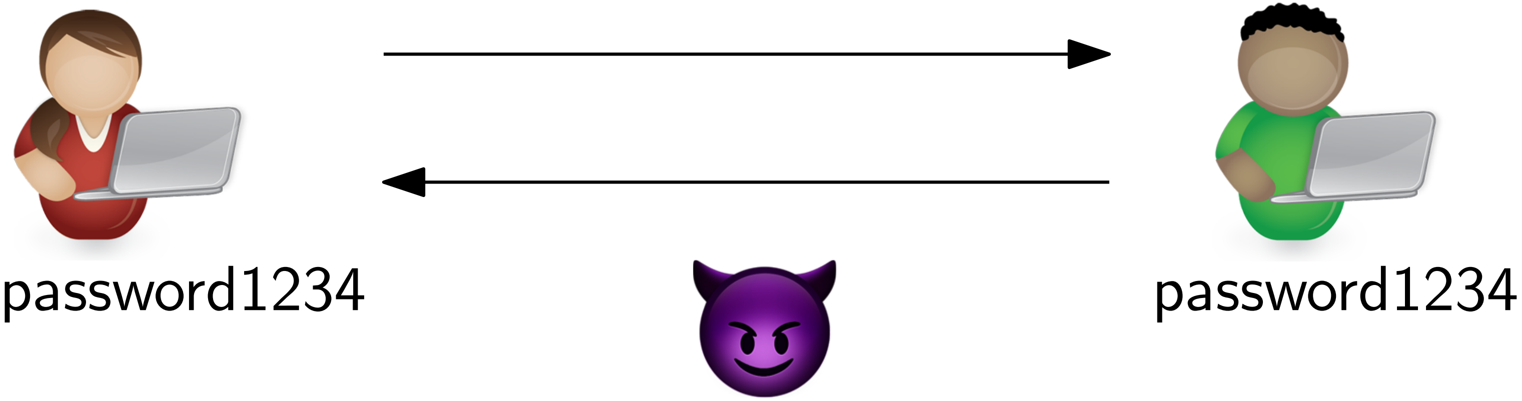
Generalized Fuzzy Password-Authenticated Key Exchange from Error Correcting Codes

Jonathan Bootle, Sebastian Faller, Julia Hesse, Kristina Hostáková, Johannes Ottenhues

Frebruary 14, 2025

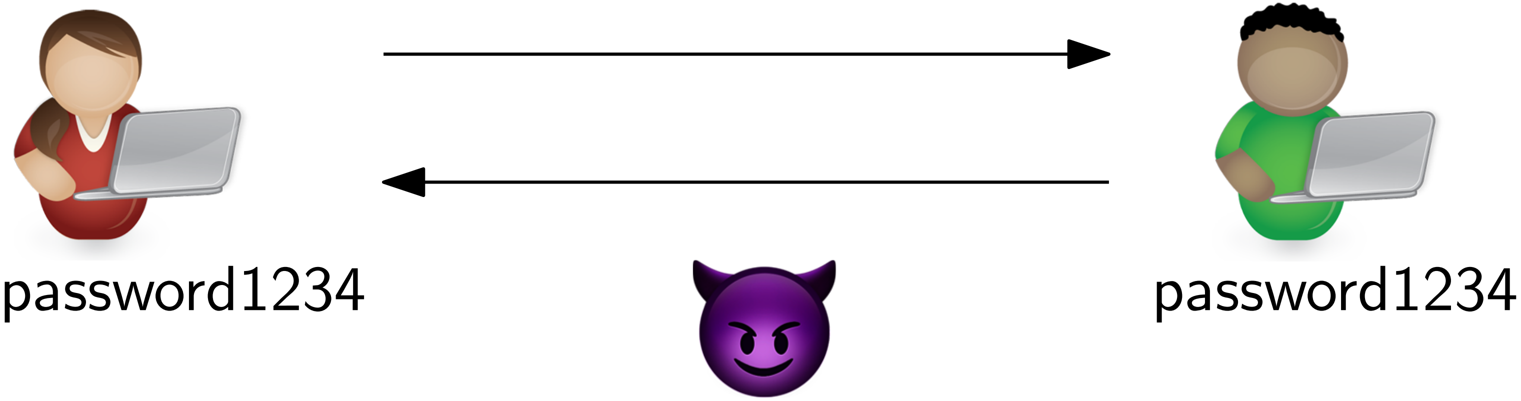
Password Authenticated Key Exchange (PAKE)



password1234

password1234

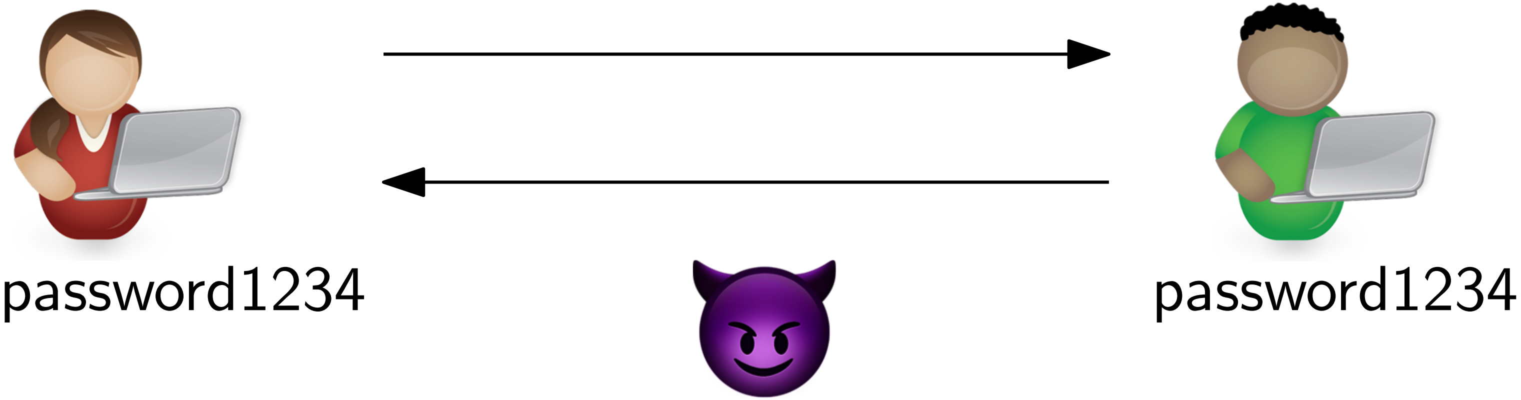
Password Authenticated Key Exchange (PAKE)



