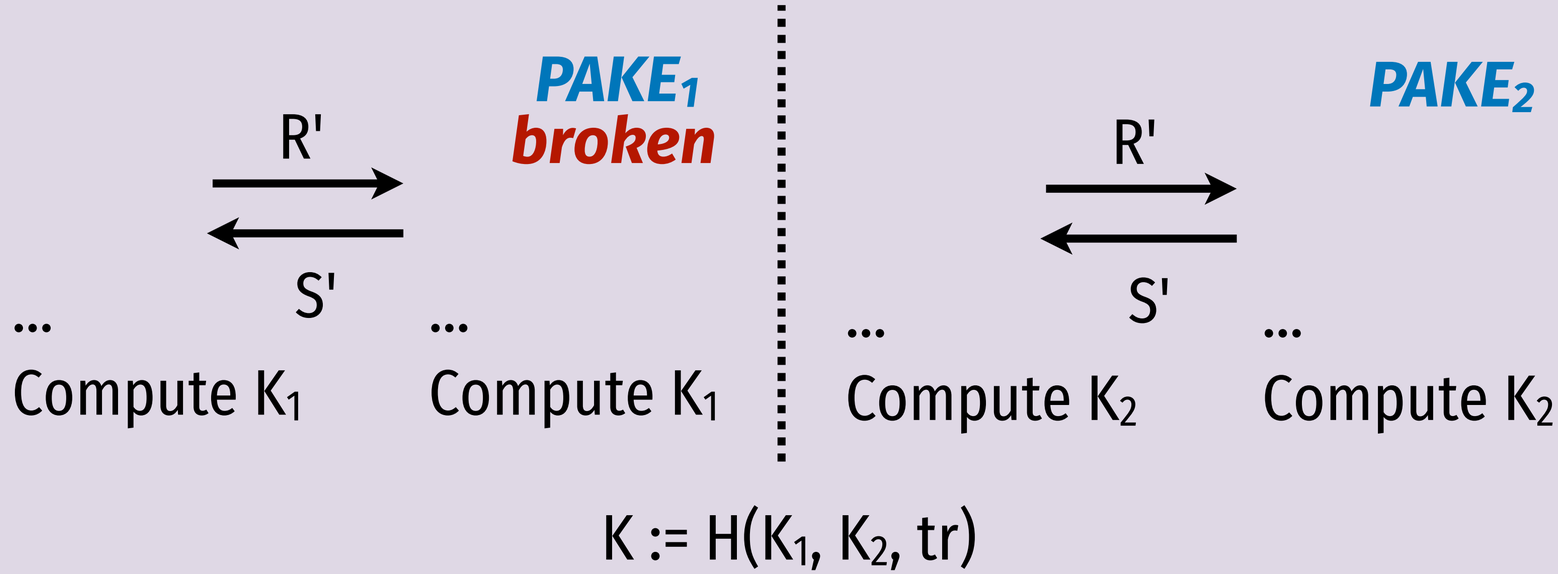
Building ParComb

Requirement: $PAKE_1$ and $PAKE_2$ are statistically password-hiding



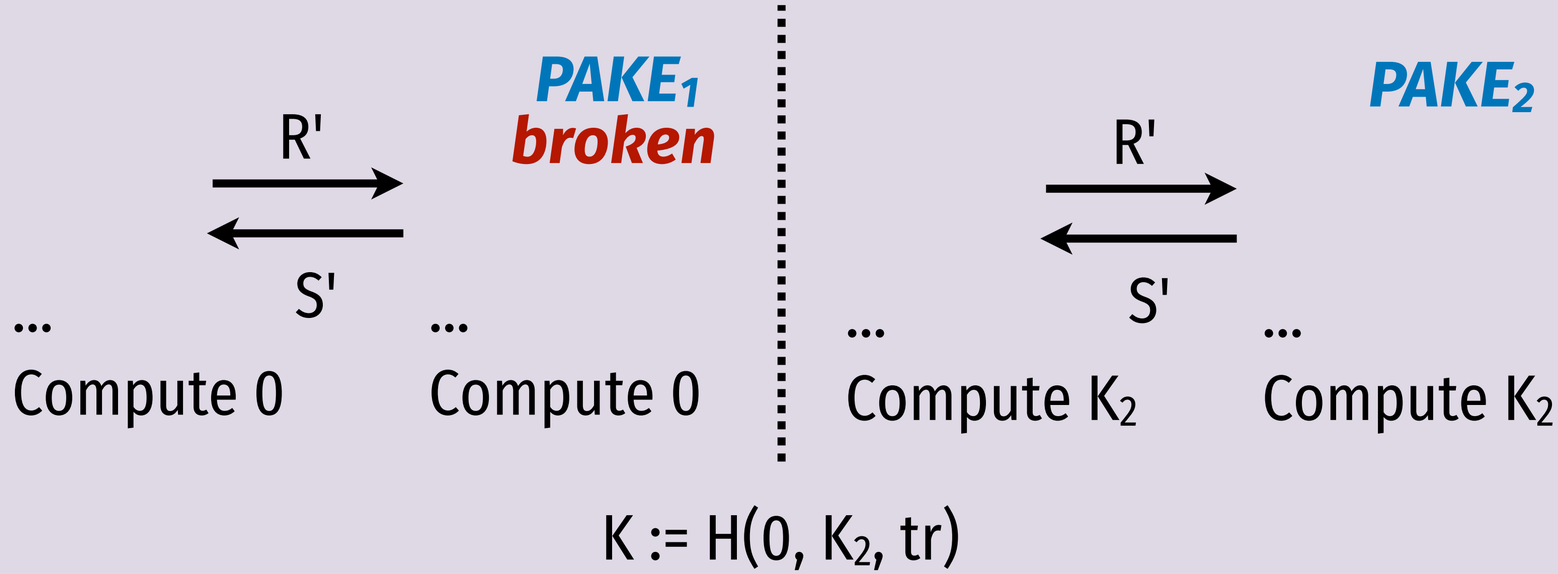
Building ParComb

Requirement: $PAKE_1$ and $PAKE_2$ are statistically password-hiding



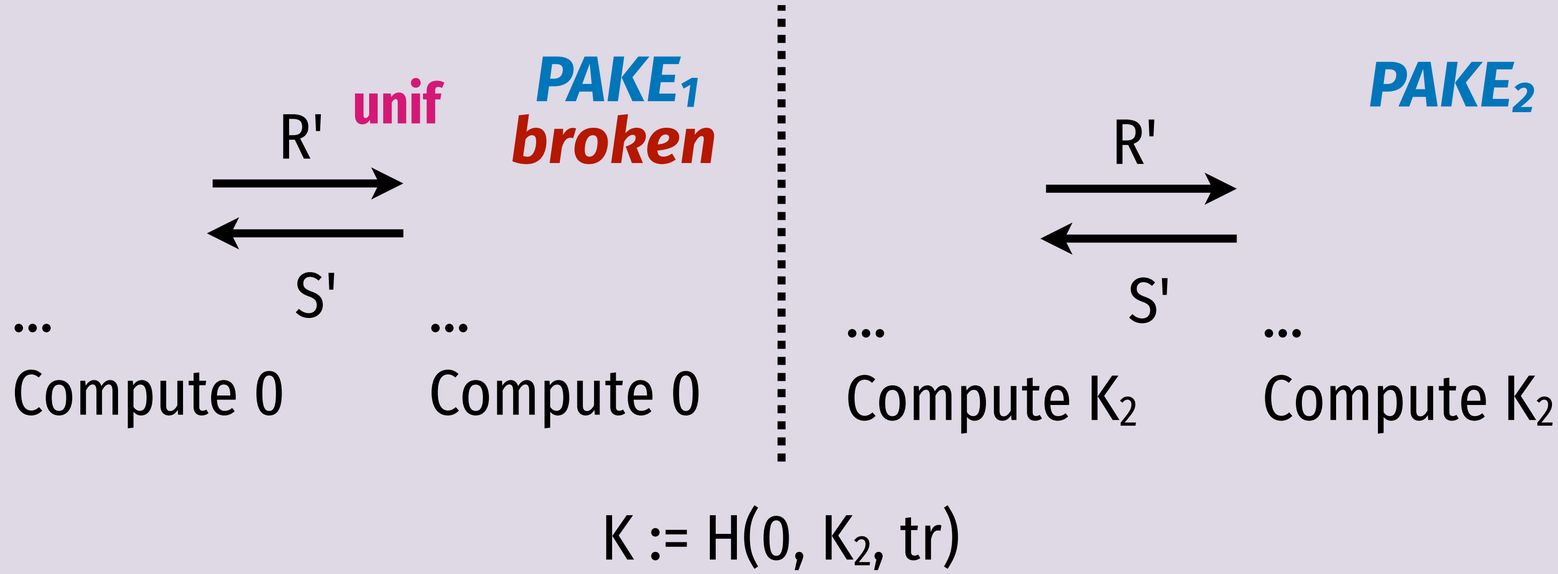
Building ParComb

Requirement: $PAKE_1$ and $PAKE_2$ are statistically password-hiding



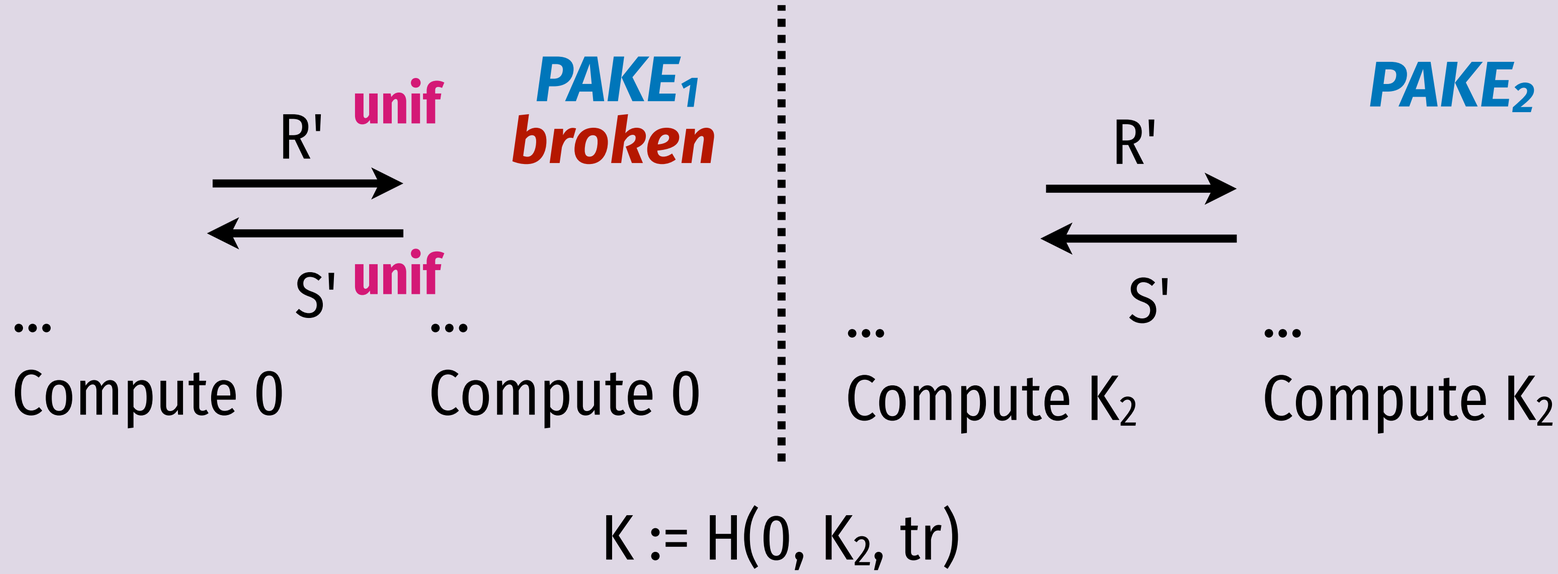
Building ParComb

Requirement: $PAKE_1$ and $PAKE_2$ are statistically password-hiding



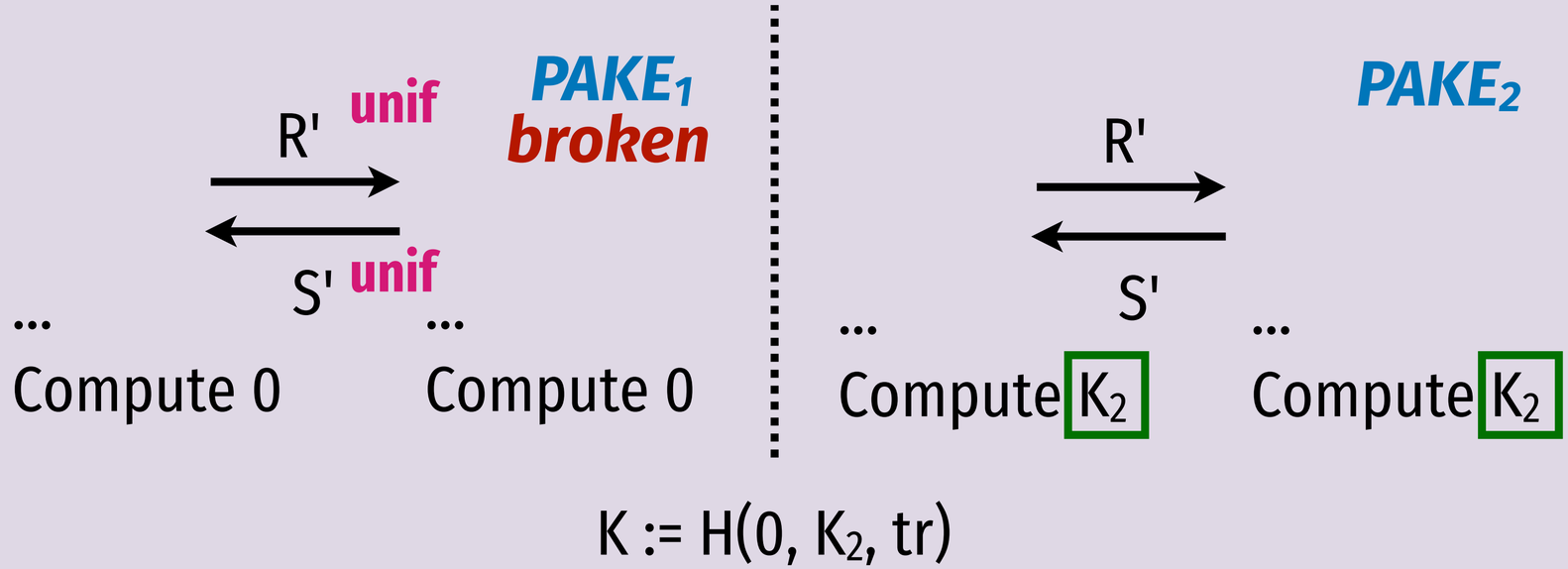
Building ParComb

Requirement: $PAKE_1$ and $PAKE_2$ are statistically password-hiding



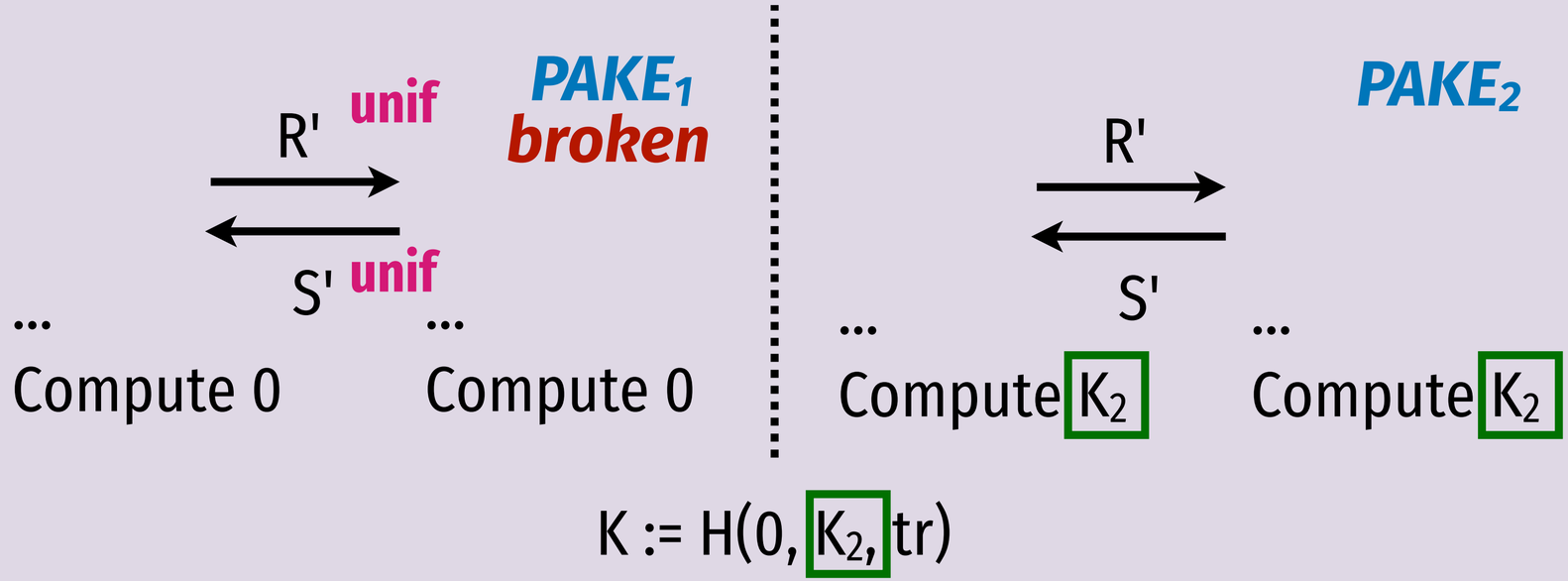
Building ParComb

Requirement: $PAKE_1$ and $PAKE_2$ are statistically password-hiding



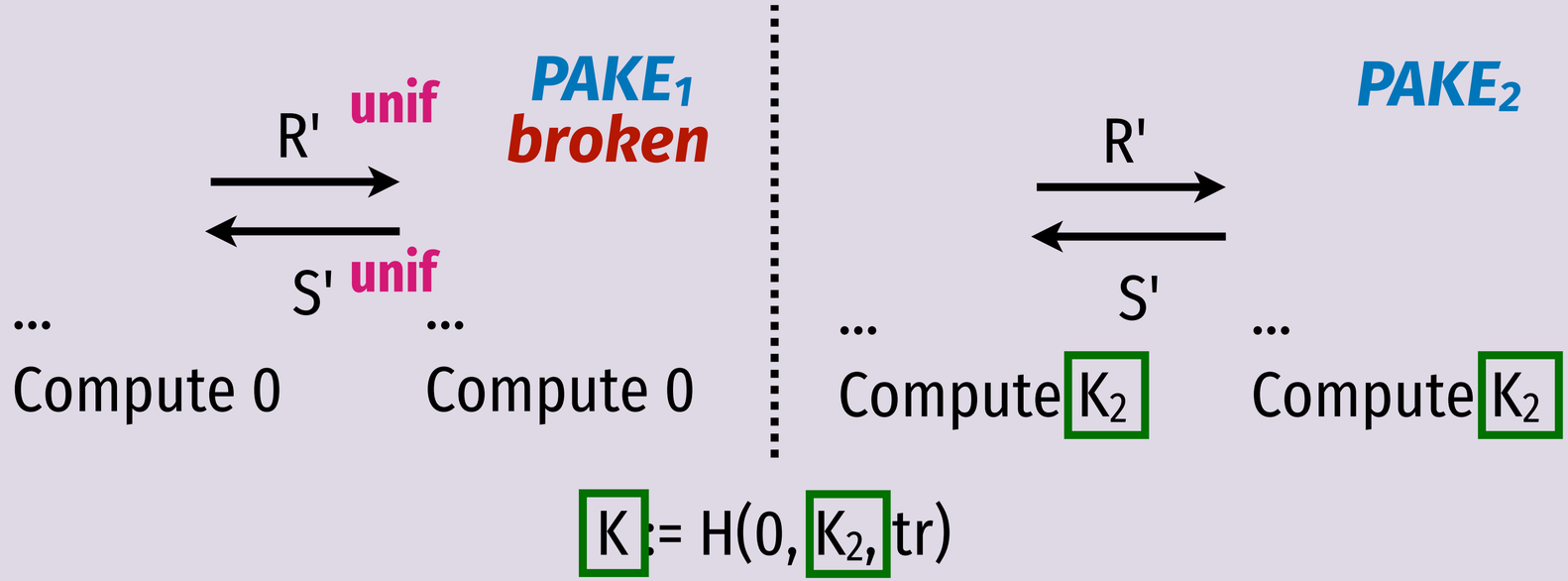
Building ParComb

Requirement: $PAKE_1$ and $PAKE_2$ are statistically password-hiding



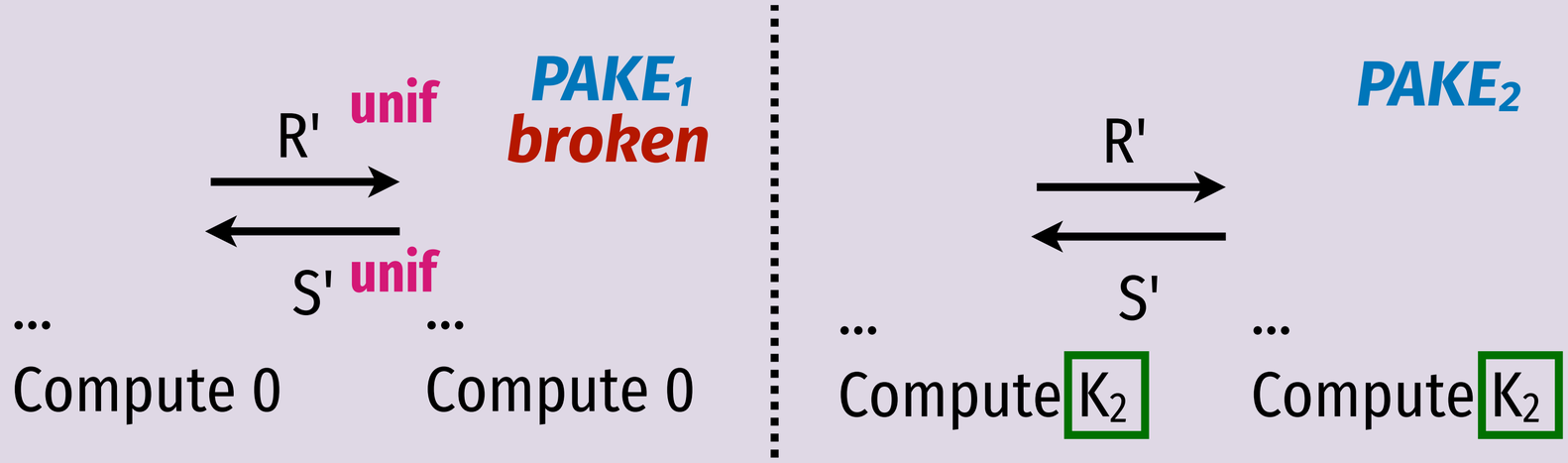
Building ParComb

Requirement: $PAKE_1$ and $PAKE_2$ are statistically password-hiding



Building ParComb

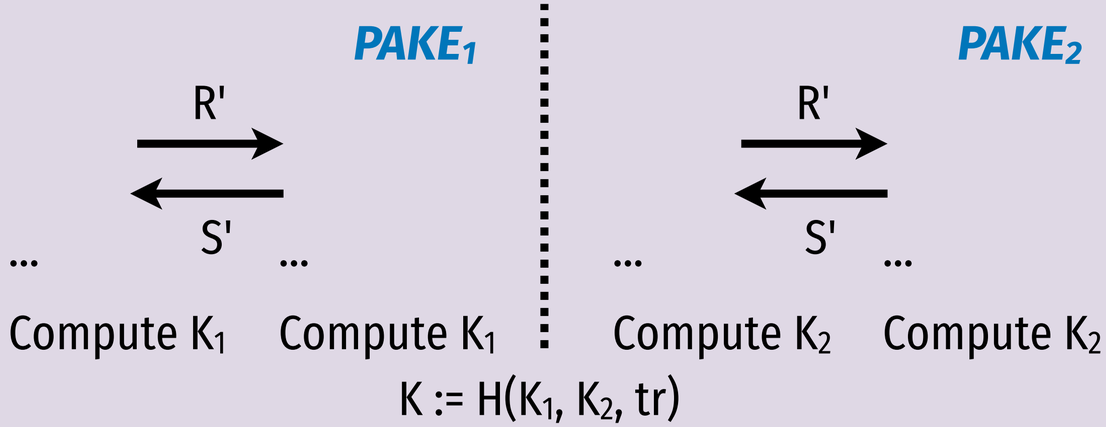
Requirement: $PAKE_1$ and $PAKE_2$ are statistically password-hiding



