Key Exchange in the Post-Snowden Era: Universally Composable Subversion-Resilient PAKE

Suvradip Chakraborty
VISA Research

Lorenzo Magliocco
Sapienza University of Rome

Bernardo Magri
University of Manchester
Primev

Daniele Venturi
Sapienza University of Rome

ASIACRYPT 2024

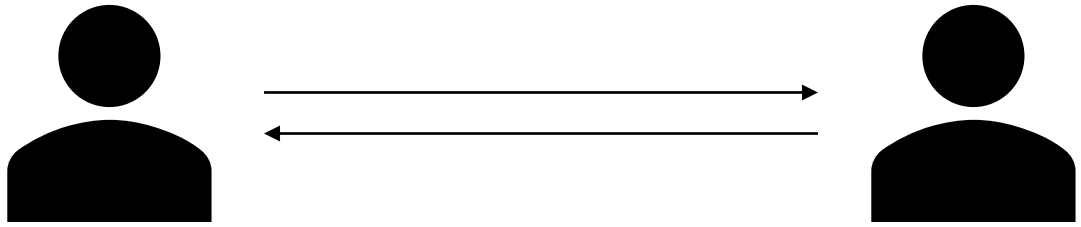
Bird's eye view

Intro to subversion	$\mathcal{F}_{RE}$ from srOT
$\mathcal{F}_{RE}$ to $\mathcal{F}_{PAKE}$	Wrapping up

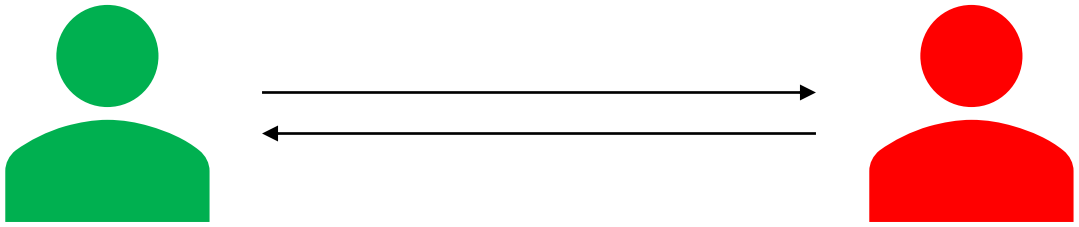
Bird's eye view

Intro to subversion	$\mathcal{F}_{RE}$ from srOT
$\mathcal{F}_{RE}$ to $\mathcal{F}_{PAKE}$	Wrapping up