password1234

password1234

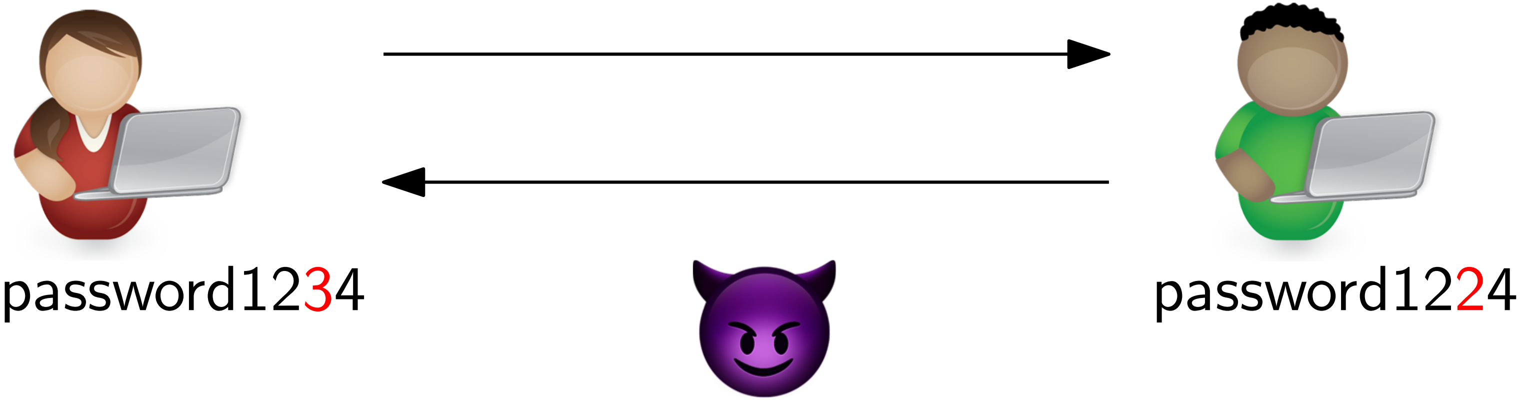
Password Authenticated Key Exchange (PAKE)



password1234

password1234

Fuzzy PAKE

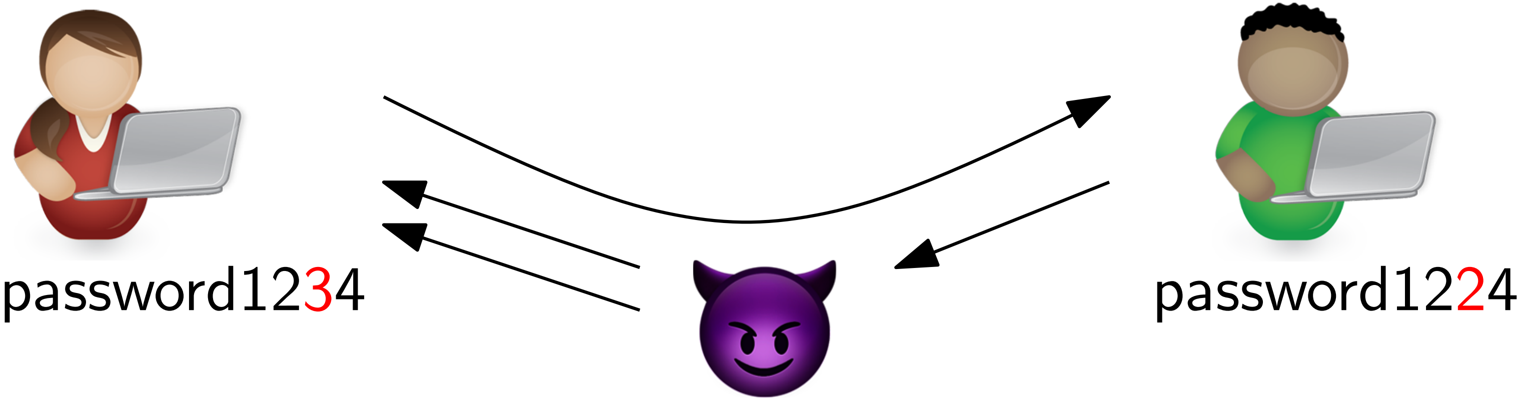


password1234

password1224

- Typo tolerance
- Biometric readings as passwords
- Pairing of nearby IoT devices with sensor data as passwords

Fuzzy PAKE - Attack Model



password1234

password1224

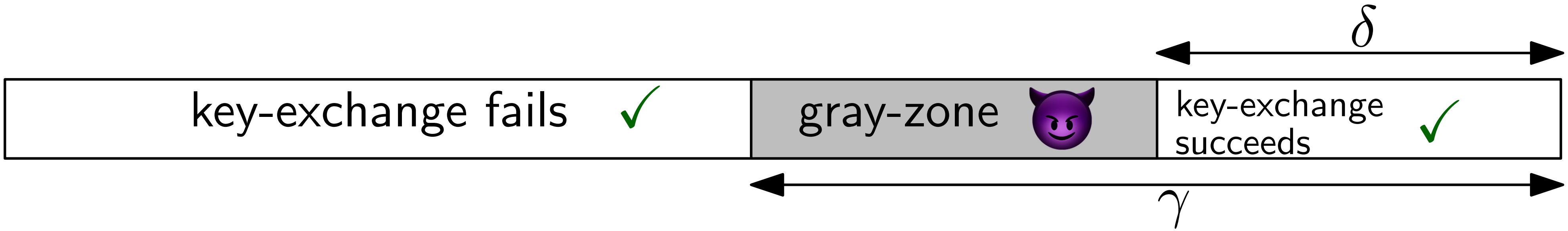
- Same session-key, if and only if the passwords are close
- $\mathcal{A}$ should not learn anything about the passwords
- No offline attacks
- $\mathcal{A}$ should only have one password guess per session

Existing Approaches from [DHP$^+$18]

- Garbled circuit based protocol for arbitrary distances
- Robust secret sharing based, for Hamming distance

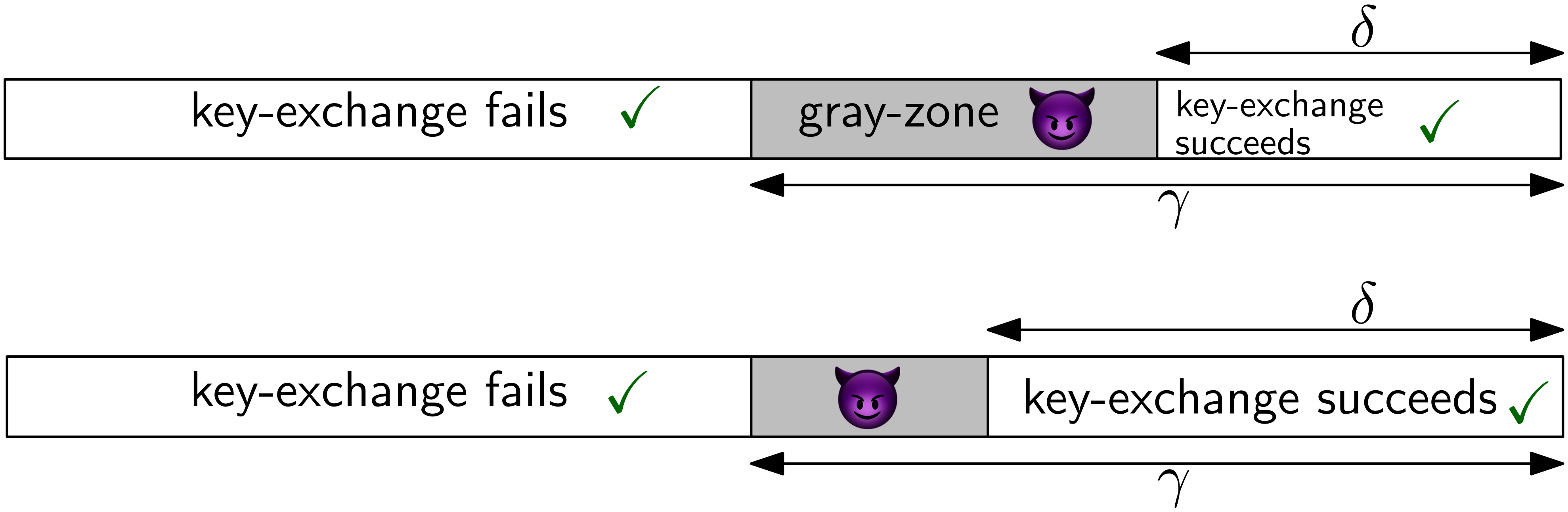
Existing Approaches from [DHP$^+$18]