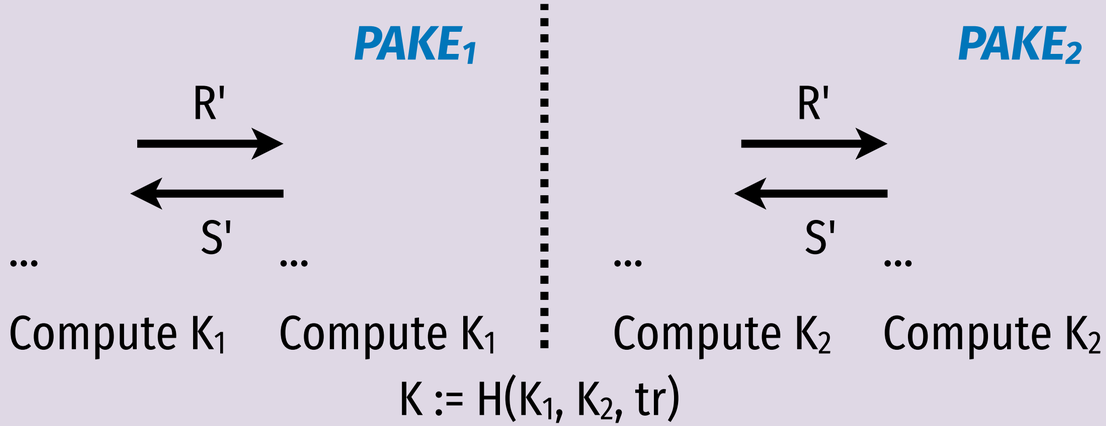
$K := H(0, K_2, tr)$

PAKE is still secure

Properties of ParComb

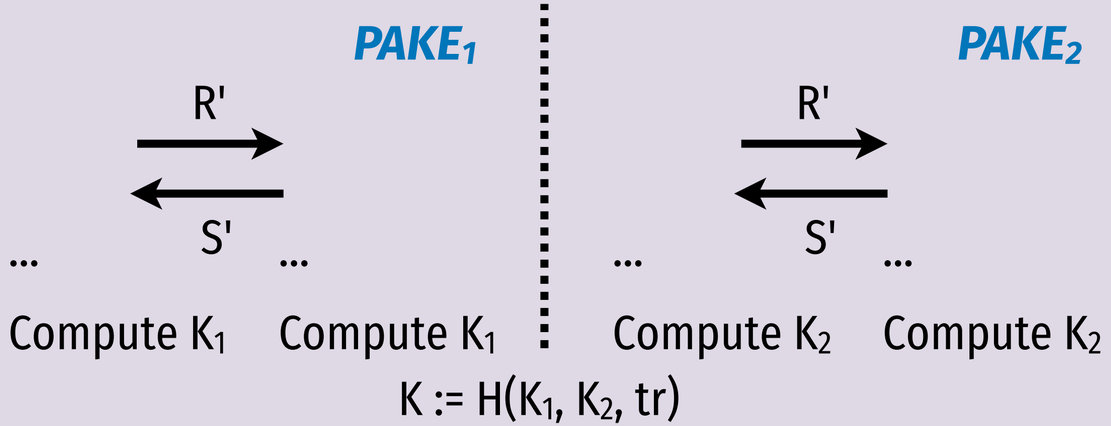


Properties of ParComb



Theorem 1: Let PAKE_1 and PAKE_2 be 1-round statistically password-hiding, then

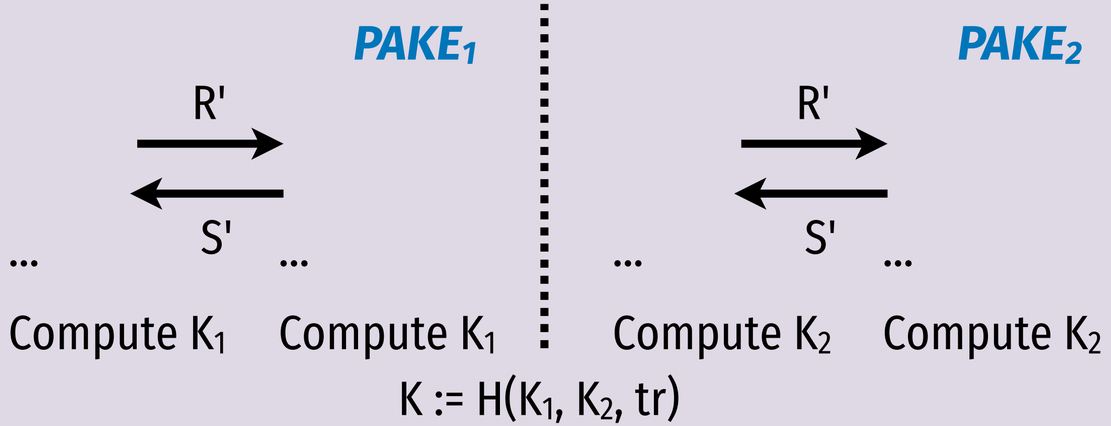
Properties of ParComb



Theorem 1: Let PAKE_1 and PAKE_2 be 1-round statistically password-hiding, then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

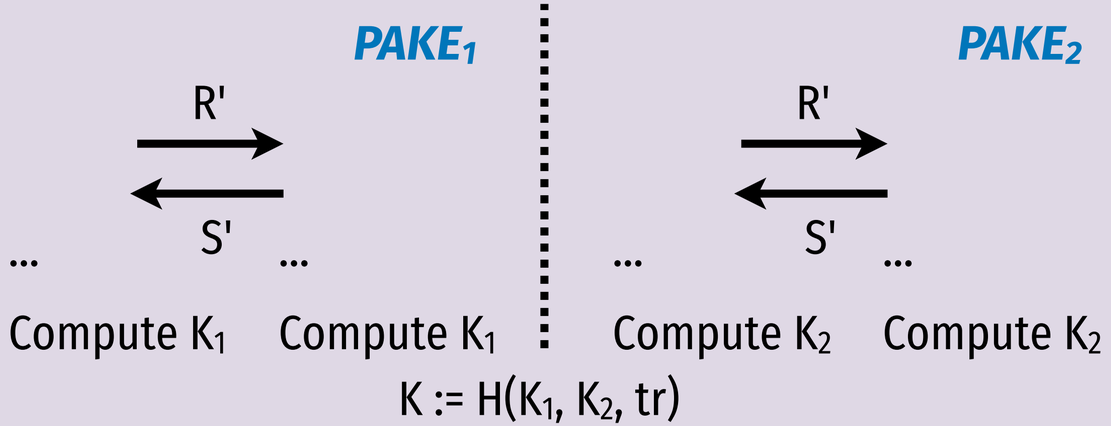
Properties of ParComb



Theorem 1: Let PAKE_1 and PAKE_2 be 1-round statistically password-hiding, then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

Properties of ParComb

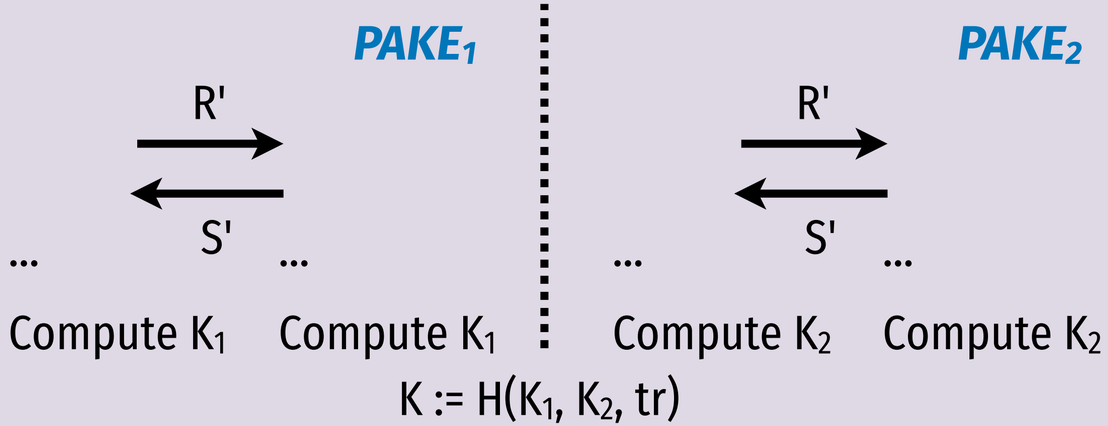


Theorem 1: Let PAKE_1 and PAKE_2 be 1-round statistically password-hiding, then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

EKE

Properties of ParComb



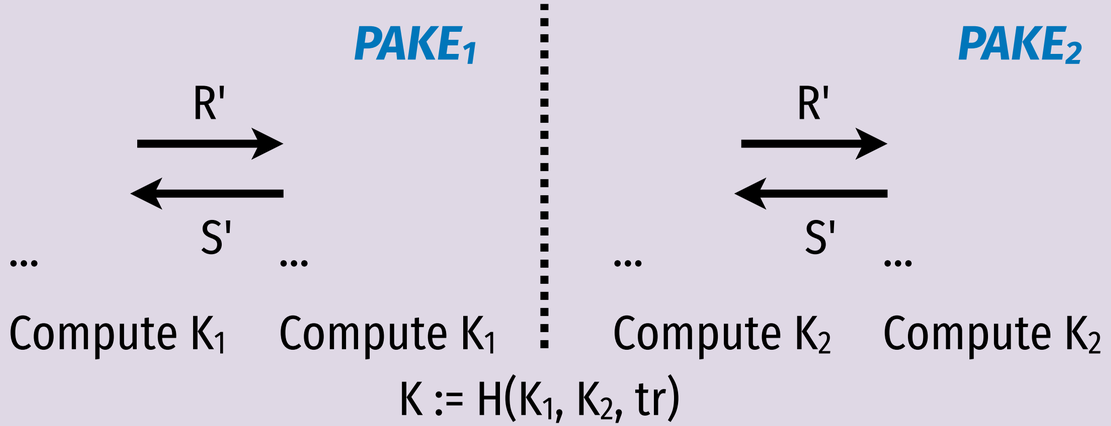
Theorem 1: Let PAKE_1 and PAKE_2 be 1-round statistically password-hiding, then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

EKE

No 1-round statistically password-hiding PQ scheme realizing $\mathcal{F}_{\mathrm{PAKE}}$ exists