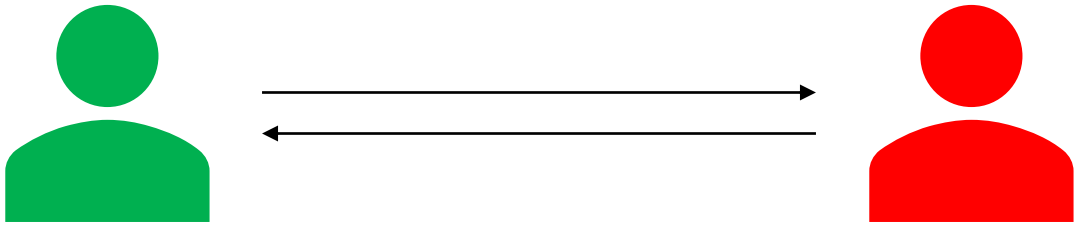
“Traditional” MPC



“Traditional” MPC

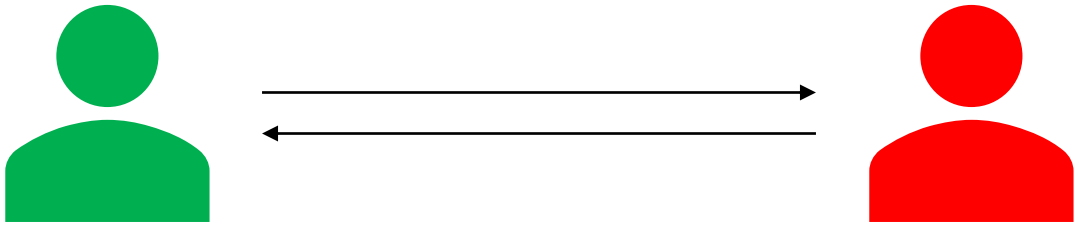


“Traditional” MPC



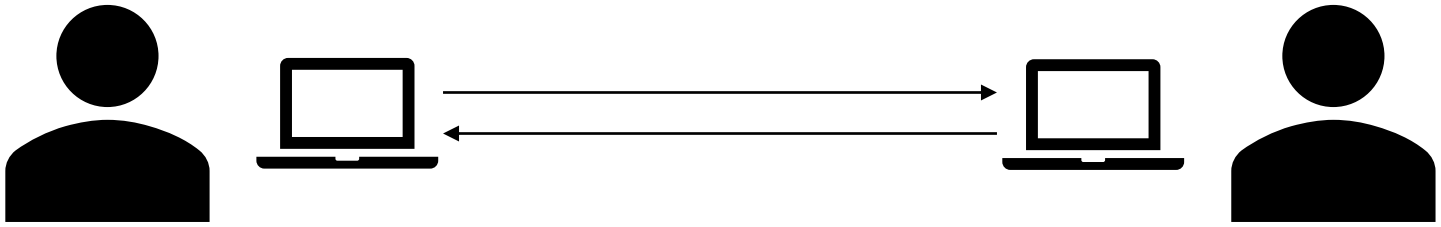
- **Malicious corruptions** are very general.

“Traditional” MPC

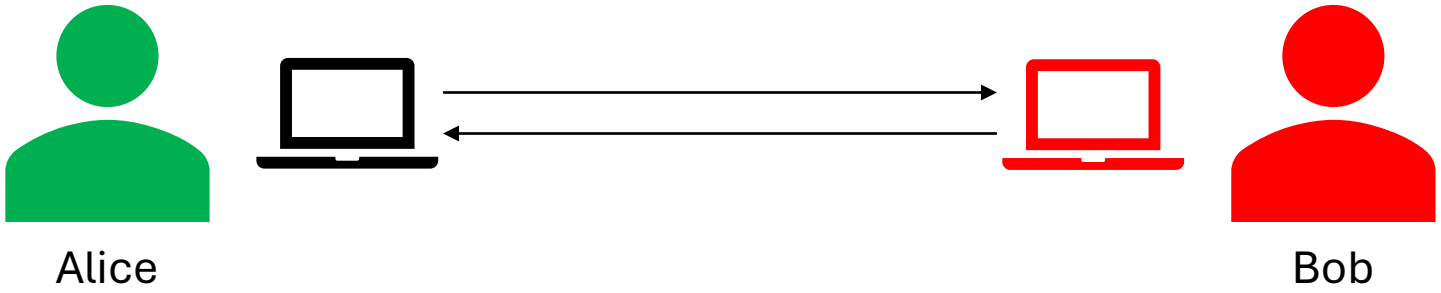


- **Malicious corruptions** are very general.
- What about **honest parties**?