- Garbled circuit based protocol for arbitrary distances
- Robust secret sharing based, for Hamming distance



Existing Approaches from [DHP$^+$18]

- Garbled circuit based protocol for arbitrary distances
- Robust secret sharing based, for Hamming distance

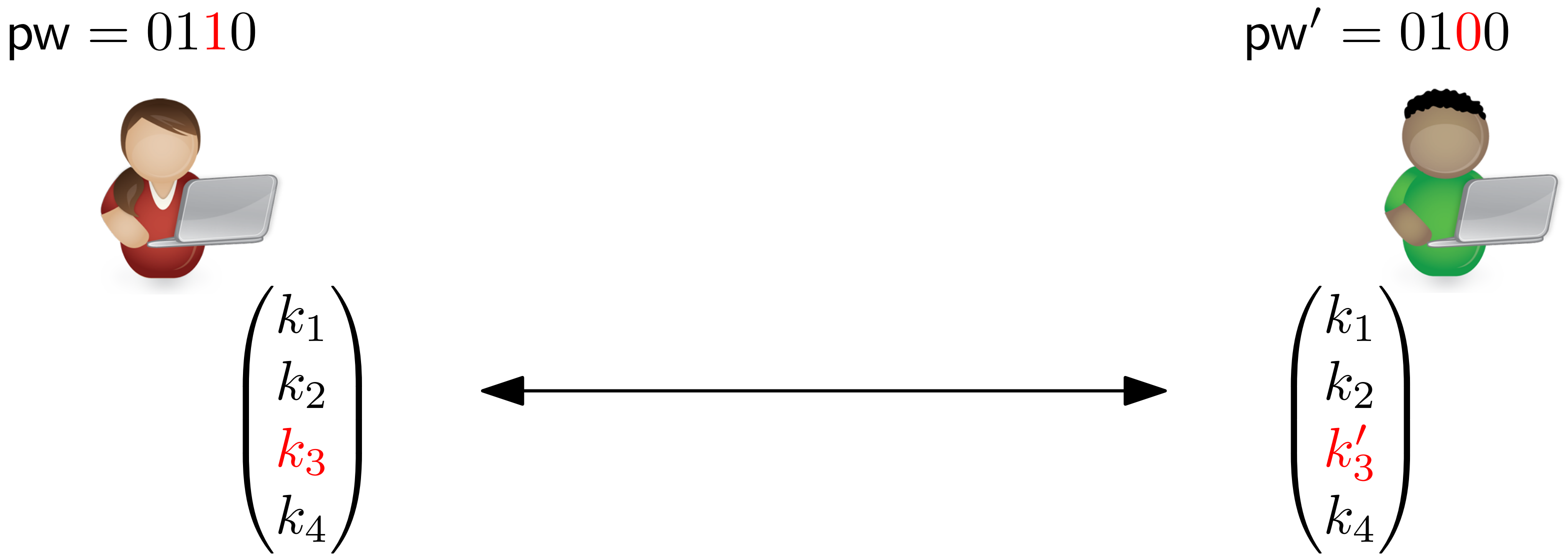


Intuition of the fPAKE-RSS protocol

$\mathsf{pw} = 01{\color{red}1}0$

$\mathsf{pw}' = 01{\color{red}0}0$

Intuition of the fPAKE-RSS protocol

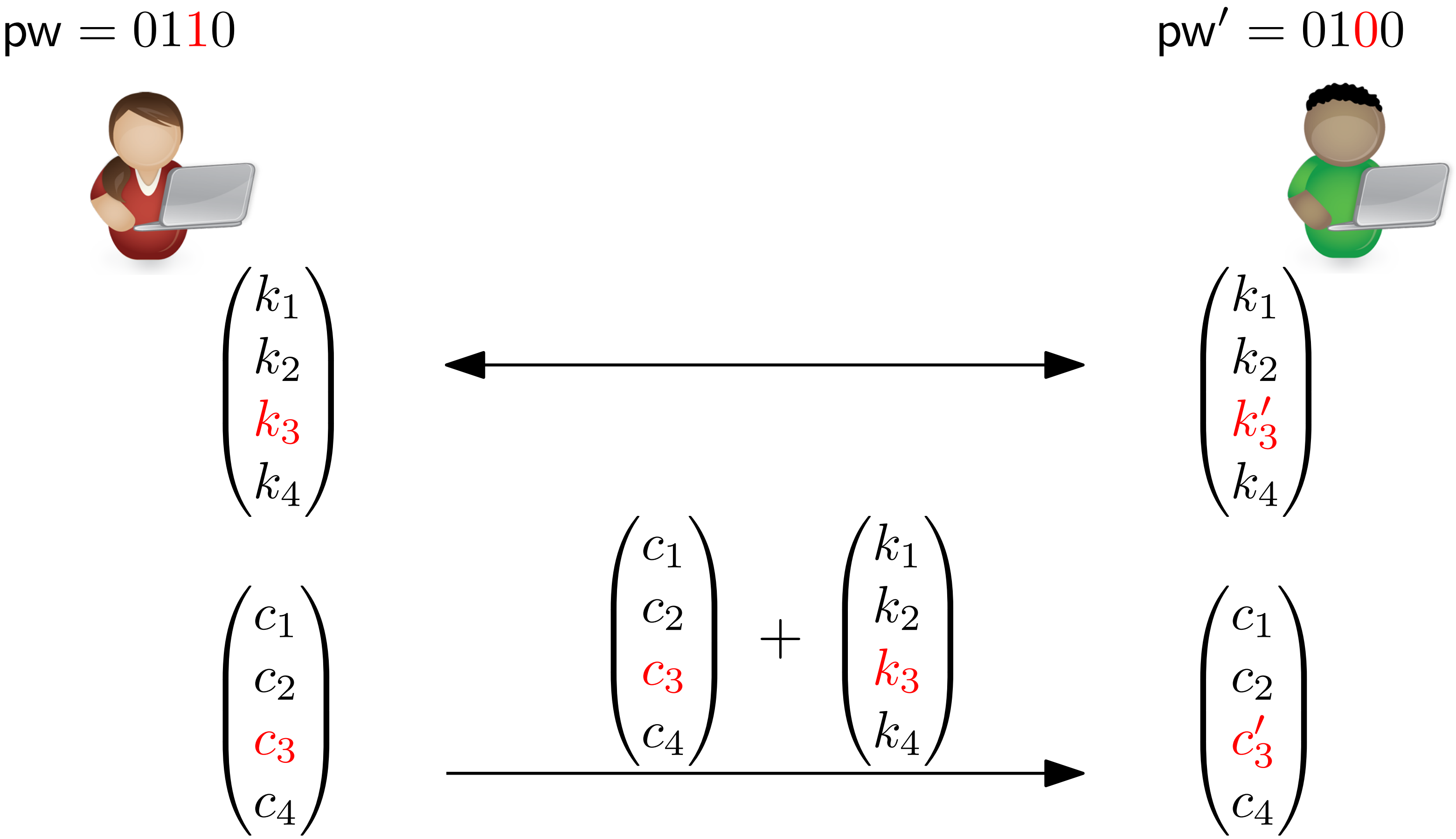


$\mathsf{pw} = 01\textcolor{red}{1}0$

$\mathsf{pw}' = 01\textcolor{red}{0}0$

$$\begin{pmatrix} k_1 \\ k_2 \\ \textcolor{red}{k_3} \\ k_4 \end{pmatrix} \longleftrightarrow \begin{pmatrix} k_1 \\ k_2 \\ \textcolor{red}{k'_3} \\ k_4 \end{pmatrix}$$