Properties of ParComb



Theorem 1: Let PAKE_1 and PAKE_2 be 1-round statistically password-hiding, then

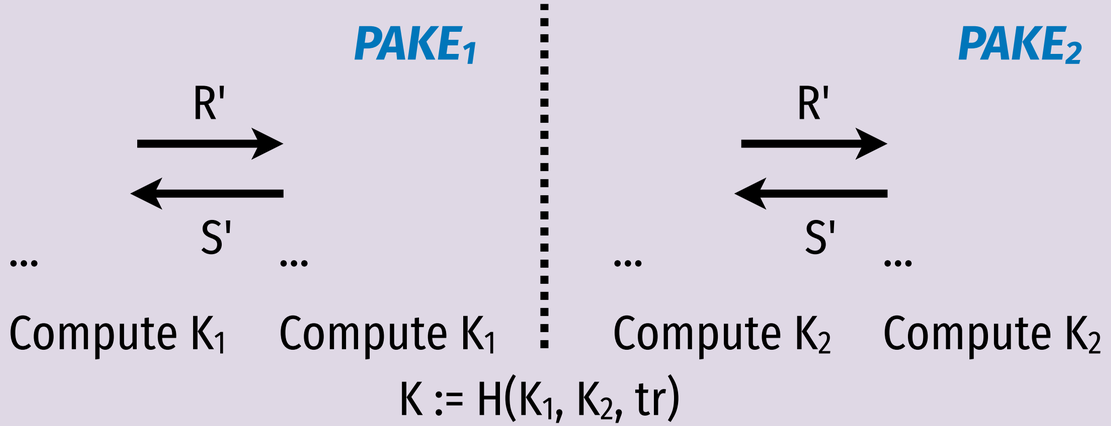
1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

EKE

Theorem 2:

No 1-round statistically password-hiding PQ scheme realizing $\mathcal{F}_{\mathrm{PAKE}}$ exists

Properties of ParComb



Theorem 1: Let PAKE_1 and PAKE_2 be 1-round statistically password-hiding, then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

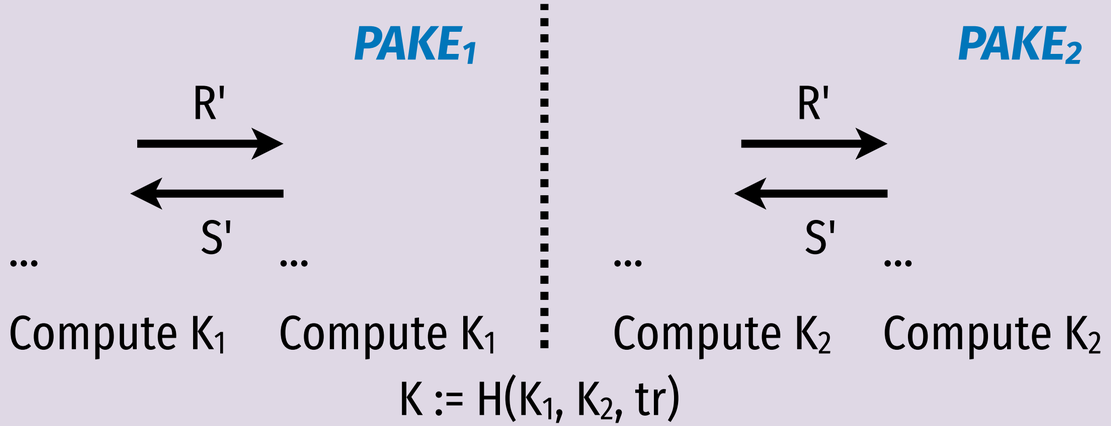
EKE

Theorem 2:

No 1-round statistically password-hiding PQ scheme realizing $\mathcal{F}_{\mathrm{PAKE}}$ exists

1. **PAKE_1** is $\mathcal{F}_{\mathbf{lePAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathbf{lePAKE}}$

Properties of ParComb



Theorem 1: Let PAKE_1 and PAKE_2 be 1-round statistically password-hiding, then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

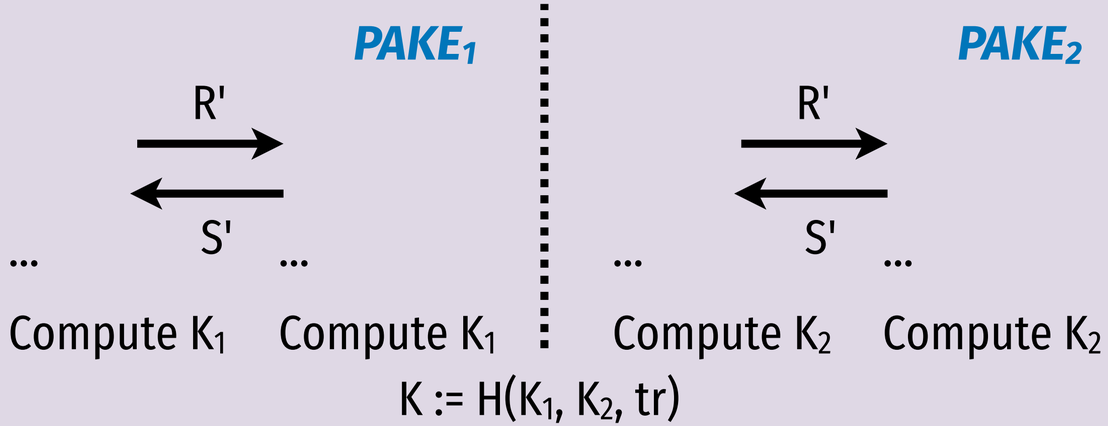
EKE

Theorem 2:

No 1-round statistically password-hiding PQ scheme realizing $\mathcal{F}_{\mathrm{PAKE}}$ exists

1. **PAKE_1** is $\mathcal{F}_{\mathbf{lePAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathbf{lePAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathbf{lePAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathbf{lePAKE}}$

Properties of ParComb



Theorem 1: Let PAKE_1 and PAKE_2 be 1-round statistically password-hiding, then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

EKE

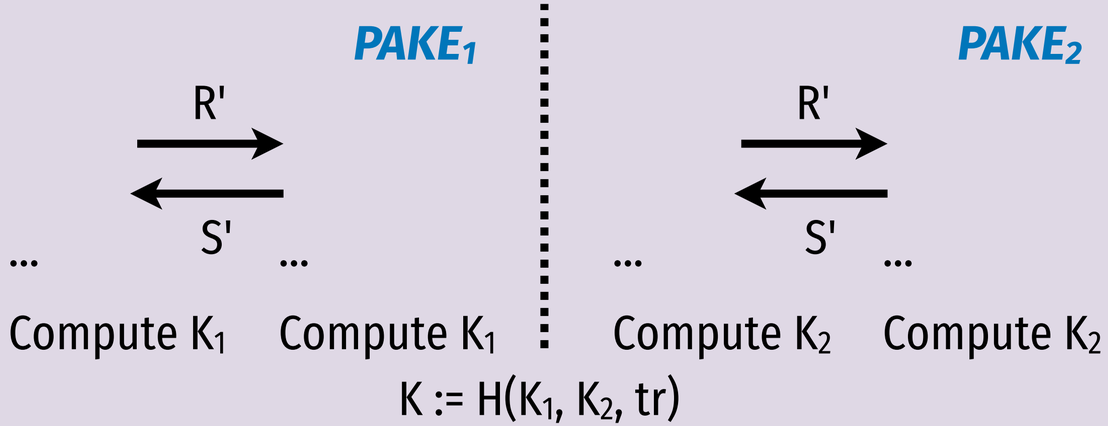
No 1-round statistically password-hiding PQ scheme realizing $\mathcal{F}_{\mathrm{PAKE}}$ exists

Theorem 2:

1. **PAKE_1** is $\mathcal{F}_{\mathbf{lePAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathbf{lePAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathbf{lePAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathbf{lePAKE}}$

SPAKE2, CPace

Properties of ParComb



Theorem 1: Let PAKE_1 and PAKE_2 be 1-round statistically password-hiding, then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

EKE

No 1-round statistically password-hiding PQ scheme realizing $\mathcal{F}_{\mathrm{PAKE}}$ exists

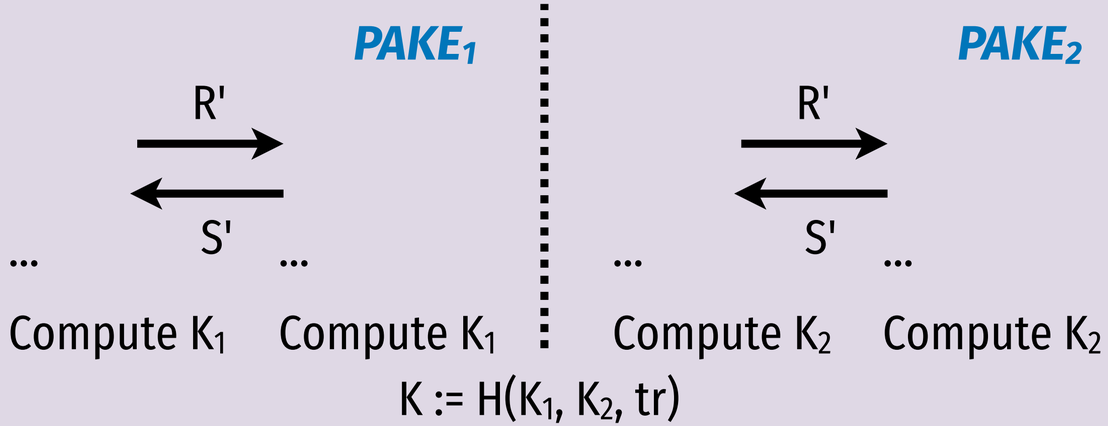
Theorem 2:

1. **PAKE_1** is $\mathcal{F}_{\mathbf{lePAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathbf{lePAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathbf{lePAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathbf{lePAKE}}$

SPAKE2,
CPace

Bonus lemma: either $\mathcal{F}_{\mathbf{blePAKE}} \Rightarrow \mathcal{F}_{\mathbf{blePAKE}}$

Properties of ParComb



Theorem 1: Let PAKE_1 and PAKE_2 be 1-round statistically password-hiding, then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

EKE

No 1-round statistically password-hiding PQ scheme realizing $\mathcal{F}_{\mathrm{PAKE}}$ exists

Theorem 2:

1. **PAKE_1** is $\mathcal{F}_{\mathbf{lePAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathbf{lePAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathbf{lePAKE}} \Rightarrow \mathrm{ParComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathbf{lePAKE}}$

SPAKE2, CPace

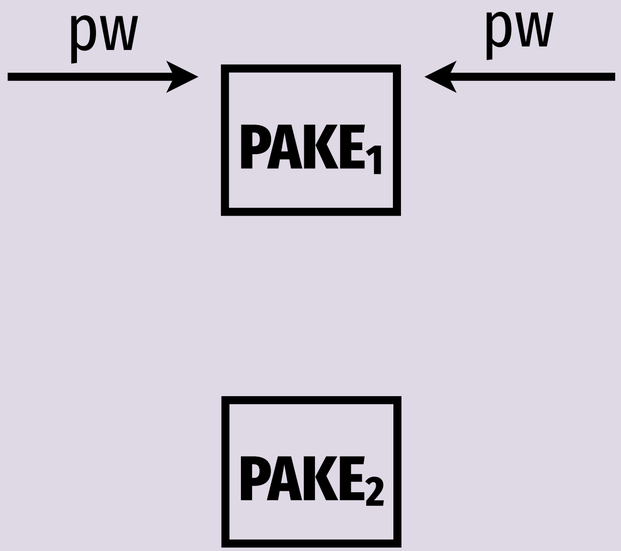
Bonus lemma: either $\mathcal{F}_{\mathbf{blePAKE}} \Rightarrow \mathcal{F}_{\mathbf{blePAKE}}$