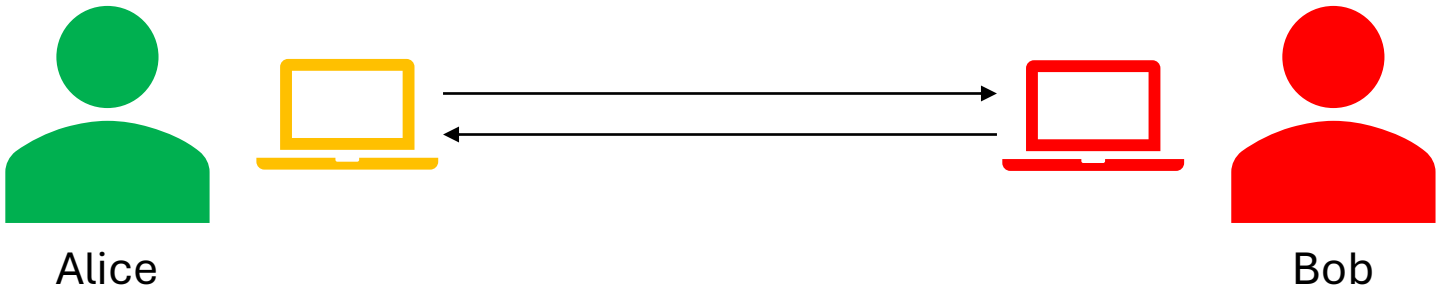
MPC with Subversion Attacks



MPC with Subversion Attacks



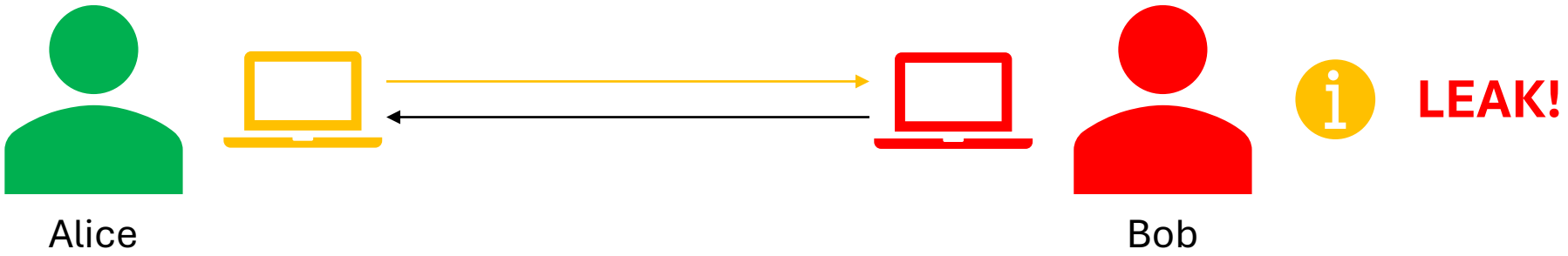
MPC with Subversion Attacks



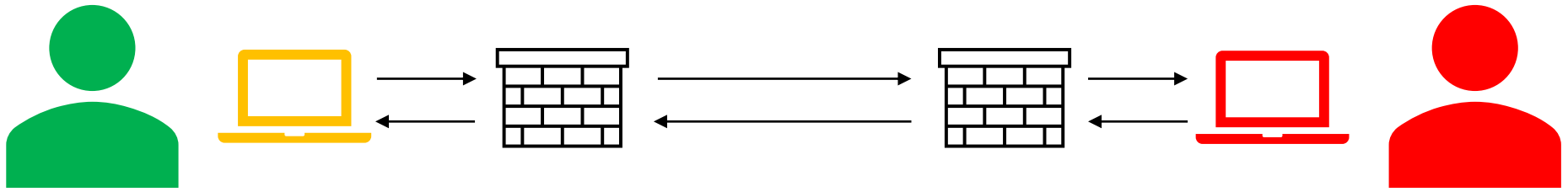
MPC with Subversion Attacks

MPC with Subversion Attacks

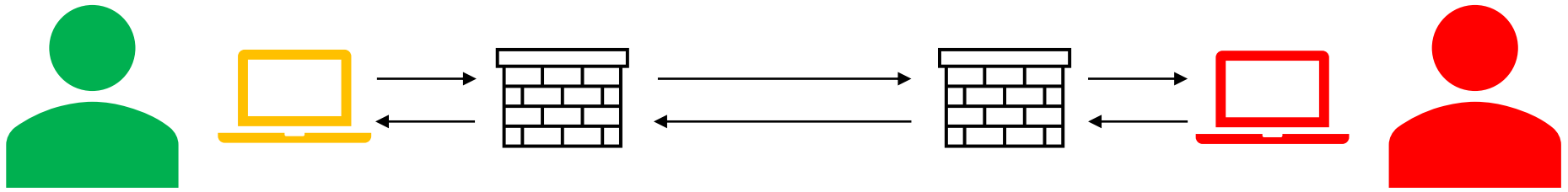
MPC with Subversion Attacks



Cryptographic Reverse Firewalls

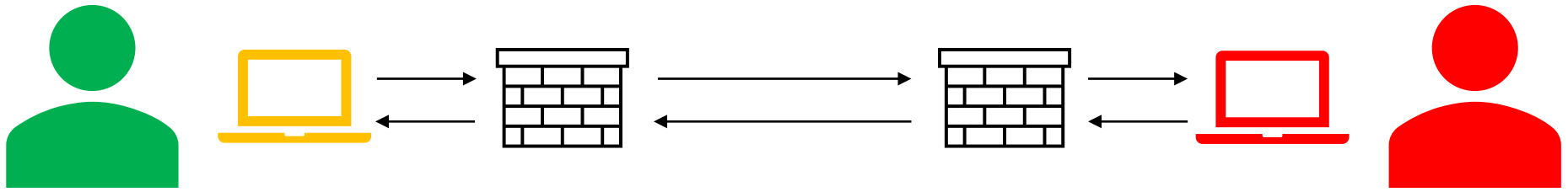


Cryptographic Reverse Firewalls



- First formalized in [**MS14**] (EUROCRYPT'15).

Cryptographic Reverse Firewalls



- First formalized in [**MS14**] (EUROCRYPT'15).
- Extended to the UC framework in [**CMNV22**] (EUROCRYPT'22).