Intuition of the fPAKE-RSS protocol



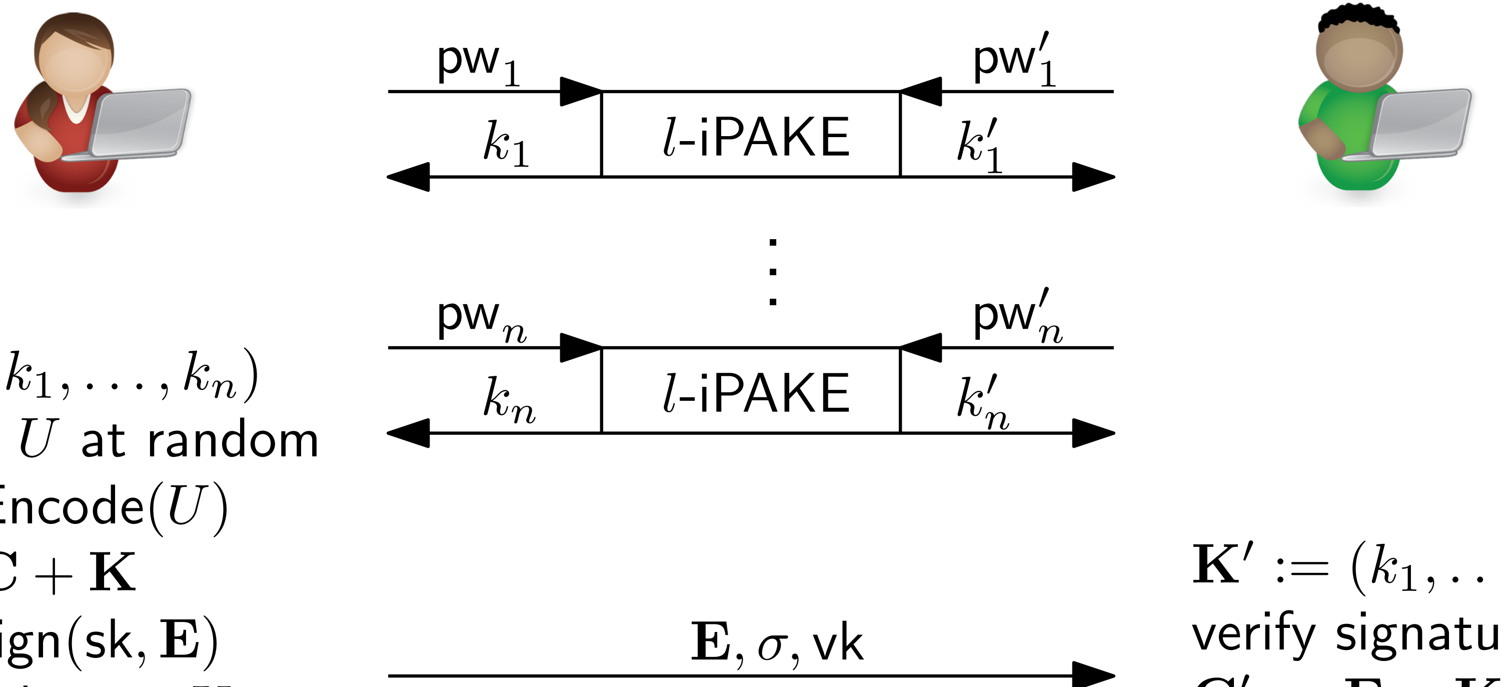
pw $= 0110$

pw$'$ $= 0100$

$$\begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix} \longleftrightarrow \begin{pmatrix} k_1 \\ k_2 \\ k'_3 \\ k_4 \end{pmatrix}$$

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} \xrightarrow{\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} + \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix}} \begin{pmatrix} c_1 \\ c_2 \\ c'_3 \\ c_4 \end{pmatrix}$$

Attack on the fPAKE-RSS protocol



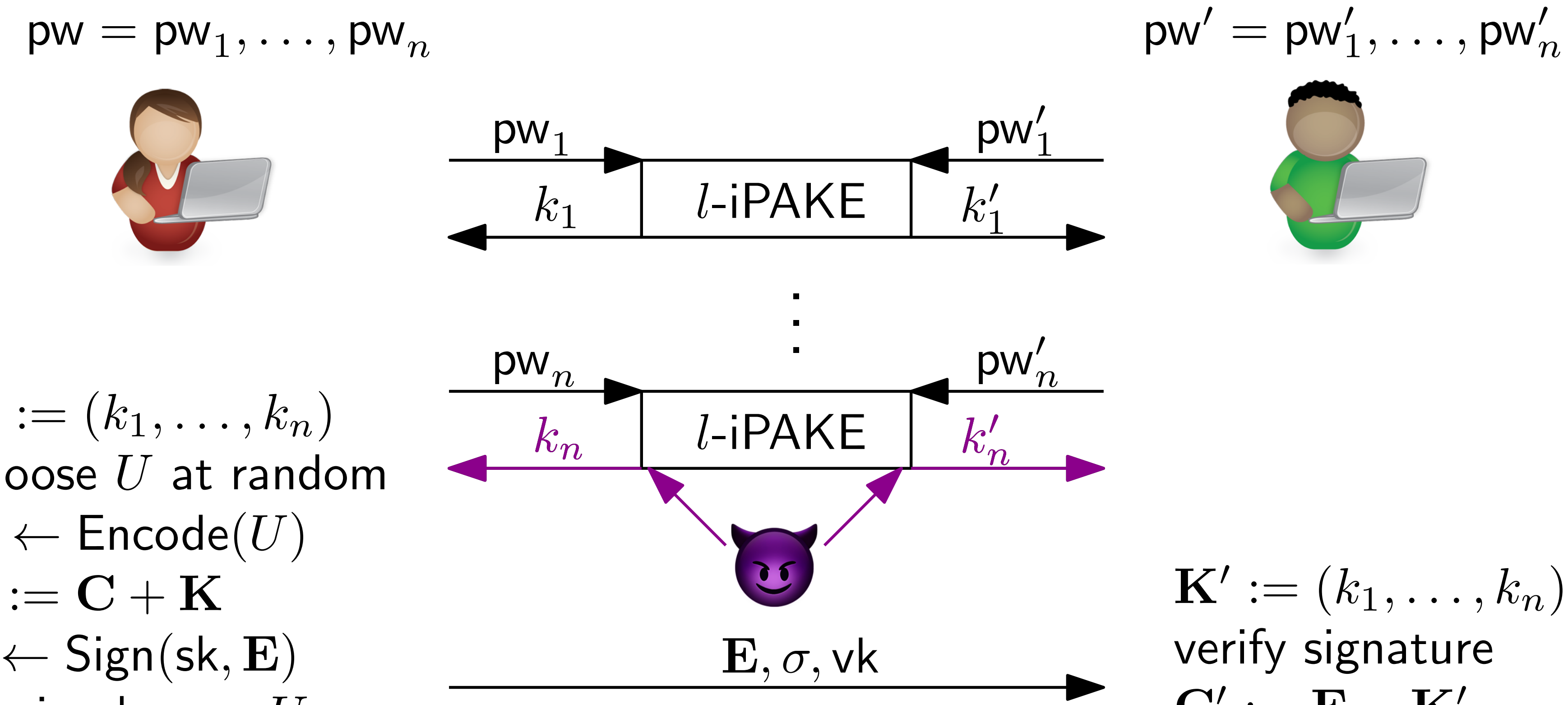
$\mathsf{pw} = \mathsf{pw}_1, \ldots, \mathsf{pw}_n$