Theorem 3: X-GA-PAKE (CSIDH-based) is a statistically pw-hiding 1-round blePAKE

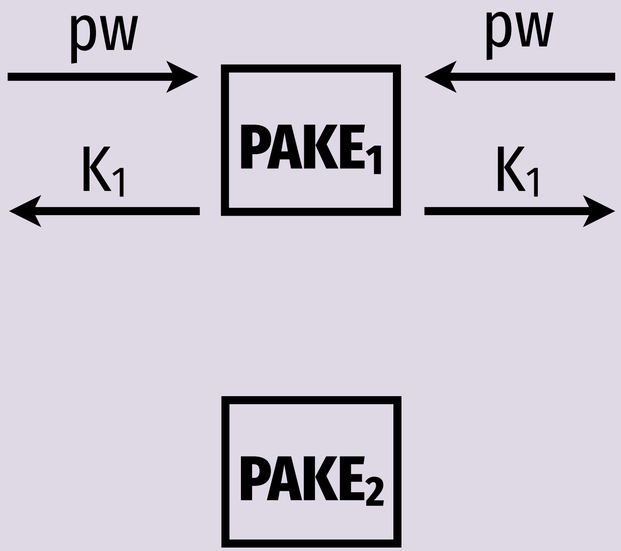
Building SeqComb

Building SeqComb

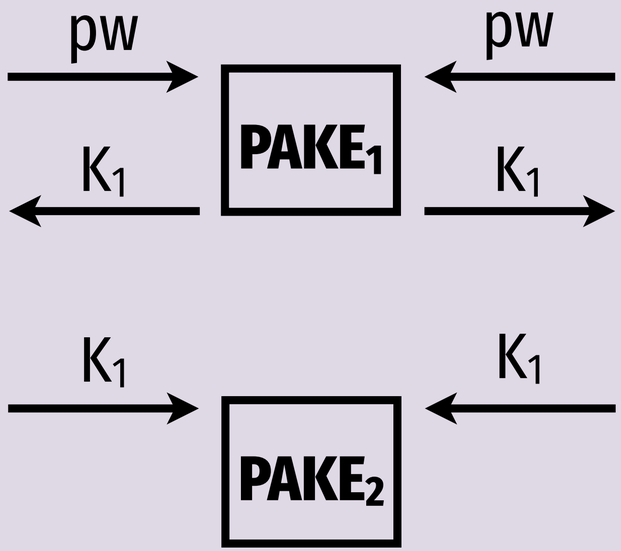
Building SeqComb



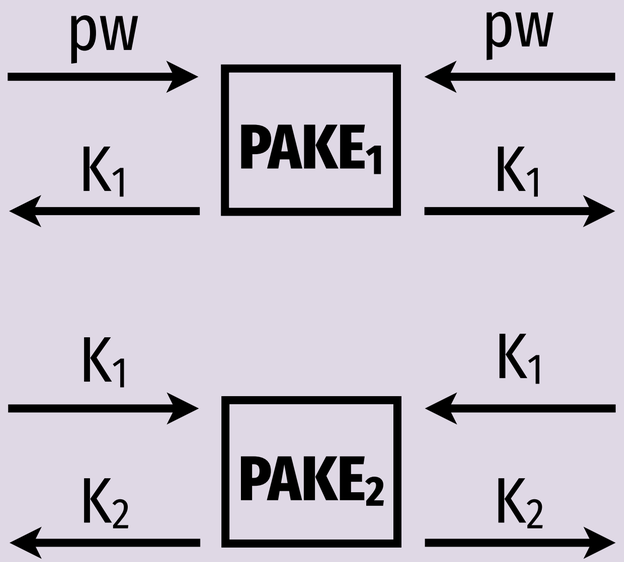
Building SeqComb



Building SeqComb



Building SeqComb



Building SeqComb

$K := H(K_1, K_2, tr)$

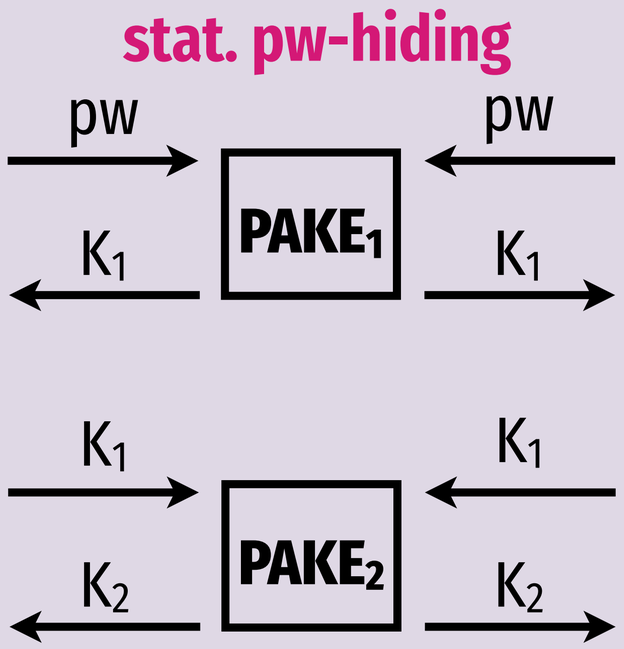
Building SeqComb

$PAKE_1$ can have the τ problem

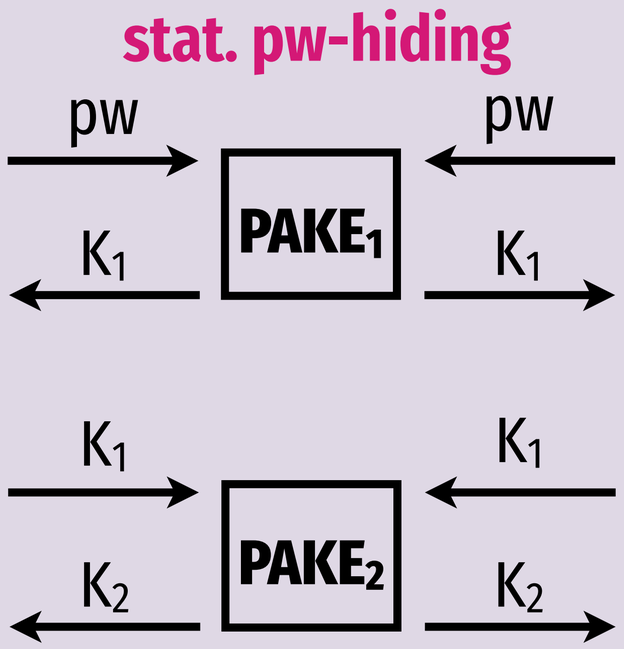
$K := H(K_1, K_2, tr)$

Building SeqComb

$PAKE_1$ can have the τ problem

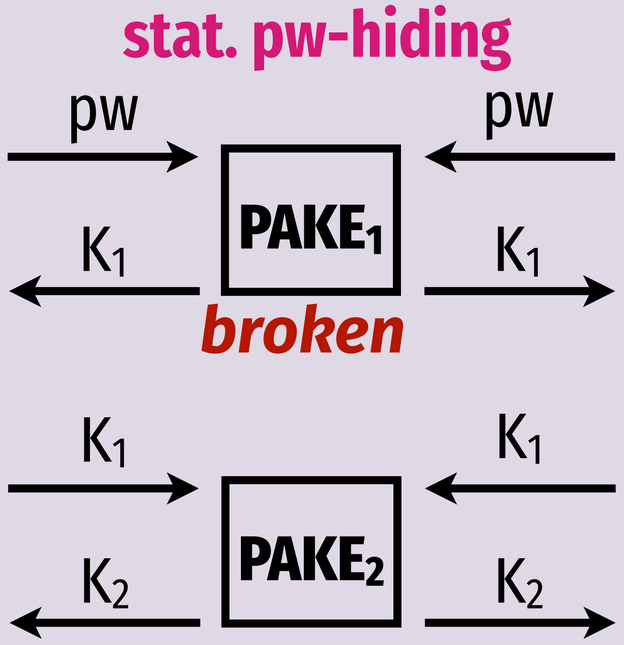


Building SeqComb



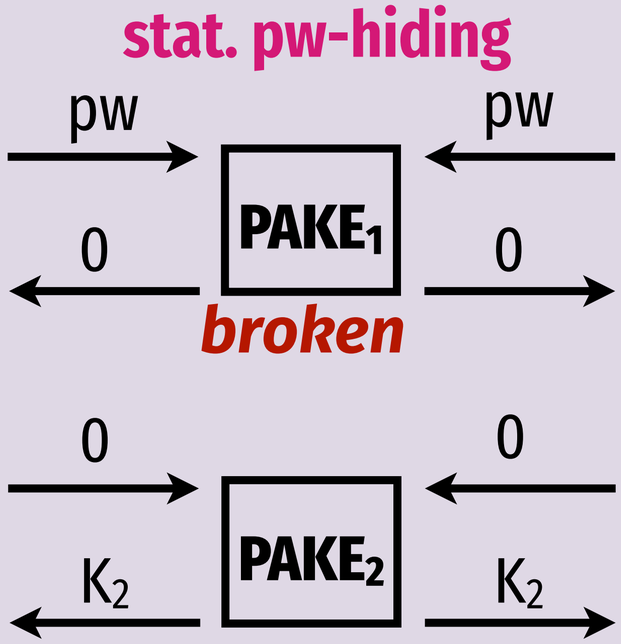
$K := H(K_1, K_2, tr)$

Building SeqComb



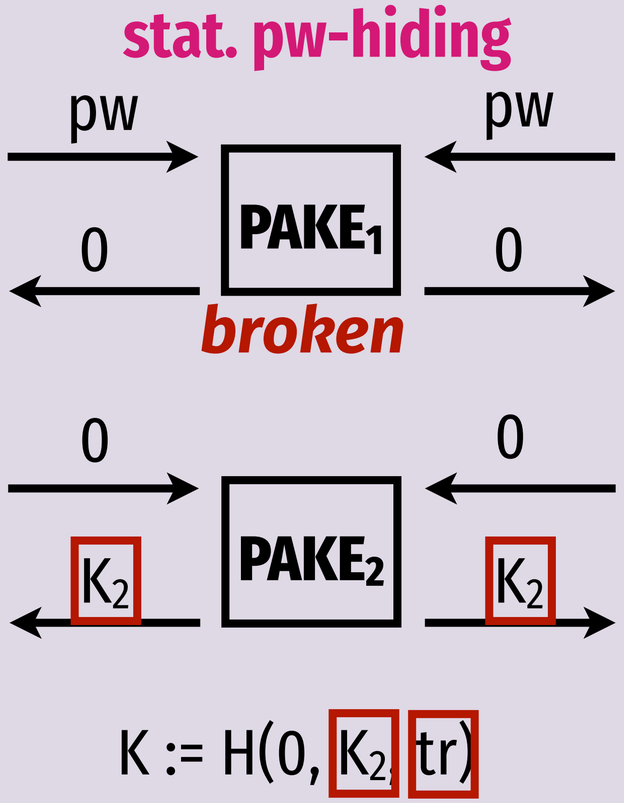
$K := H(K_1, K_2, tr)$

Building SeqComb

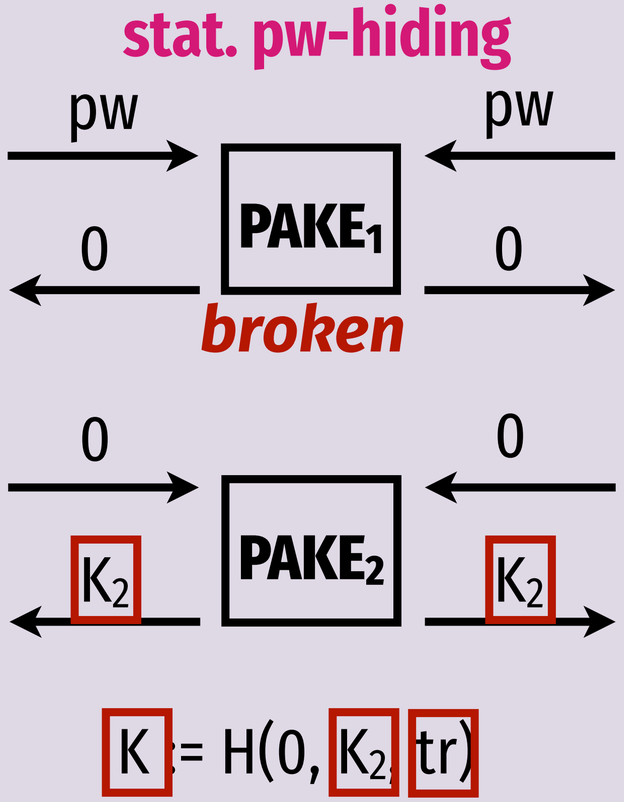


$K := H(0, K_2, tr)$

Building SeqComb

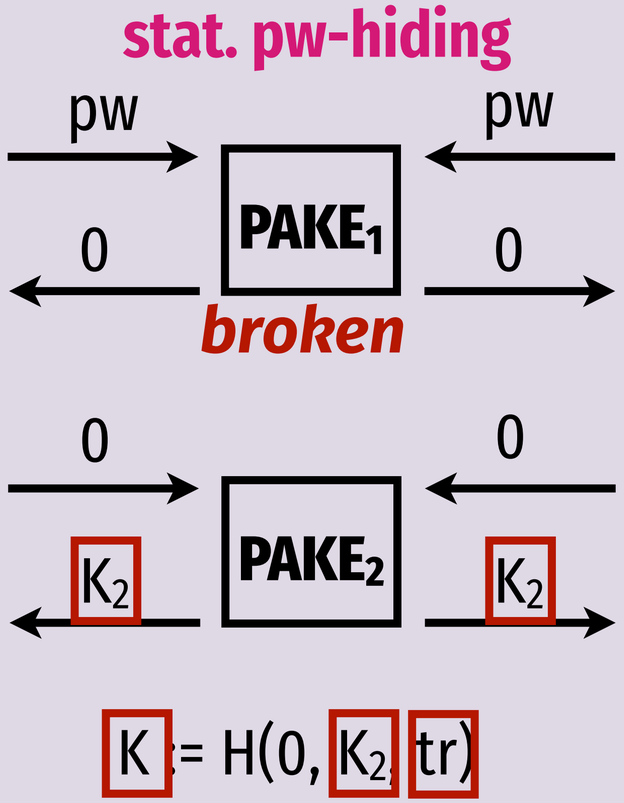


Building SeqComb



Building SeqComb

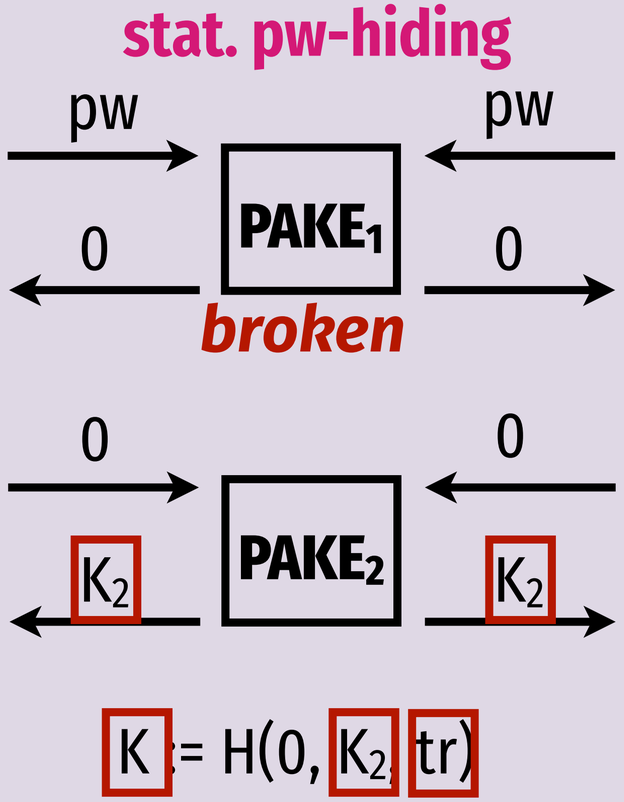
Session key is predictable!



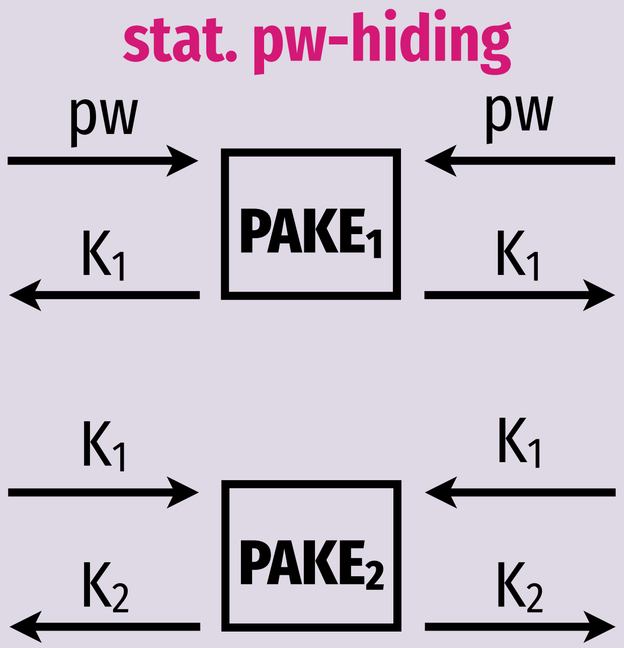
Building SeqComb

Session key is predictable!