Bird's eye view

Intro to subversion	**$\mathcal{F}_{RE}$ from srOT**
$\mathcal{F}_{RE}$ to $\mathcal{F}_{PAKE}$	Wrapping up

PAKE from OT

- **Starting point:** PAKE from OT due to **[CDVW12]** (PKC'12).

PAKE from OT

- **Starting point:** PAKE from OT due to **[CDVW12]** (PKC'12).
- **First**: PAKE from OT assuming authenticated channels (AKA **Randomized Equality**).

PAKE from OT

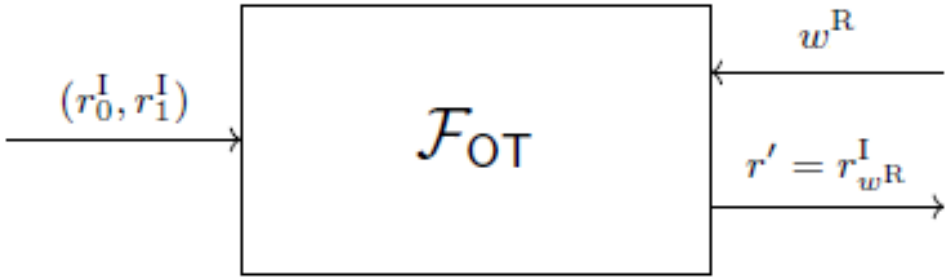
- **Starting point:** PAKE from OT due to **[CDVW12]** (PKC'12).
- **First**: PAKE from OT assuming authenticated channels (AKA **Randomized Equality**).
- **Then:** Remove authenticated channels.

Building block for PAKE

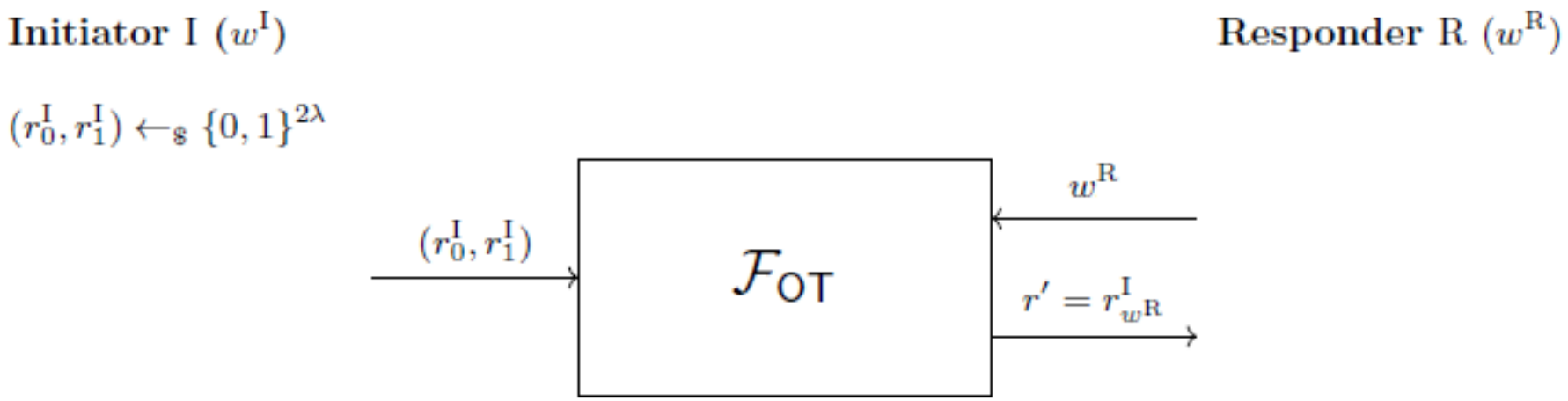
Initiator I (w^{I})

Responder R (w^{R})