$\mathsf{pw}' = \mathsf{pw}'_1, \ldots, \mathsf{pw}'_n$

$\mathbf{K} := (k_1, \ldots, k_n)$
choose U at random
$\mathbf{C} \leftarrow \mathsf{Encode}(U)$
$\mathbf{E} := \mathbf{C} + \mathbf{K}$
$\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathbf{E})$
session-key $:= U$

$\mathbf{K}' := (k_1, \ldots, k_n)$
verify signature
$\mathbf{C}' := \mathbf{E} - \mathbf{K}'$
$U' \leftarrow \mathsf{Decode}(\mathbf{C}')$
session-key $:= U'$

Attack on the fPAKE-RSS protocol



$\text{pw} = \text{pw}_1, \ldots, \text{pw}_n$
$\text{pw}' = \text{pw}'_1, \ldots, \text{pw}'_n$
pw_1
pw'_1
k_1
l-iPAKE
k'_1
$\vdots$
pw_n
pw'_n
k_n
l-iPAKE
k'_n
$\mathbf{E}, \sigma, \text{vk}$

$\mathbf{K} := (k_1, \ldots, k_n)$
choose U at random
$\mathbf{C} \leftarrow \text{Encode}(U)$
$\mathbf{E} := \mathbf{C} + \mathbf{K}$
$\sigma \leftarrow \text{Sign}(\text{sk}, \mathbf{E})$
session-key $:= U$

$\mathbf{K}' := (k_1, \ldots, k_n)$
verify signature
$\mathbf{C}' := \mathbf{E} - \mathbf{K}'$
$U' \leftarrow \text{Decode}(\mathbf{C}')$
session-key $:= U'$

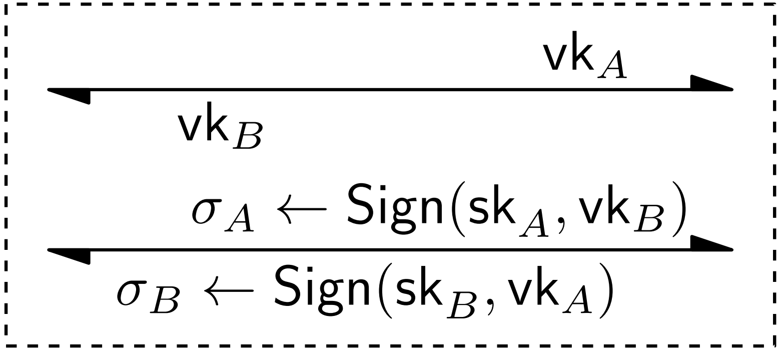
Summary of the Attack

- Requirements:
  - Both parties run the protocol
  - Both parties use the same password
  - The attacker actively interferes in the protocol
- Effect:
  - The attacker learns one bit of their choice of the password

Variations of the attack also work when the passwords are not identical

Overview of our fuzzy PAKE protocol

$\mathsf{pw} = \mathsf{pw}_1, \ldots, \mathsf{pw}_n$



vk_A

vk_B

$\sigma_A \leftarrow \mathsf{Sign}(\mathsf{sk}_A, \mathsf{vk}_B)$

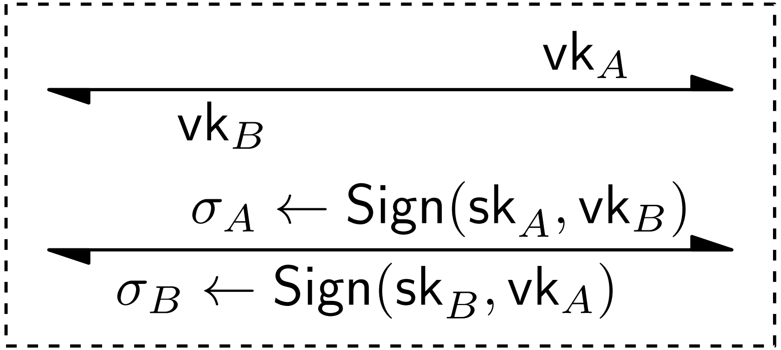
$\sigma_B \leftarrow \mathsf{Sign}(\mathsf{sk}_B, \mathsf{vk}_A)$

$\mathsf{pw}' = \mathsf{pw}'_1, \ldots, \mathsf{pw}'_n$

Overview of our fuzzy PAKE protocol

$\mathsf{pw} = \mathsf{pw}_1, \ldots, \mathsf{pw}_n$

$\mathsf{pw}' = \mathsf{pw}'_1, \ldots, \mathsf{pw}'_n$



vk_A

vk_B

$\sigma_A \leftarrow \mathsf{Sign}(\mathsf{sk}_A, \mathsf{vk}_B)$

$\sigma_B \leftarrow \mathsf{Sign}(\mathsf{sk}_B, \mathsf{vk}_A)$

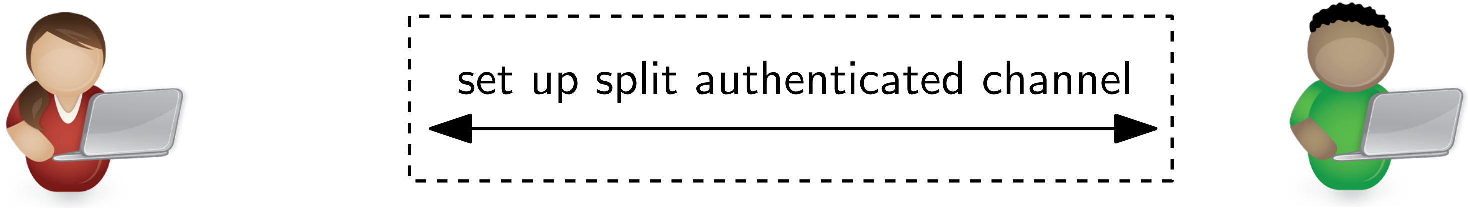
$\mathbf{C} \leftarrow \mathsf{Encode}(\mathsf{secret})$
$\mathbf{E} := \mathsf{blind}(\mathbf{C})$
$h := H(\mathbf{C})$

$\xrightarrow{\mathbf{E}, h}$

$\mathbf{C}' := \mathsf{unblind}(\mathbf{E})$
$L \leftarrow \mathsf{ListDecode}(\mathbf{C}')$
identify correct $\mathbf{C}$ via h
$\mathsf{secret} := \mathsf{Decode}(\mathbf{C})$