Input more into PAKE_2

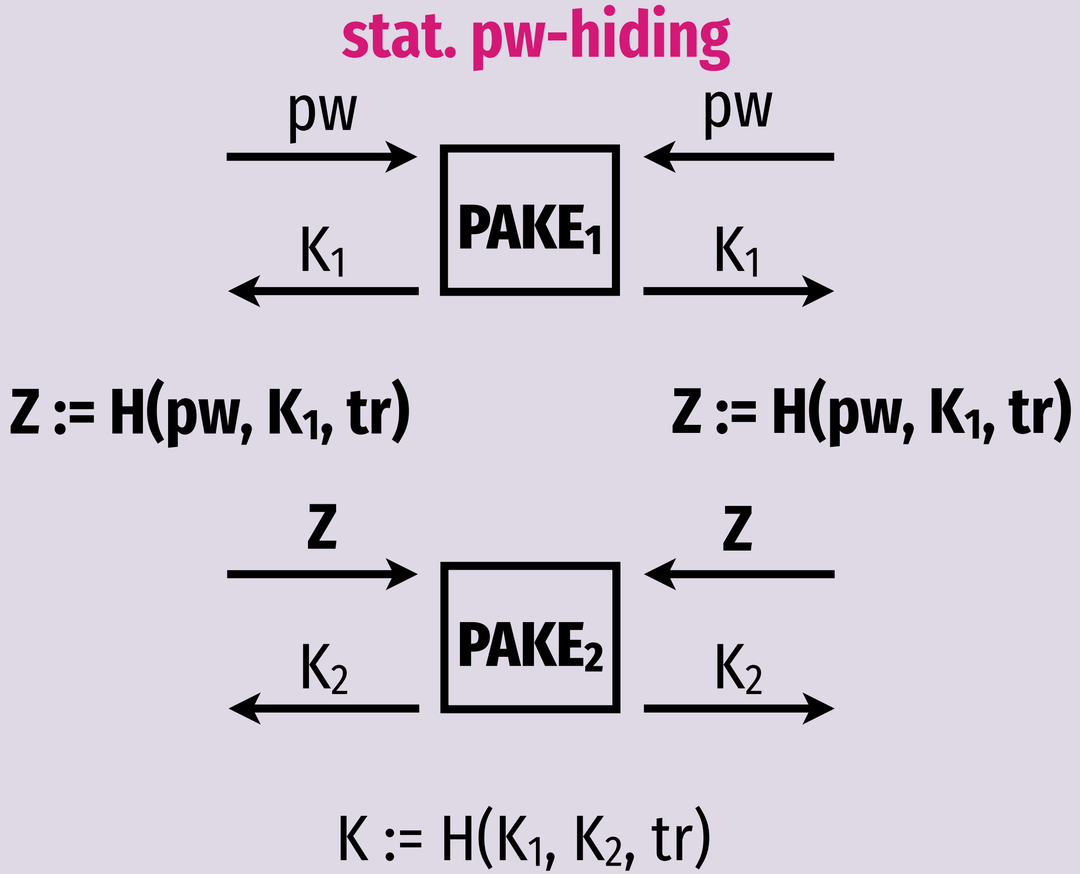


Building SeqComb

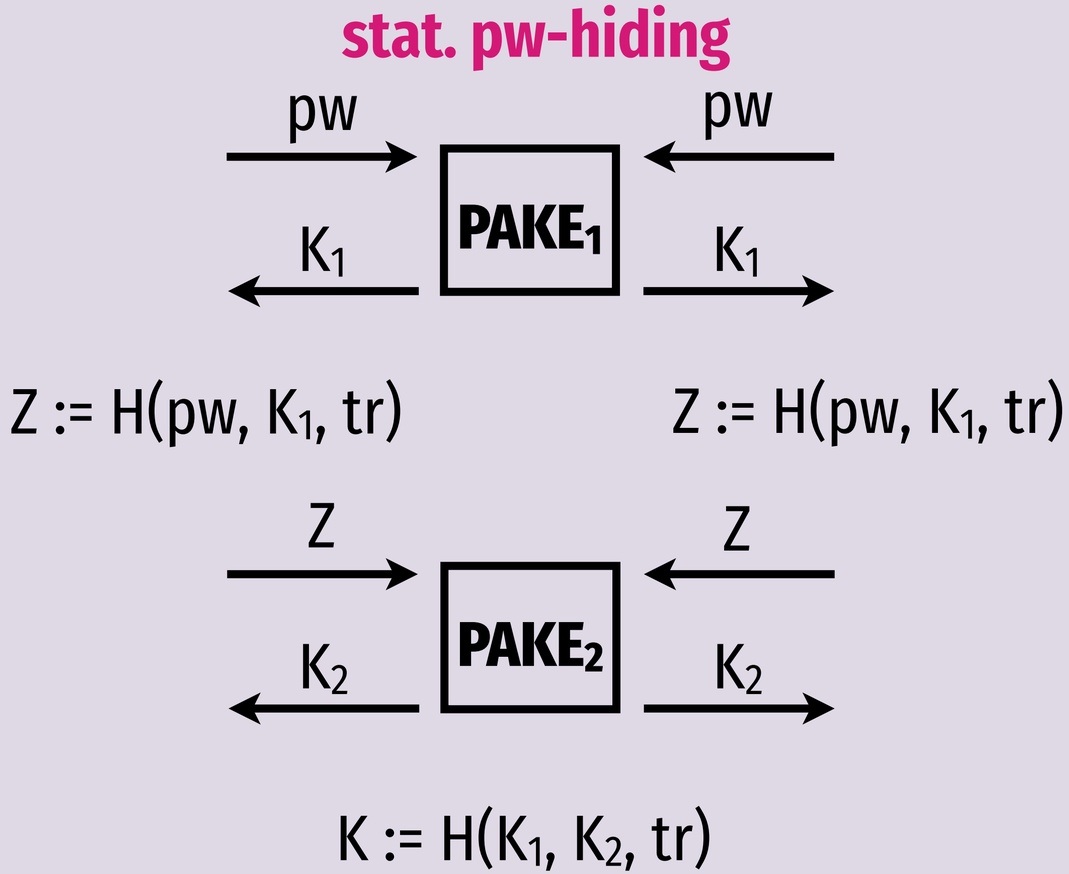


$K := H(K_1, K_2, tr)$

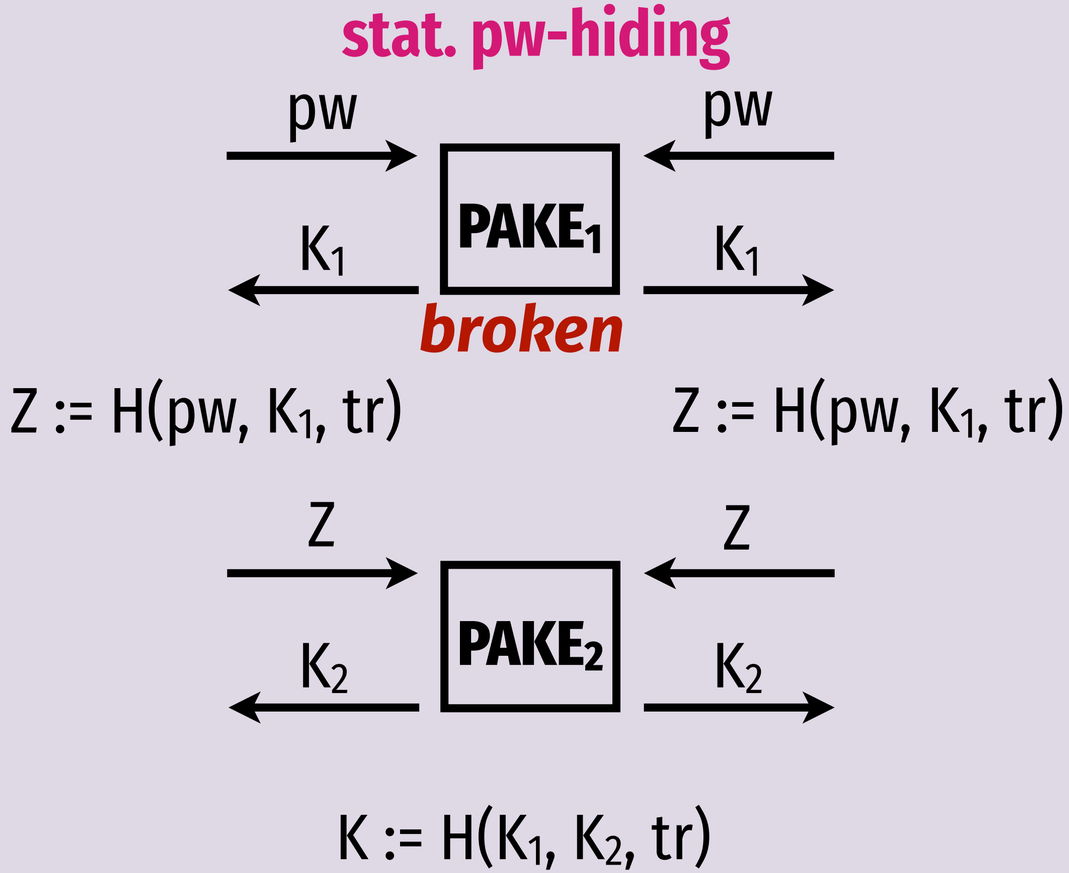
Building SeqComb



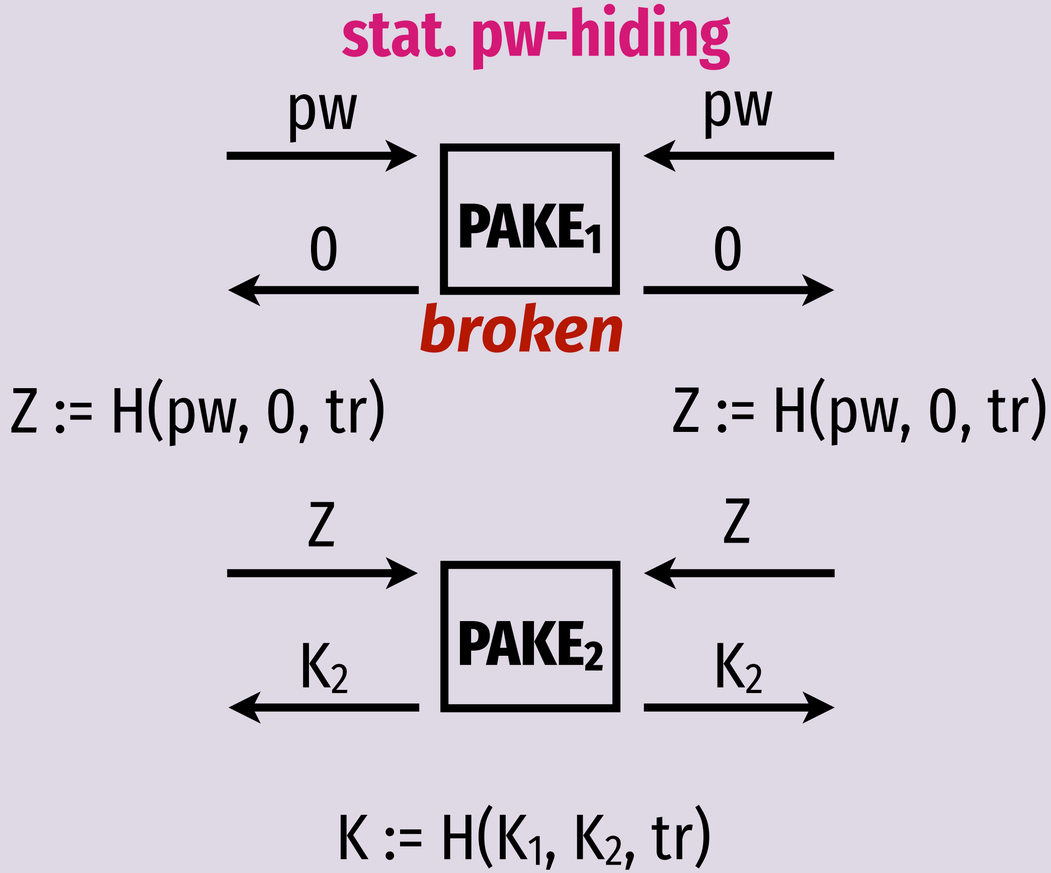
Building SeqComb



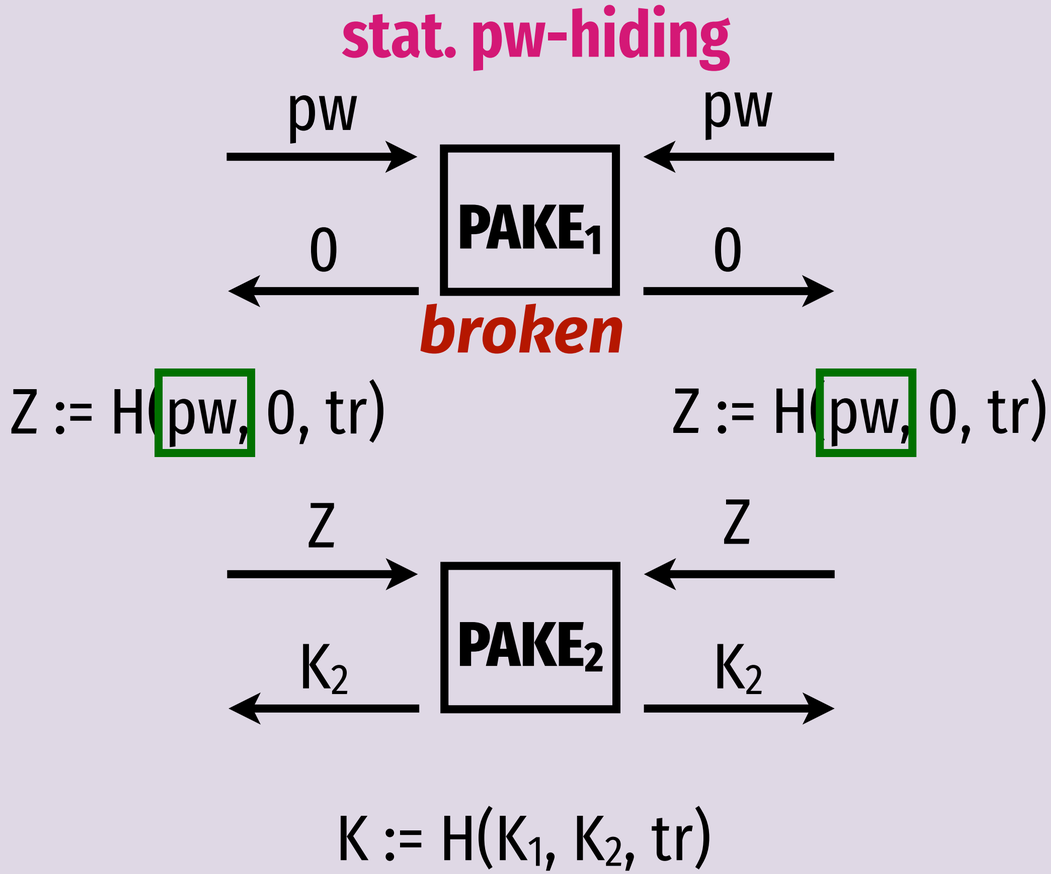
Building SeqComb



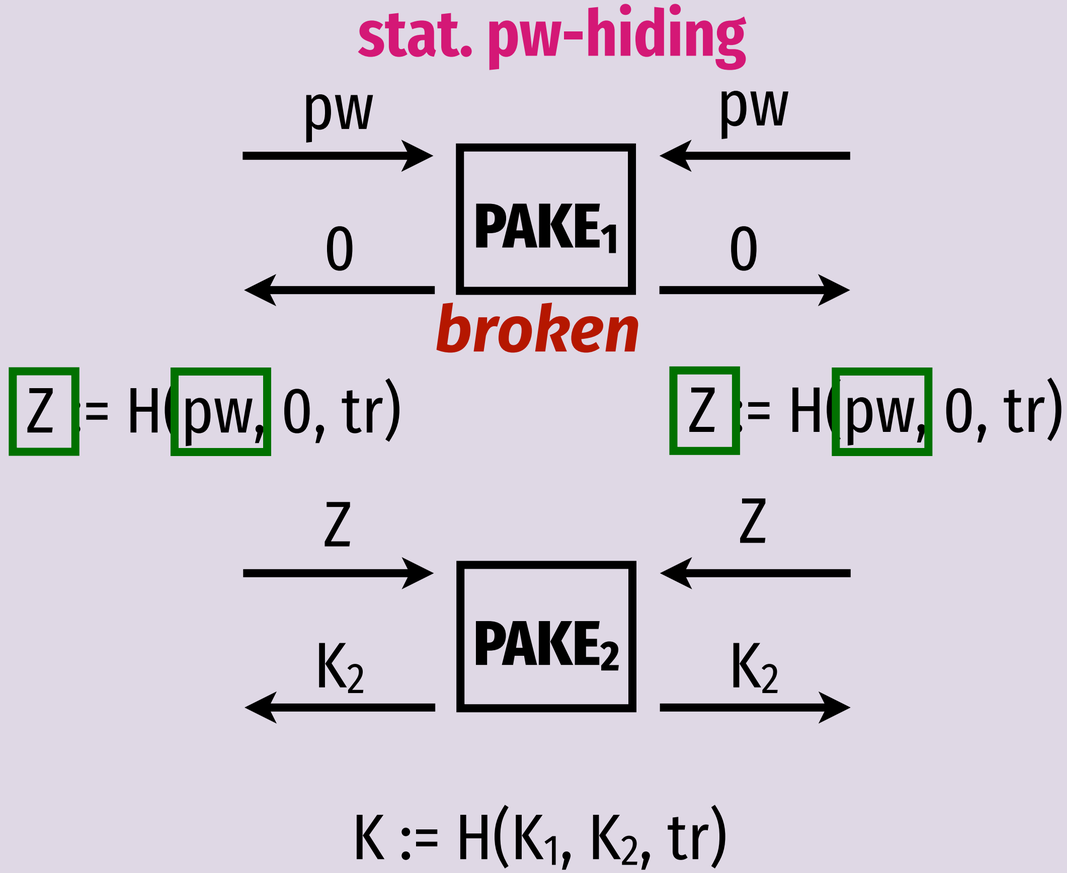
Building SeqComb



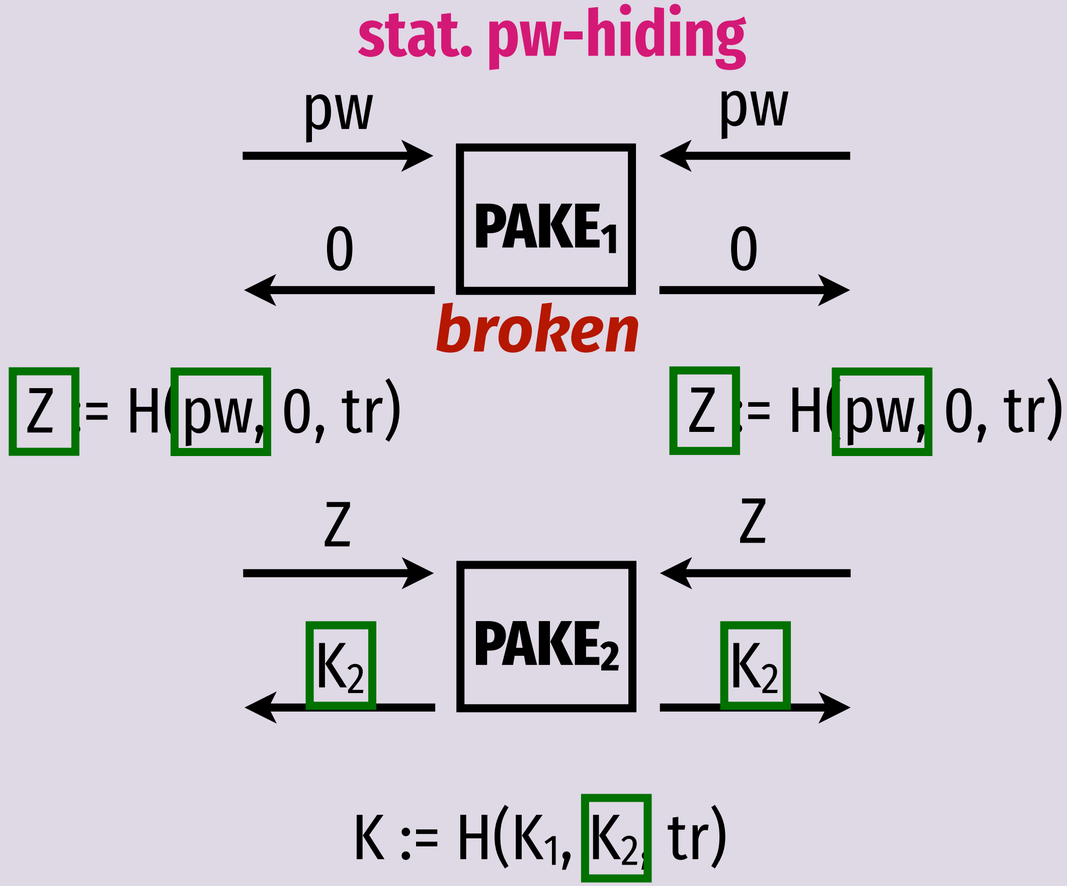
Building SeqComb



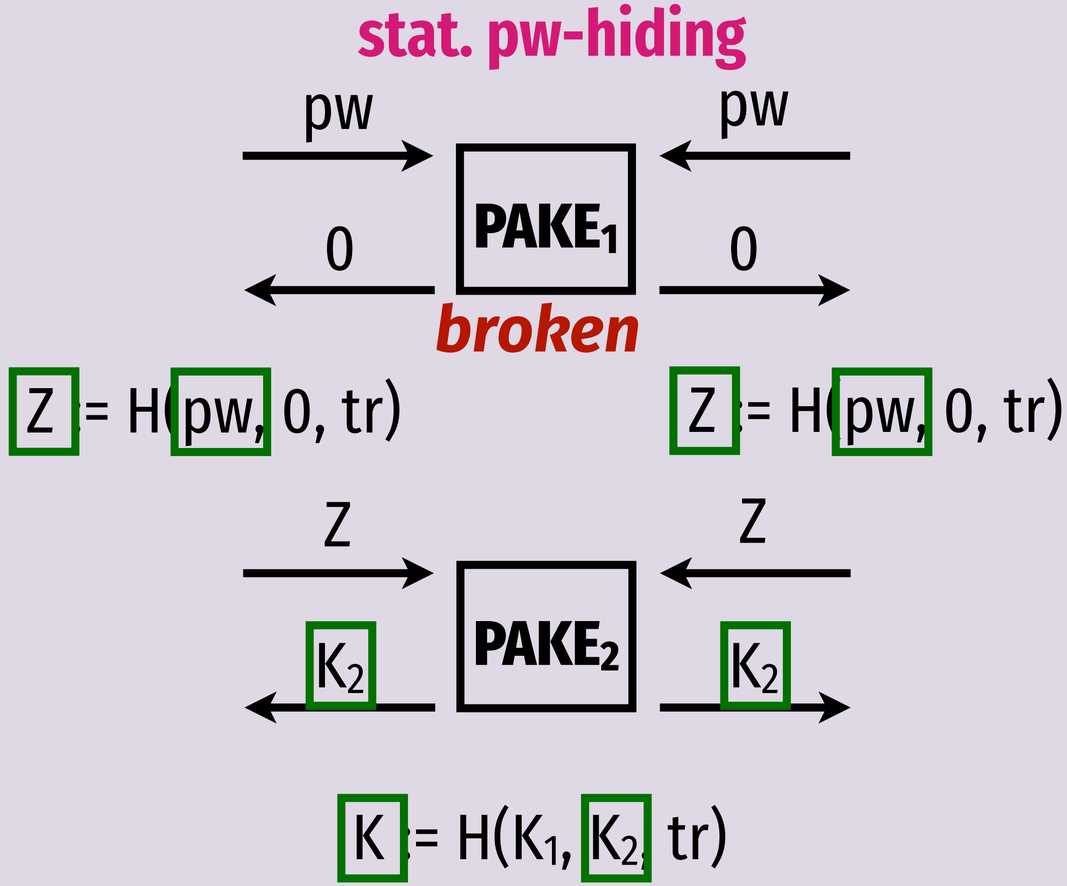
Building SeqComb



Building SeqComb

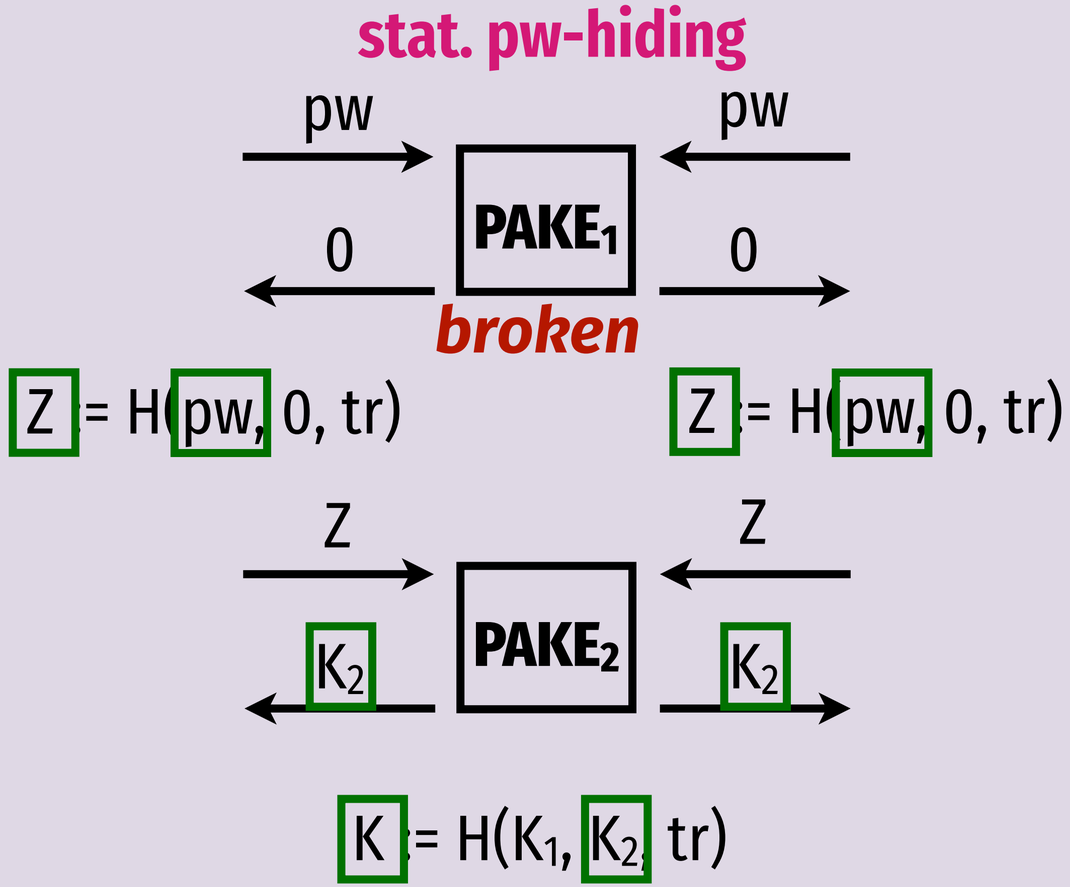


Building SeqComb

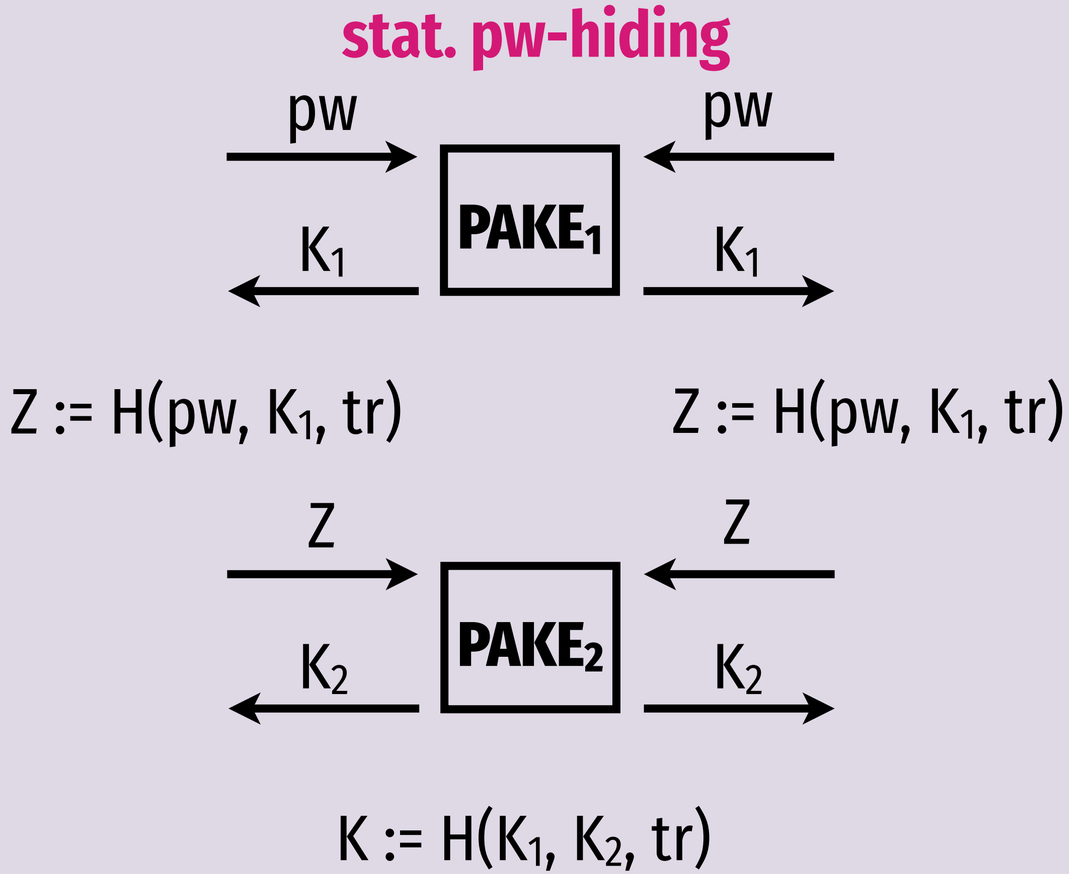


Building SeqComb

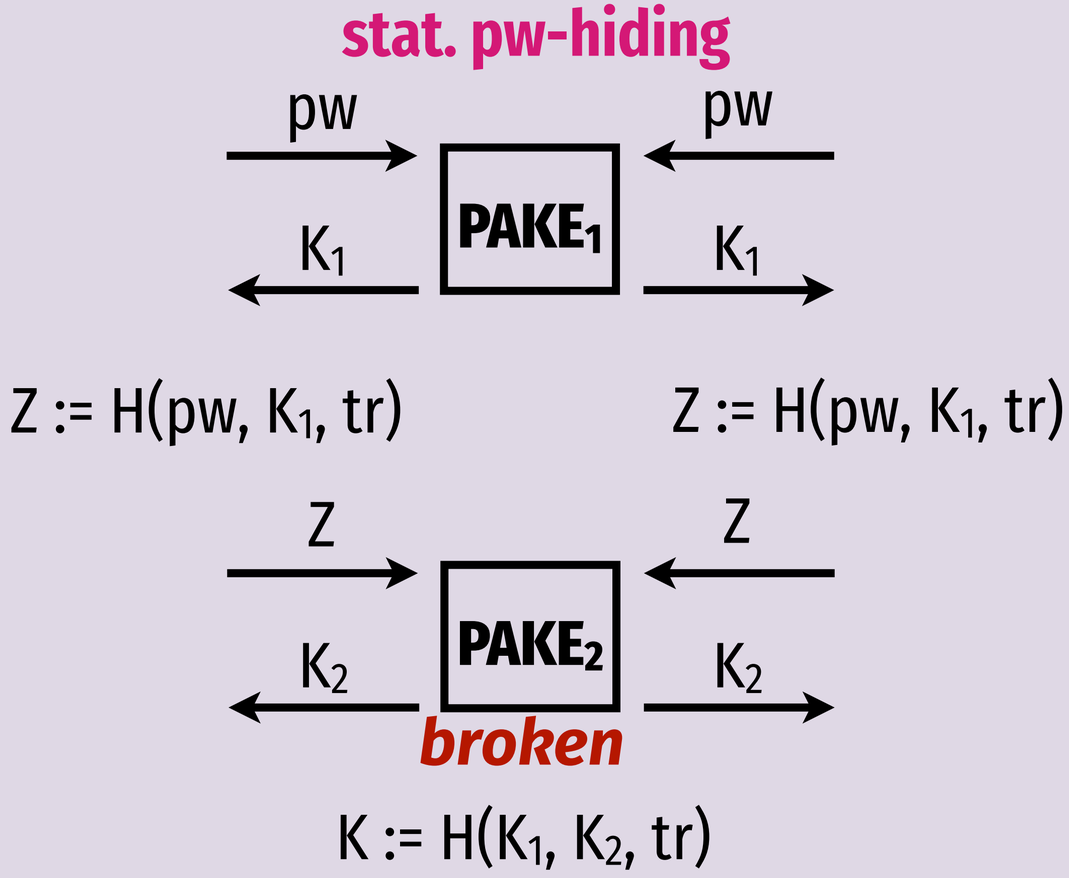
Combined PAKE is still secure!