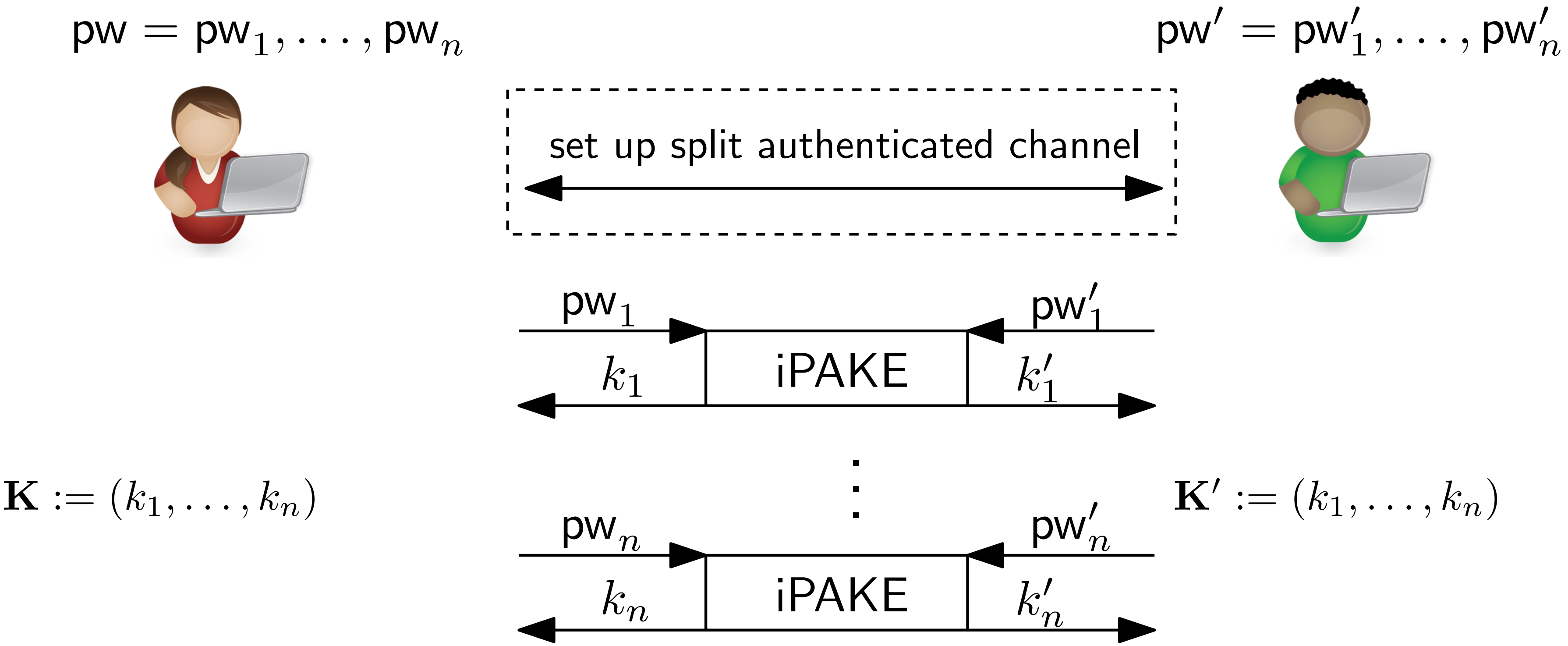
Our fuzzy PAKE protocol

$\mathsf{pw} = \mathsf{pw}_1, \ldots, \mathsf{pw}_n$ $\mathsf{pw}' = \mathsf{pw}'_1, \ldots, \mathsf{pw}'_n$