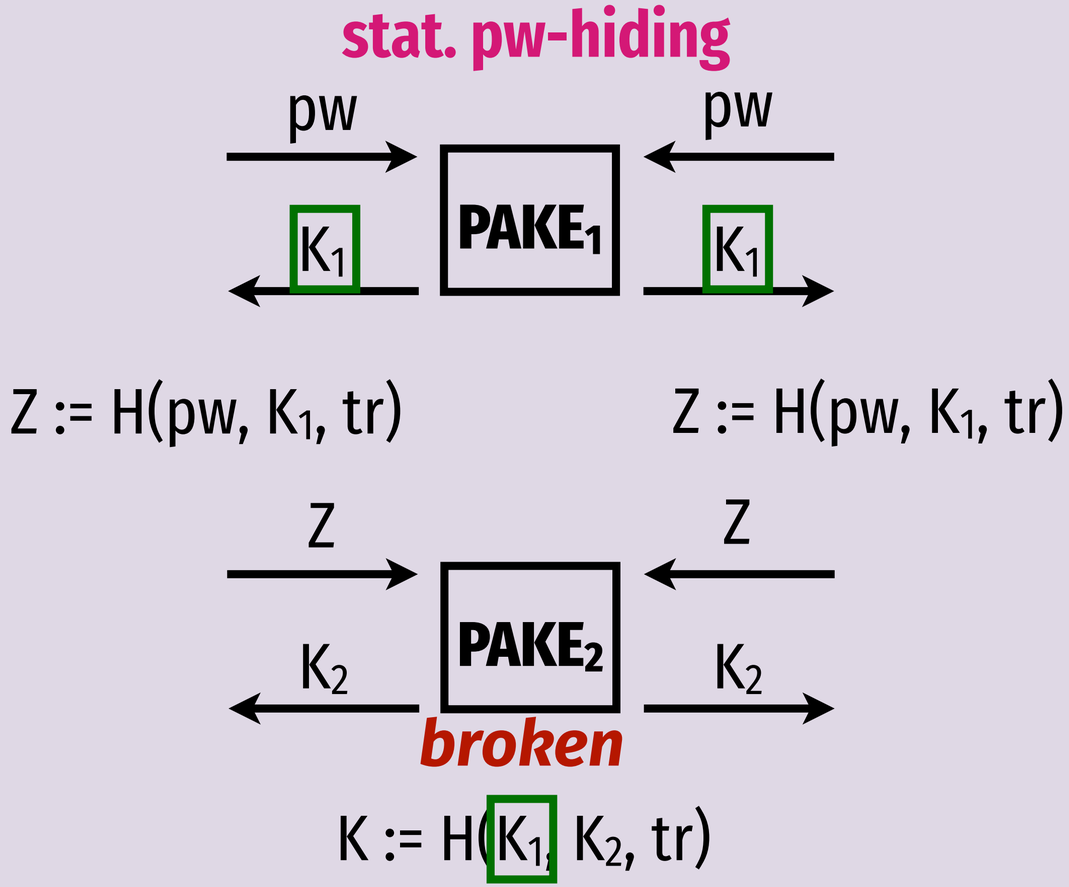
Building SeqComb



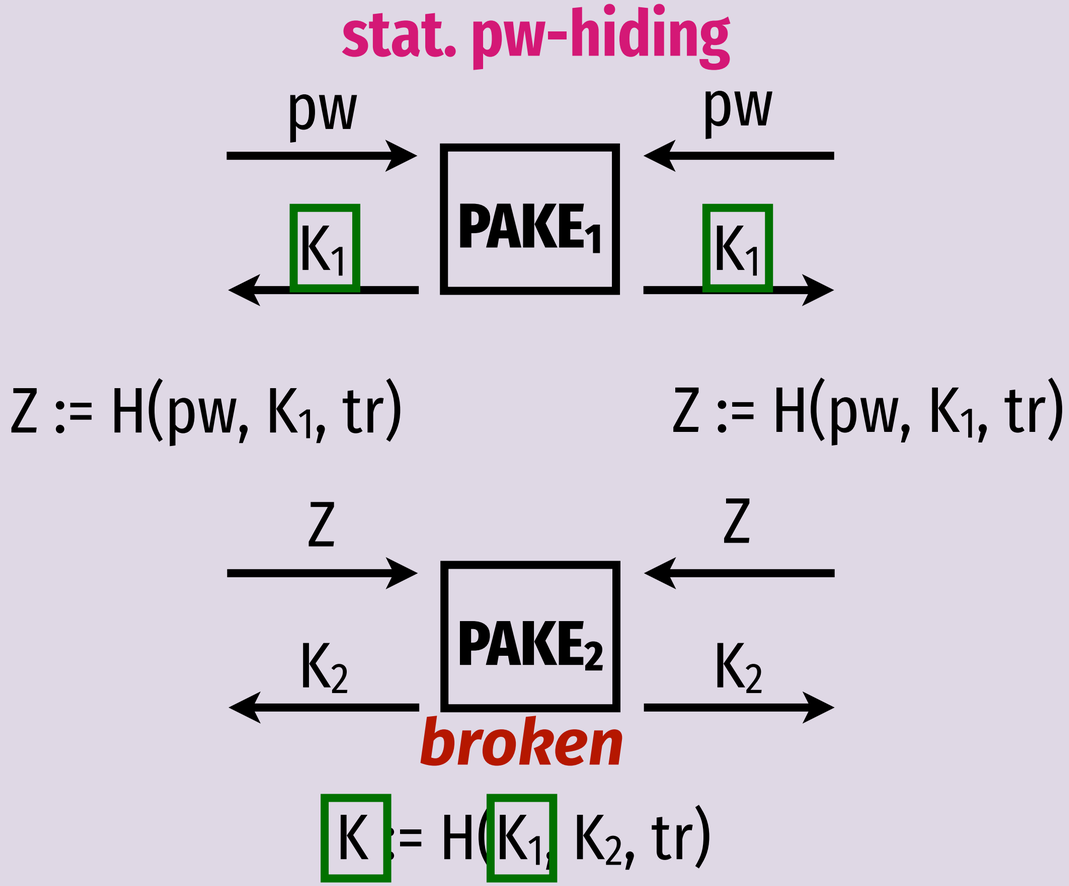
Building SeqComb



Building SeqComb

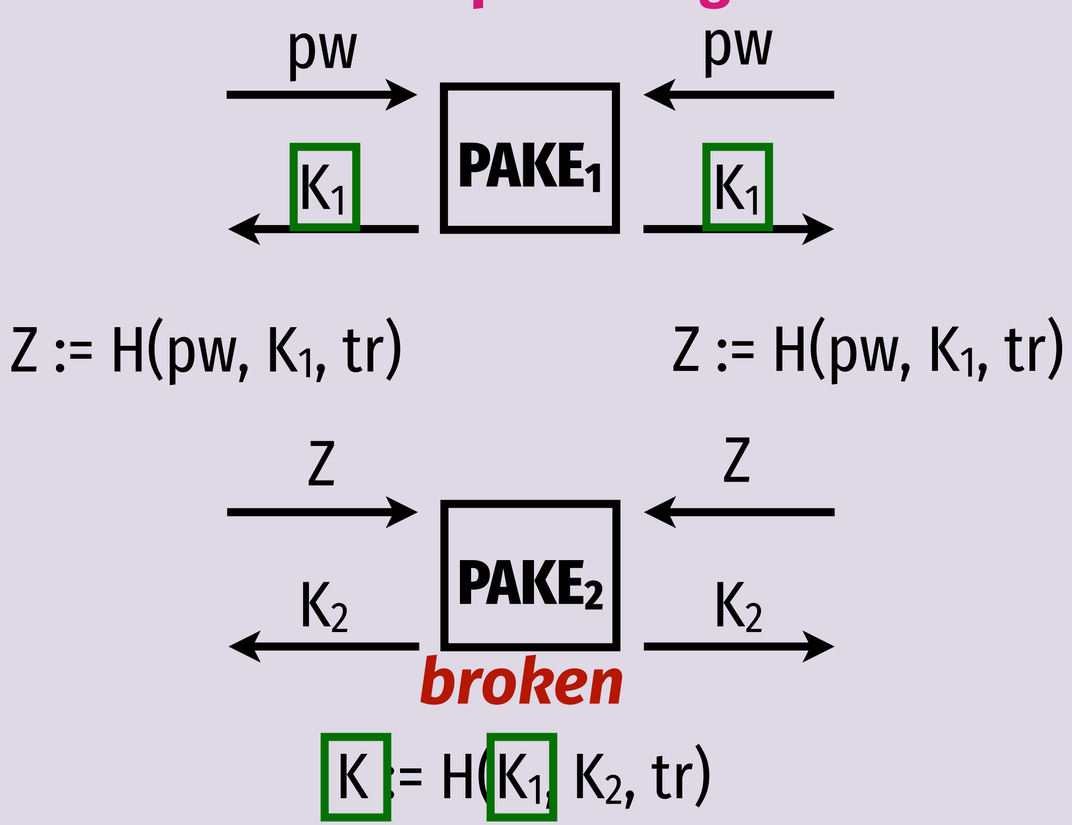


Building SeqComb



Building SeqComb

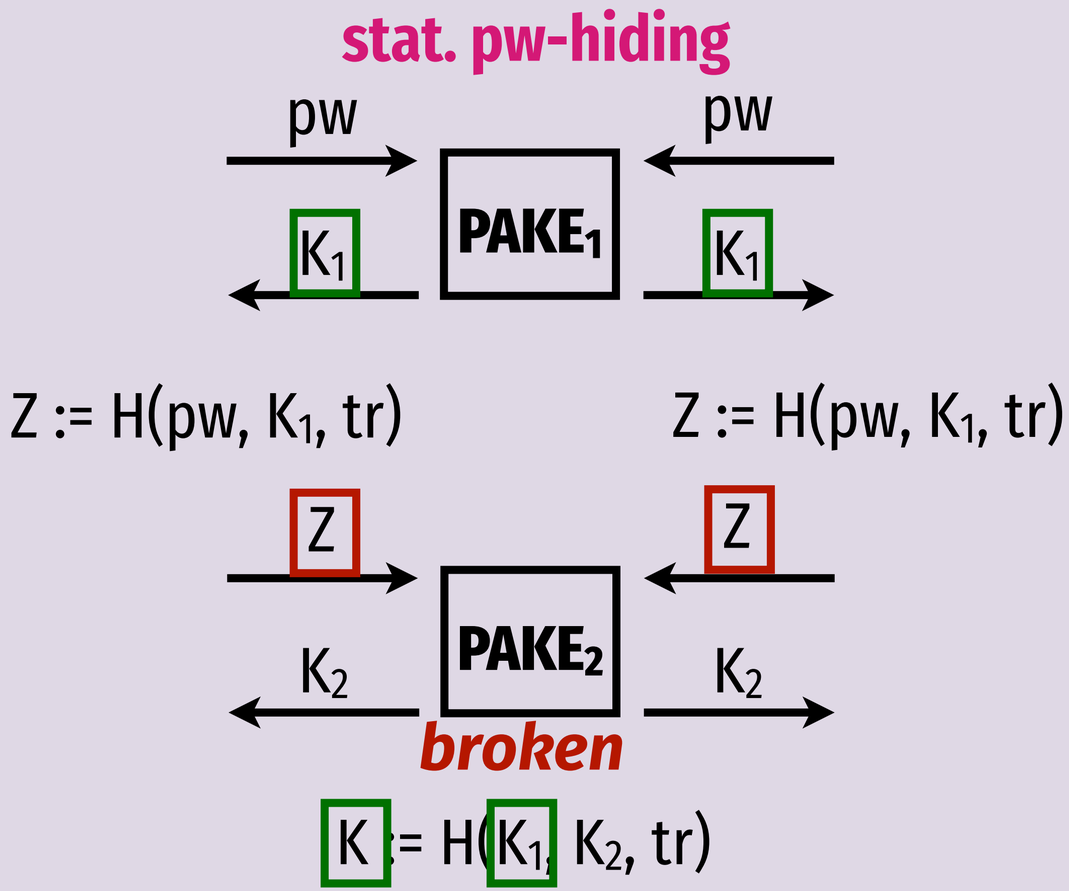
Session key is secure



Building SeqComb

Session key is secure

If Z leaks, the success of $PAKE_1$ leaks

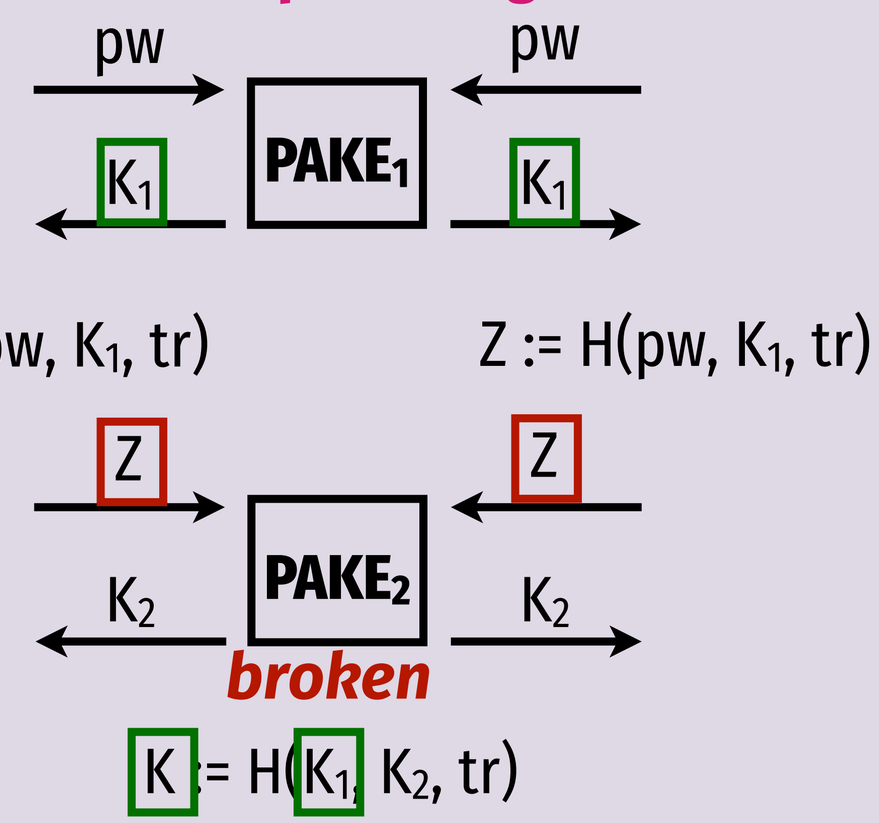


Building SeqComb

Session key is secure

If Z leaks, the success of $PAKE_1$ leaks

New $PAKE_2$ property: **statistical preshared key equality hiding.**



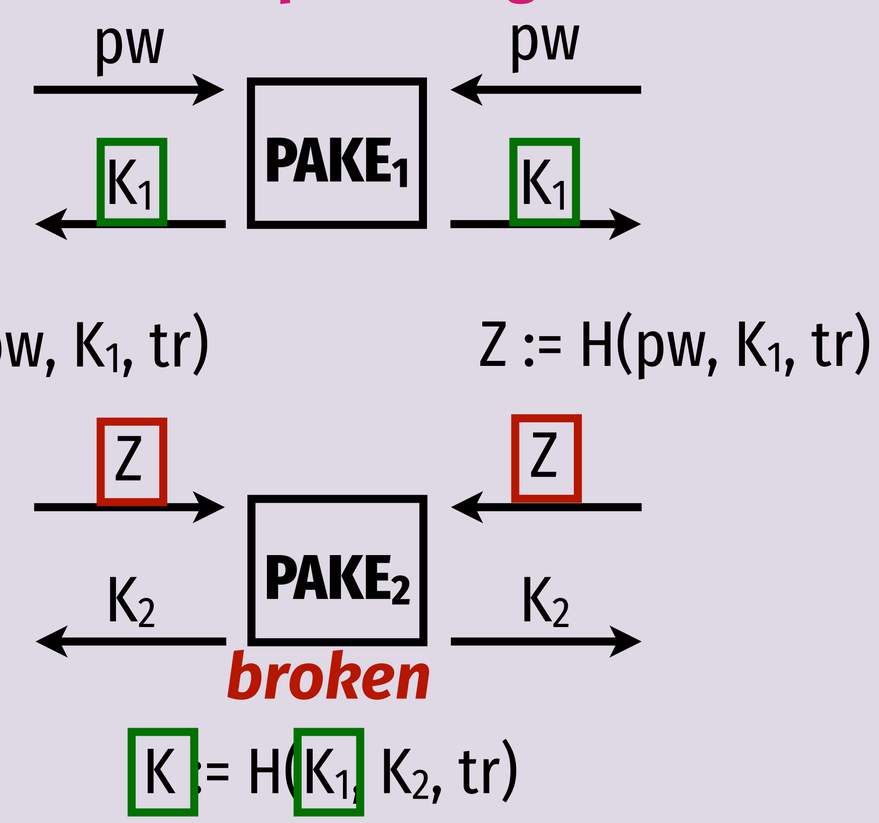
Building SeqComb

Session key is secure

If Z leaks, the success of $PAKE_1$ leaks

New $PAKE_2$ property: **statistical preshared key equality hiding.**

If inputs are high entropy, their equality is statistically hidden



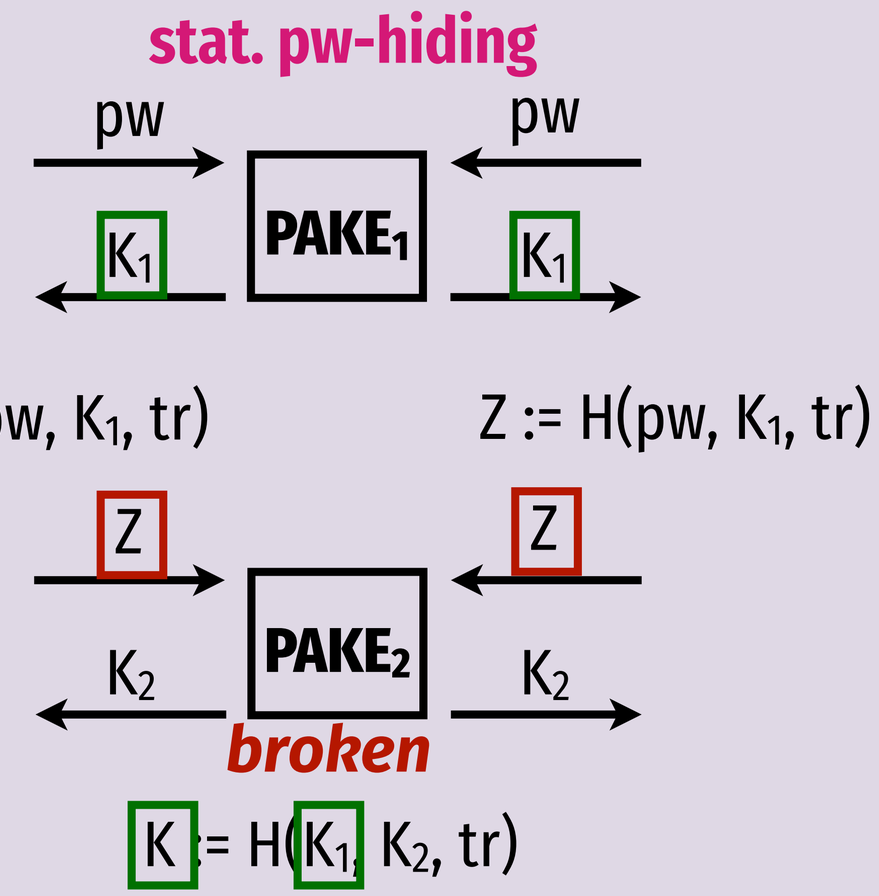
Building SeqComb

Session key is secure

If Z leaks, the success of $PAKE_1$ leaks

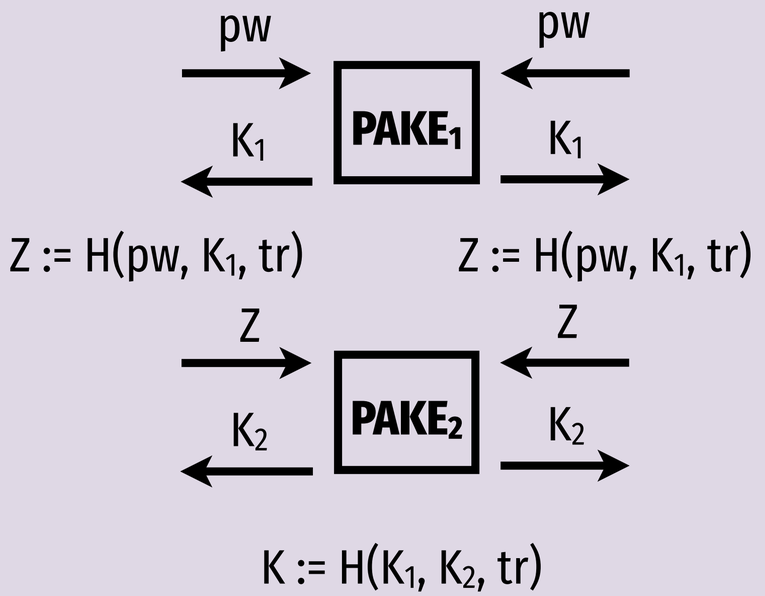
New $PAKE_2$ property: **statistical preshared key equality hiding.**

If inputs are high entropy, their equality is statistically hidden



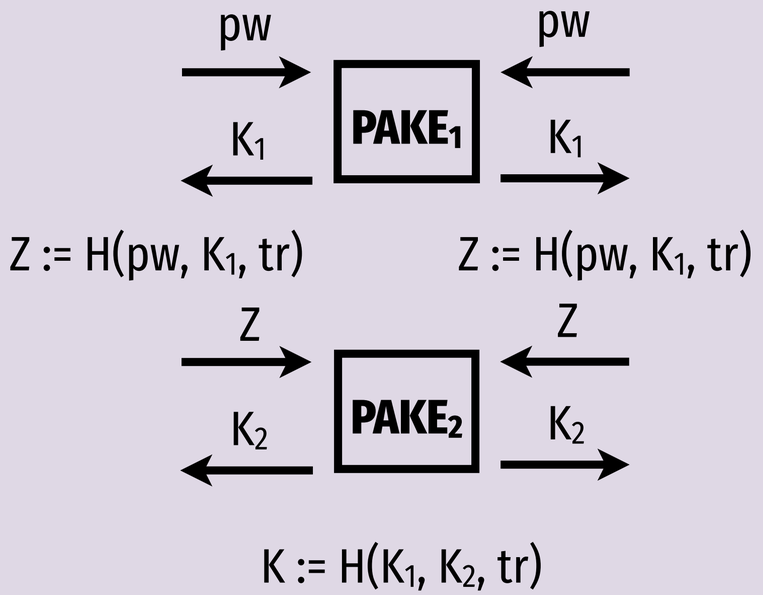
EKE-PRF and CAKE are stat. PSK-equality-hiding

Properties of SeqComb



Properties of SeqComb

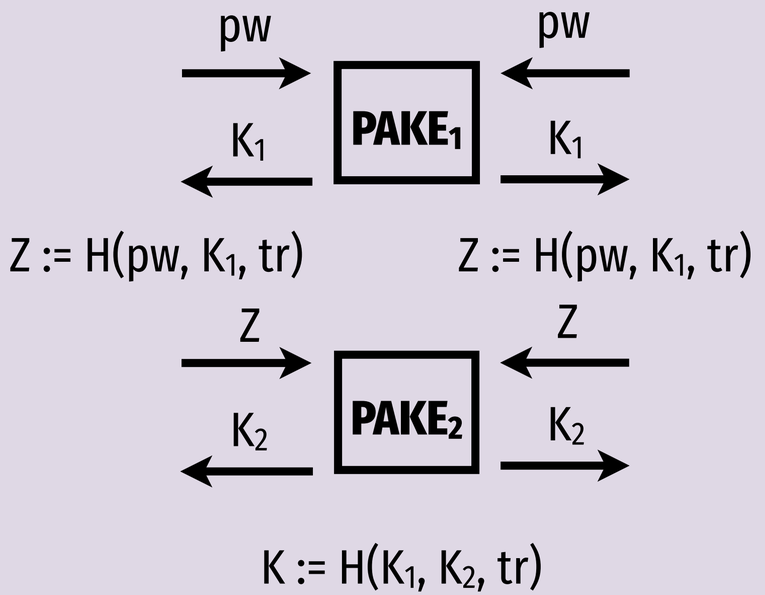
Theorem 4: Let PAKE_1 be 1-round stat. pw-hiding, and PAKE_2 be stat. PSK equality-hiding. Then



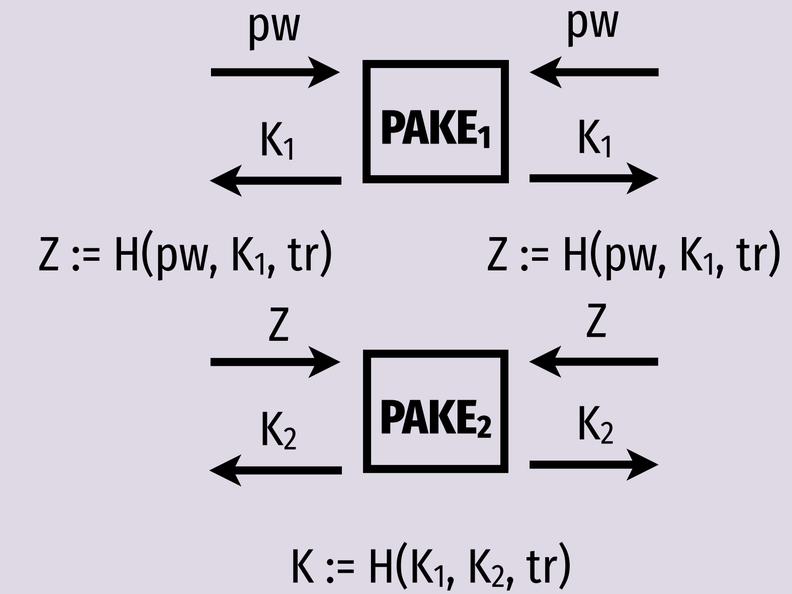
Properties of SeqComb

Theorem 4: Let PAKE_1 be 1-round stat. pw-hiding, and PAKE_2 be stat. PSK equality-hiding. Then