set up split authenticated channel

Our fuzzy PAKE protocol



$\mathsf{pw} = \mathsf{pw}_1, \ldots, \mathsf{pw}_n$

$\mathsf{pw}' = \mathsf{pw}'_1, \ldots, \mathsf{pw}'_n$

set up split authenticated channel

pw_1 iPAKE pw'_1

k_1 k'_1

$\vdots$

pw_n iPAKE pw'_n

k_n k'_n

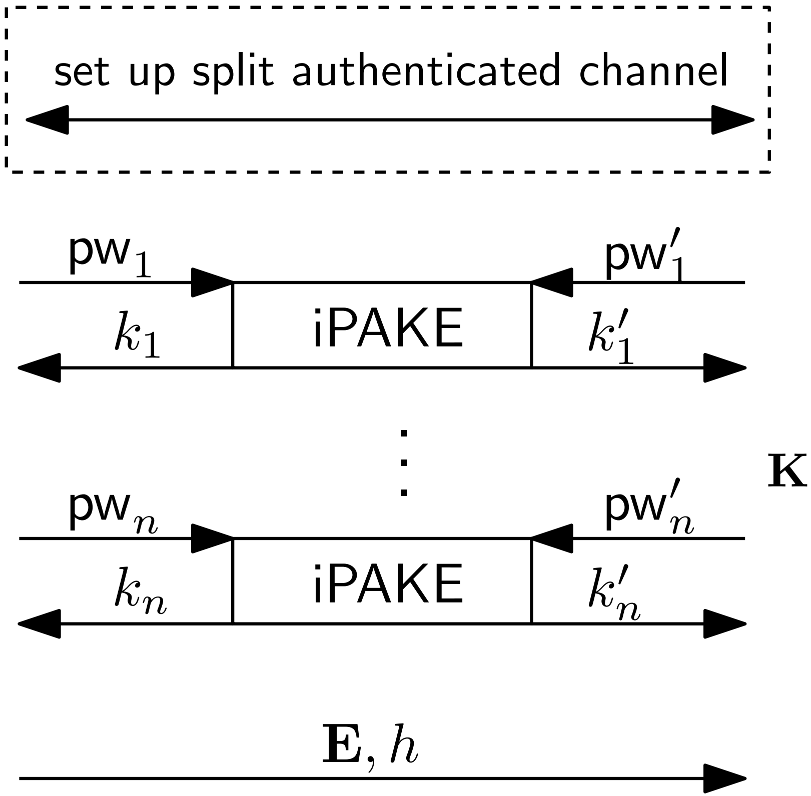
$\mathbf{K} := (k_1, \ldots, k_n)$

$\mathbf{K}' := (k_1, \ldots, k_n)$

Our fuzzy PAKE protocol

$\mathsf{pw} = \mathsf{pw}_1, \ldots, \mathsf{pw}_n$

$\mathsf{pw}' = \mathsf{pw}'_1, \ldots, \mathsf{pw}'_n$



$\mathbf{K} := (k_1, \ldots, k_n)$

$\mathbf{K}' := (k_1, \ldots, k_n)$

$\mathbf{s} := (s_1, \ldots s_k) \xleftarrow{\$} \mathbb{F}_q^k$

$\mathbf{C} \leftarrow \mathsf{Encode}(\mathbf{s})$

$\mathbf{E} := \mathbf{C} \oplus \mathbf{K}$

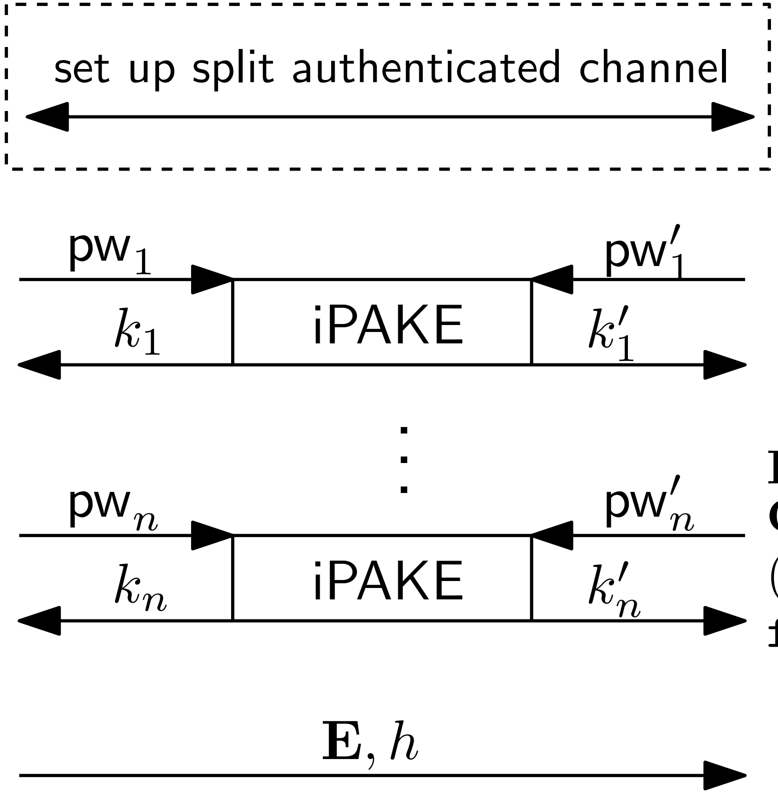
$h := H_0(\mathbf{C})$

$\mathsf{session\text{-}key} := H_1(s_1)$

Our fuzzy PAKE protocol

$\mathsf{pw} = \mathsf{pw}_1, \ldots, \mathsf{pw}_n$

$\mathsf{pw}' = \mathsf{pw}'_1, \ldots, \mathsf{pw}'_n$



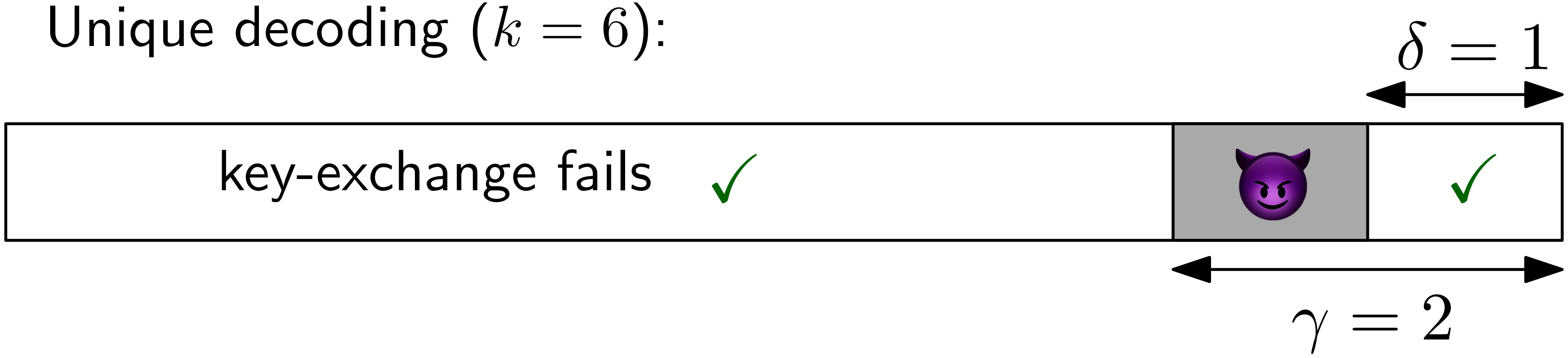
```
$\mathbf{K} := (k_1, \ldots, k_n)$

$\mathbf{s} := (s_1, \ldots s_k) \xleftarrow{\$} \mathbb{F}_q^k$
$\mathbf{C} \leftarrow \mathsf{Encode}(\mathbf{s})$
$\mathbf{E} := \mathbf{C} \oplus \mathbf{K}$
$h := H_0(\mathbf{C})$
session-key $:= H_1(s_1)$
```

```
$\mathbf{K}' := (k_1, \ldots, k_n)$
$\mathbf{C}' := \mathbf{E} \oplus \mathbf{K}'$
$(\mathbf{C}^1, \ldots, \mathbf{C}^l) \leftarrow \mathsf{ListDecode}(\mathbf{C}')$
for $j \in \{1, \ldots, l\}$:
   if $H_0(\mathbf{C}^j) = h$:
      set $\mathbf{s}' := \mathsf{Decode}(\mathbf{C}^j)$
session-key $:= H_1(s'_1)$
```

Improvements through list decoding

Correcting 1 error of an 8 character password (with an $[n, k]$ Code)

Unique decoding ($k = 6$):



$\delta = 1$

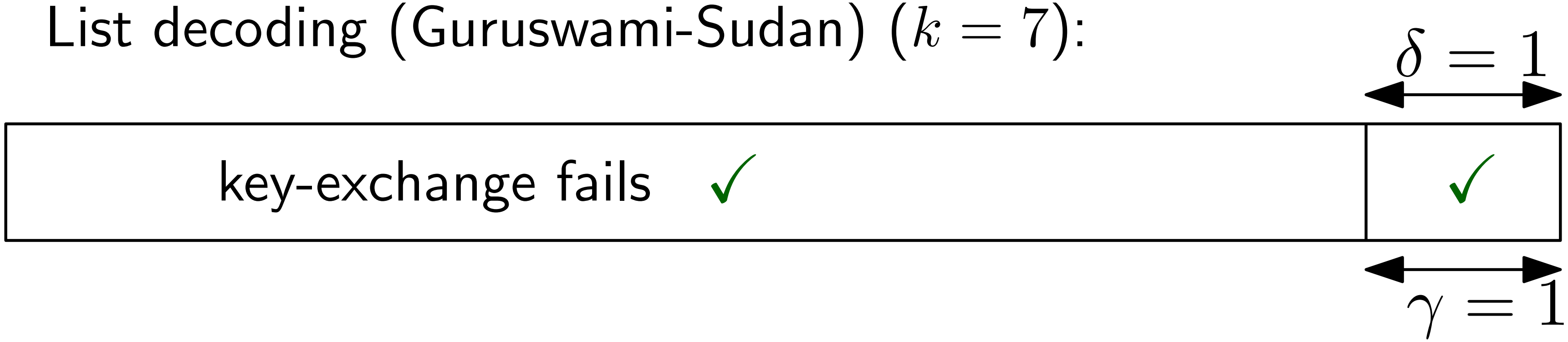
key-exchange fails ✓

😈

✓

$\gamma = 2$

List decoding (Guruswami-Sudan) ($k = 7$):



$\delta = 1$

key-exchange fails ✓

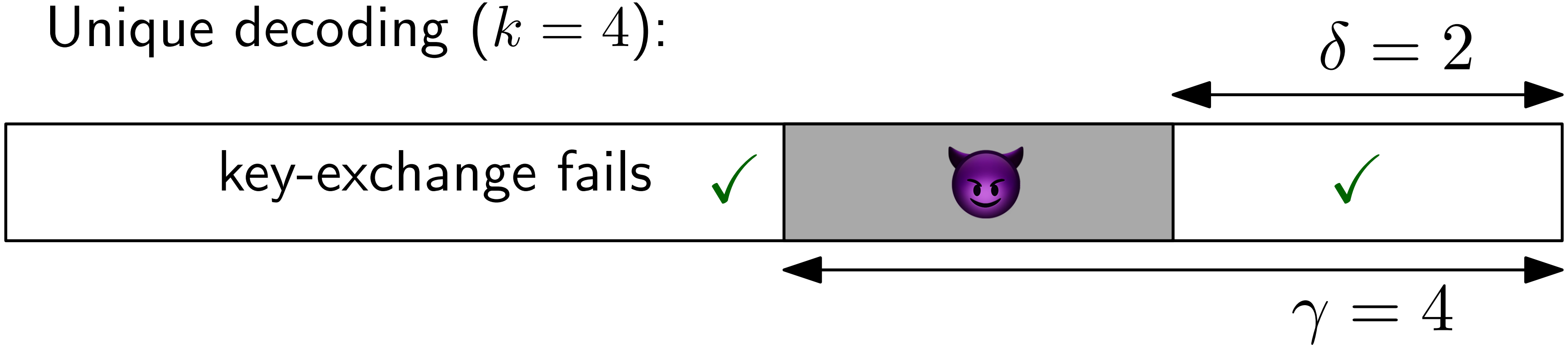
✓

$\gamma = 1$

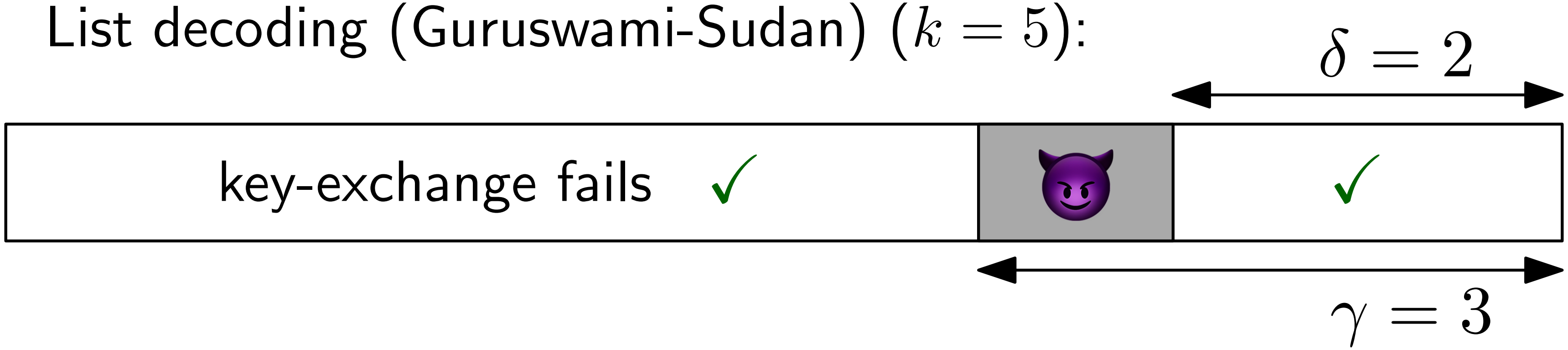
Improvements through list decoding

Correcting 2 error of an 8 character password (with an $[n, k]$ Code)

Unique decoding ($k = 4$):



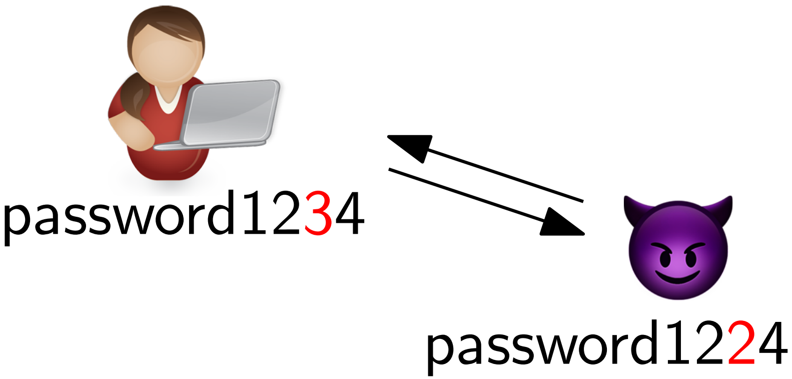
List decoding (Guruswami-Sudan) ($k = 5$):



Summary

- Attack on existing ECC-based fPAKE protocol
- New ECC-based fPAKE protocol
  - fix weakness of old protocol
  - better decoding via list decoding and hint
  - security analysis in UC framework

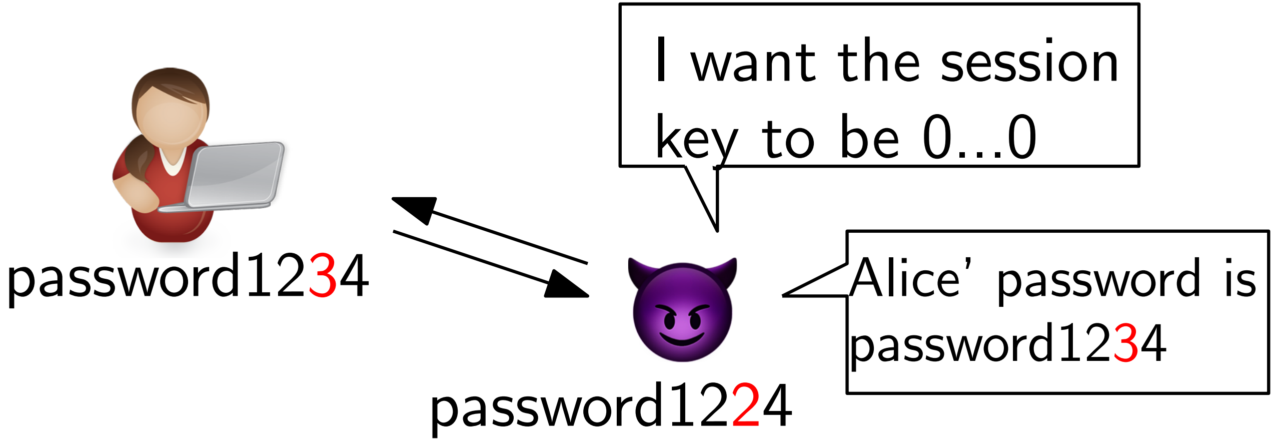
Outlook and PQ-Instantiations



password1234

password1224

Outlook and PQ-Instantiations



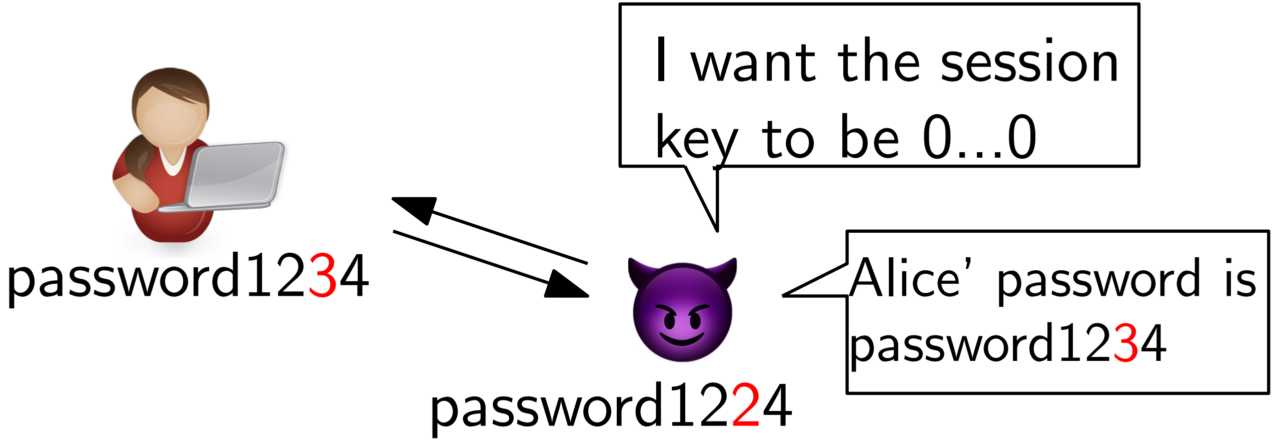
password1234

I want the session key to be 0...0

password1224

Alice' password is password1234

Outlook and PQ-Instantiations



I want the session key to be 0...0

password1234

Alice’ password is password1234

password1224

- Instantiation of underlying PAKE needs to be *implicit-only*
    - CAKE protocol of [BCP$^+$23] is a promising candidate

Questions

Paper at `ia.cr/2023/1415`