1. **PAKE_1** is $\mathcal{F}_{\mathsf{PAKE}} \Rightarrow \mathsf{SeqComb}[\mathsf{PAKE}_1, \mathsf{PAKE}_2]$ is $\mathcal{F}_{\mathsf{PAKE}}$



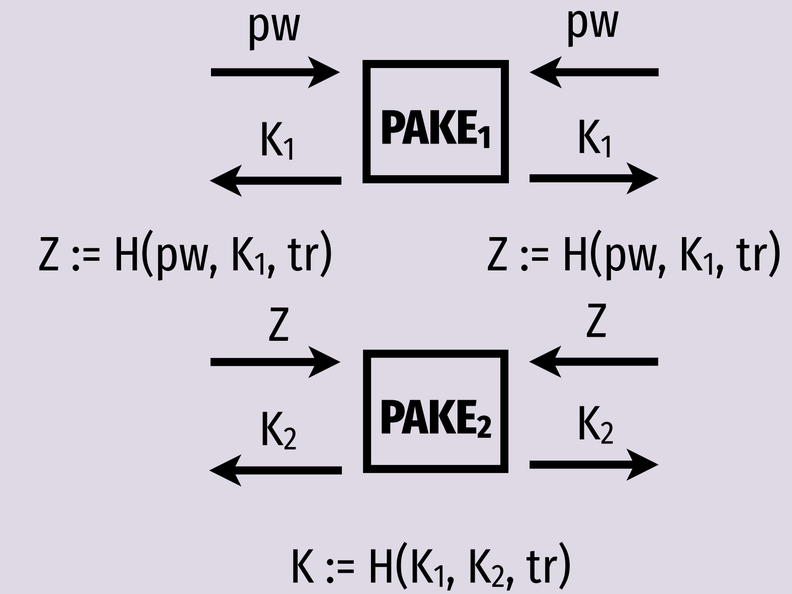
Properties of SeqComb



Theorem 4: Let PAKE_1 be 1-round stat. pw-hiding, and PAKE_2 be stat. PSK equality-hiding. Then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{SeqComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{SeqComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

Properties of SeqComb

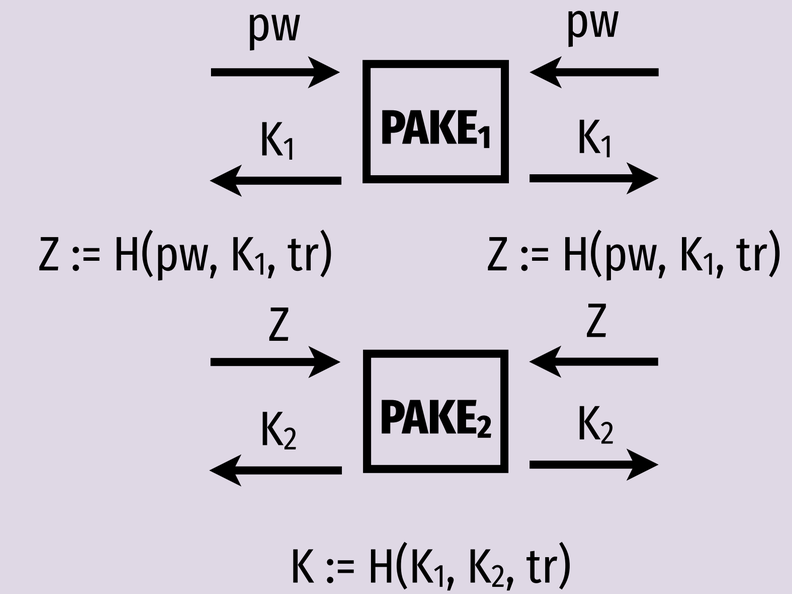


Theorem 4: Let PAKE_1 be 1-round stat. pw-hiding, and PAKE_2 be stat. PSK equality-hiding. Then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{SeqComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{SeqComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

Bonus lemma: PAKE_1 $\mathcal{F}_{\mathrm{lePAKE}} \Rightarrow \mathcal{F}_{\mathrm{rlePAKE}}$

Properties of SeqComb



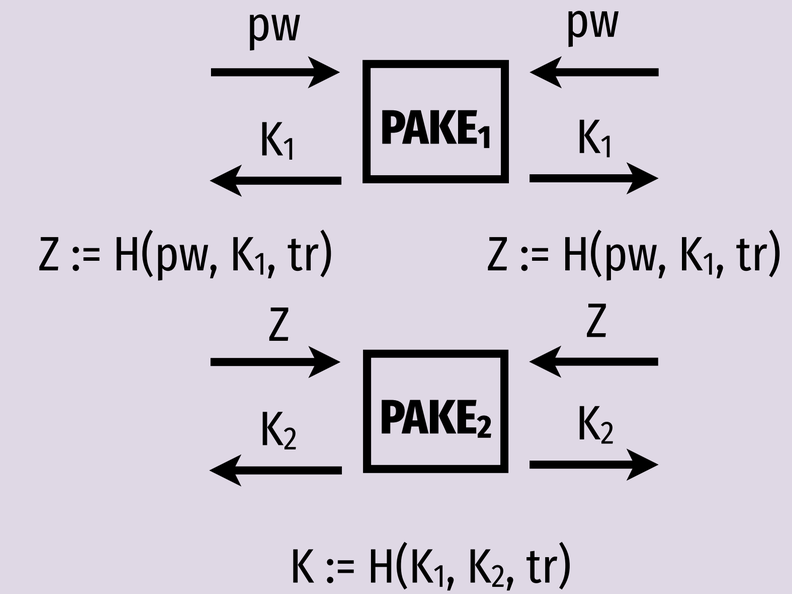
Theorem 4: Let PAKE_1 be 1-round stat. pw-hiding, and PAKE_2 be stat. PSK equality-hiding. Then

1. **PAKE_1** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{SeqComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathrm{PAKE}} \Rightarrow \mathrm{SeqComb}[\mathrm{PAKE}_1, \mathrm{PAKE}_2]$ is $\mathcal{F}_{\mathrm{PAKE}}$

Bonus lemma: PAKE_1 $\mathcal{F}_{\mathrm{lePAKE}} \Rightarrow \mathcal{F}_{\mathrm{rlePAKE}}$

Theorem 5: Let PAKE_1 be 1-round stat. pw-hiding, and **PAKE_2 be any PAKE**. Then

Properties of SeqComb



Theorem 4: Let PAKE_1 be 1-round stat. pw-hiding, and PAKE_2 be stat. PSK equality-hiding. Then

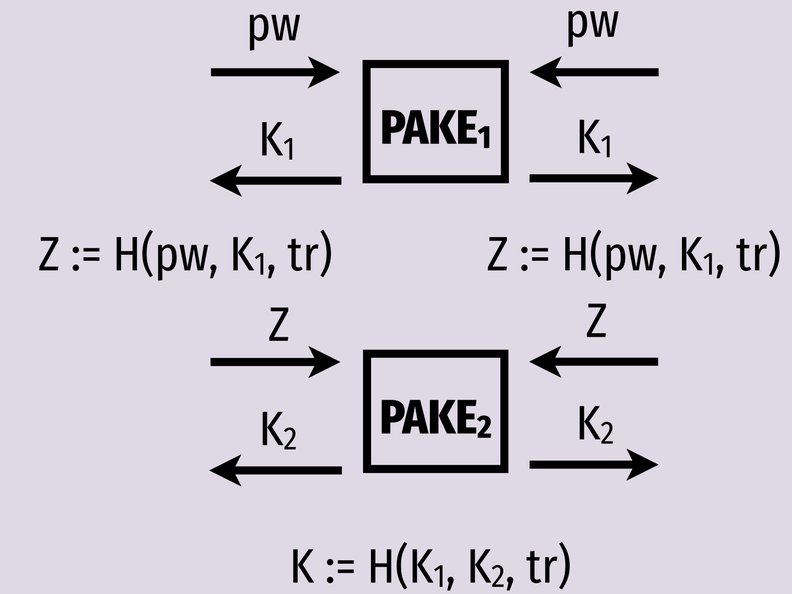
1. **PAKE_1** is $\mathcal{F}_{\text{PAKE}} \Rightarrow \text{SeqComb}[\text{PAKE}_1, \text{PAKE}_2]$ is $\mathcal{F}_{\text{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\text{PAKE}} \Rightarrow \text{SeqComb}[\text{PAKE}_1, \text{PAKE}_2]$ is $\mathcal{F}_{\text{PAKE}}$

Bonus lemma: PAKE_1 $\mathcal{F}_{\text{lePAKE}} \Rightarrow \mathcal{F}_{\text{rlePAKE}}$

Theorem 5: Let PAKE_1 be 1-round stat. pw-hiding, and **PAKE_2 be any PAKE**. Then

1. **PAKE_1** is $\mathcal{F}_{\text{PAKE}} \Rightarrow \text{SeqComb}[\text{PAKE}_1, \text{PAKE}_2]$ **nearly UC-realizes** $\mathcal{F}_{\text{PAKE}}$

Properties of SeqComb



Theorem 4: Let PAKE_1 be 1-round stat. pw-hiding, and PAKE_2 be stat. PSK equality-hiding. Then

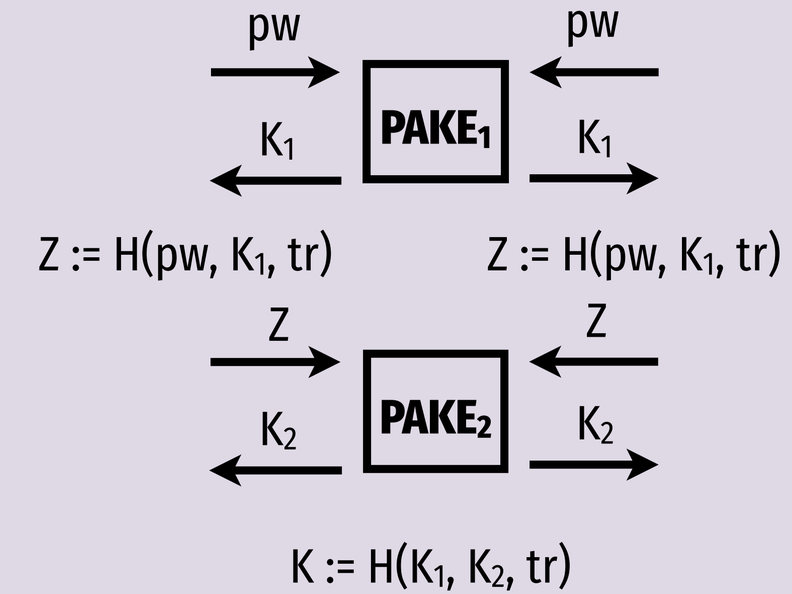
1. **PAKE_1** is $\mathcal{F}_{\text{PAKE}} \Rightarrow \text{SeqComb}[\text{PAKE}_1, \text{PAKE}_2]$ is $\mathcal{F}_{\text{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\text{PAKE}} \Rightarrow \text{SeqComb}[\text{PAKE}_1, \text{PAKE}_2]$ is $\mathcal{F}_{\text{PAKE}}$

Bonus lemma: PAKE_1 $\mathcal{F}_{\text{lePAKE}} \Rightarrow \mathcal{F}_{\text{rlePAKE}}$

Theorem 5: Let PAKE_1 be 1-round stat. pw-hiding, and **PAKE_2 be any PAKE**. Then

1. **PAKE_1** is $\mathcal{F}_{\text{PAKE}} \Rightarrow \text{SeqComb}[\text{PAKE}_1, \text{PAKE}_2]$ **nearly UC-realizes** $\mathcal{F}_{\text{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\text{PAKE}} \Rightarrow \text{SeqComb}[\text{PAKE}_1, \text{PAKE}_2]$ **nearly UC-realizes** $\mathcal{F}_{\text{PAKE}}$

Properties of SeqComb



Theorem 4: Let PAKE_1 be 1-round stat. pw-hiding, and PAKE_2 be stat. PSK equality-hiding. Then

1. **PAKE_1** is $\mathcal{F}_{\mathsf{PAKE}} \Rightarrow \mathsf{SeqComb}[\mathsf{PAKE}_1, \mathsf{PAKE}_2]$ is $\mathcal{F}_{\mathsf{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathsf{PAKE}} \Rightarrow \mathsf{SeqComb}[\mathsf{PAKE}_1, \mathsf{PAKE}_2]$ is $\mathcal{F}_{\mathsf{PAKE}}$

Bonus lemma: PAKE_1 $\mathcal{F}_{\mathsf{lePAKE}} \Rightarrow \mathcal{F}_{\mathsf{rlePAKE}}$

Theorem 5: Let PAKE_1 be 1-round stat. pw-hiding, and **PAKE_2 be any PAKE**. Then

1. **PAKE_1** is $\mathcal{F}_{\mathsf{PAKE}} \Rightarrow \mathsf{SeqComb}[\mathsf{PAKE}_1, \mathsf{PAKE}_2]$ **nearly UC-realizes** $\mathcal{F}_{\mathsf{PAKE}}$
2. **PAKE_2** is $\mathcal{F}_{\mathsf{PAKE}} \Rightarrow \mathsf{SeqComb}[\mathsf{PAKE}_1, \mathsf{PAKE}_2]$ **nearly UC-realizes** $\mathcal{F}_{\mathsf{PAKE}}$ **in QROM**

Conclusion

Conclusion

First construction of a **hybrid PAKE**

Conclusion

First construction of a **hybrid PAKE**

Presented **two methods**, parallel and sequential

Conclusion

First construction of a **hybrid PAKE**

Presented **two methods**, parallel and sequential

Sequential has **mild assumptions**, working with efficient existing PAKEs

Conclusion

First construction of a **hybrid PAKE**

Presented **two methods**, parallel and sequential

Sequential has **mild assumptions**, working with efficient existing PAKEs

Combiner	$PAKE_1$	$PAKE_2$	Combined	Model
ParComb	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE}}$	ROM
	$\mathcal{F}_{\text{blePAKE+PH}}$	$\mathcal{F}_{\text{lePAKE+PH}}$	$\mathcal{F}_{\text{blePAKE}}$	ROM
SeqComb	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE+PEH}}$	$\mathcal{F}_{\text{PAKE}}$	ROM
	$\mathcal{F}_{\text{lePAKE+PH}}$	$\mathcal{F}_{\text{PAKE+PEH}}$	$\mathcal{F}_{\text{rlePAKE}}$	ROM
	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE}}$	$(\mathcal{F}_{\text{PAKE}})$	QROM

+PH means stat. pw-hiding
+PEH means stat. PSK-equality-hiding
$(\mathcal{F})$ means nearly UC-realizes

Conclusion

First construction of a **hybrid PAKE**

Presented **two methods**, parallel and sequential

Sequential has **mild assumptions**, working with efficient existing PAKEs

Combiner	$PAKE_1$	$PAKE_2$	Combined	Model
ParComb	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE}}$	ROM
	$\mathcal{F}_{\text{blePAKE+PH}}$	$\mathcal{F}_{\text{lePAKE+PH}}$	$\mathcal{F}_{\text{blePAKE}}$	ROM
SeqComb	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE+PEH}}$	$\mathcal{F}_{\text{PAKE}}$	ROM
	$\mathcal{F}_{\text{lePAKE+PH}}$	$\mathcal{F}_{\text{PAKE+PEH}}$	$\mathcal{F}_{\text{rlePAKE}}$	ROM
	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE}}$	$(\mathcal{F}_{\text{PAKE}})$	QROM

+PH means stat. pw-hiding
+PEH means stat. PSK-equality-hiding
$(\mathcal{F})$ means nearly UC-realizes

https://www.ietf.org/archive/id/draft-vos-cfrg-pqpake-00.html

Conclusion

First construction of a **hybrid PAKE**

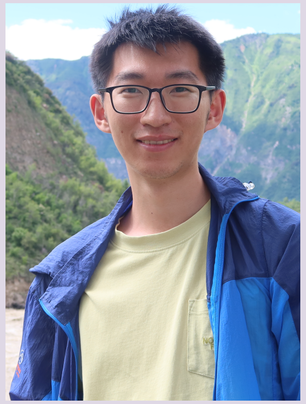
Presented **two methods**, parallel and sequential

Sequential has **mild assumptions**, working with efficient existing PAKEs

Combiner	$PAKE_1$	$PAKE_2$	Combined	Model
ParComb	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE}}$	ROM
	$\mathcal{F}_{\text{blePAKE+PH}}$	$\mathcal{F}_{\text{lePAKE+PH}}$	$\mathcal{F}_{\text{blePAKE}}$	ROM
SeqComb	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE+PEH}}$	$\mathcal{F}_{\text{PAKE}}$	ROM
	$\mathcal{F}_{\text{lePAKE+PH}}$	$\mathcal{F}_{\text{PAKE+PEH}}$	$\mathcal{F}_{\text{rlePAKE}}$	ROM
	$\mathcal{F}_{\text{PAKE+PH}}$	$\mathcal{F}_{\text{PAKE}}$	$(\mathcal{F}_{\text{PAKE}})$	QROM

+PH means stat. pw-hiding
+PEH means stat. PSK-equality-hiding
$(\mathcal{F})$ means nearly UC-realizes






Come find us!

https://www.ietf.org/archive/id/draft-vos-cfrg-pqpake-00.html

ia.cr/2024/1621
ia.cr/2024/1630