$(r_0^{\mathrm{I}}, r_1^{\mathrm{I}}) \leftarrow_{\$} \{0,1\}^{2\lambda}$

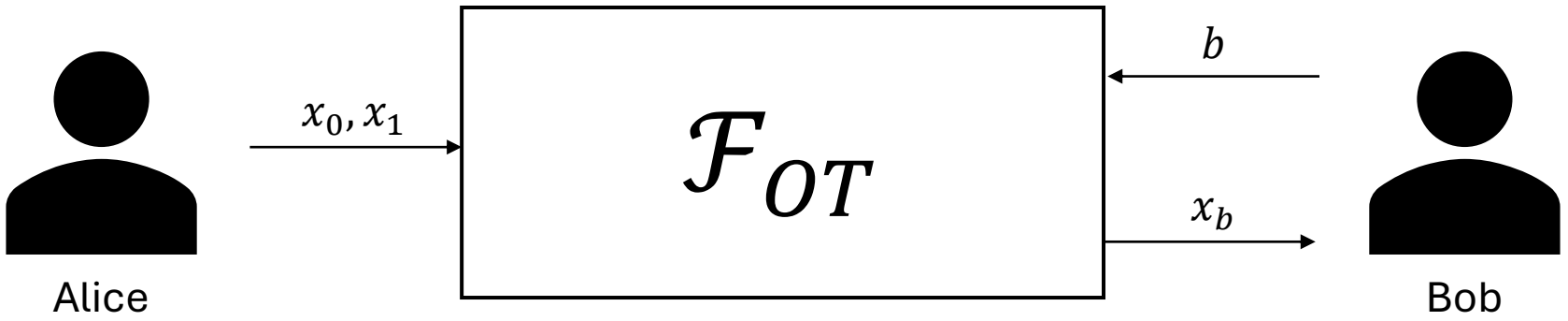


Building block for PAKE

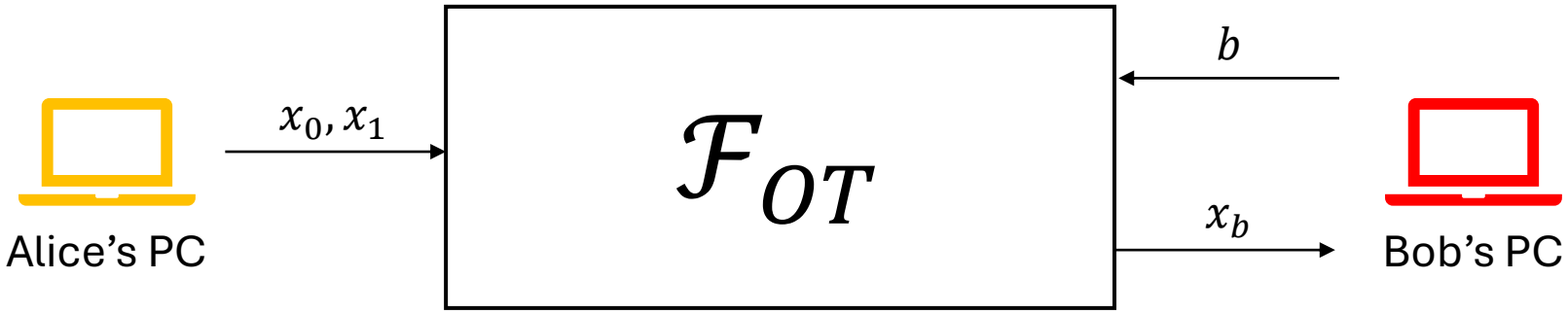


- Both parties do the same for **every bit** of the password.

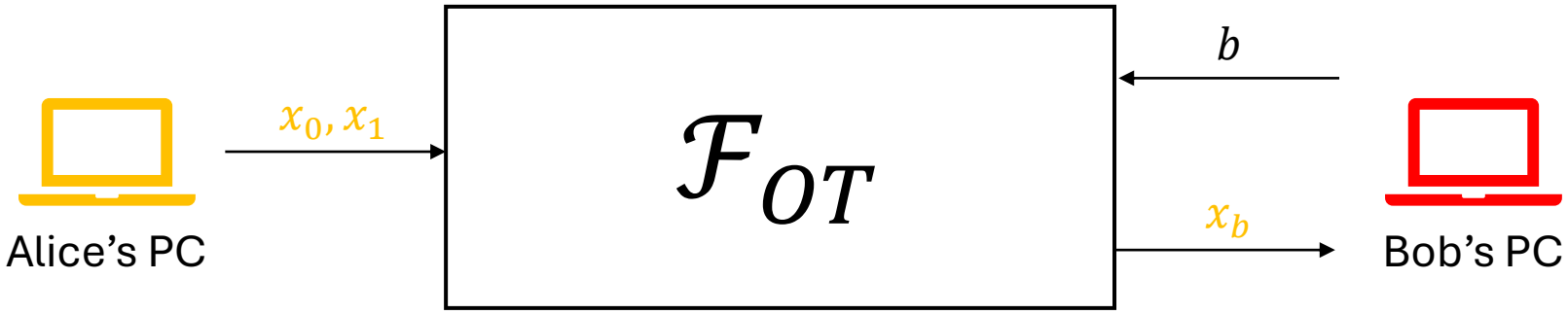
Oblivious Transfer in the presence of subversion



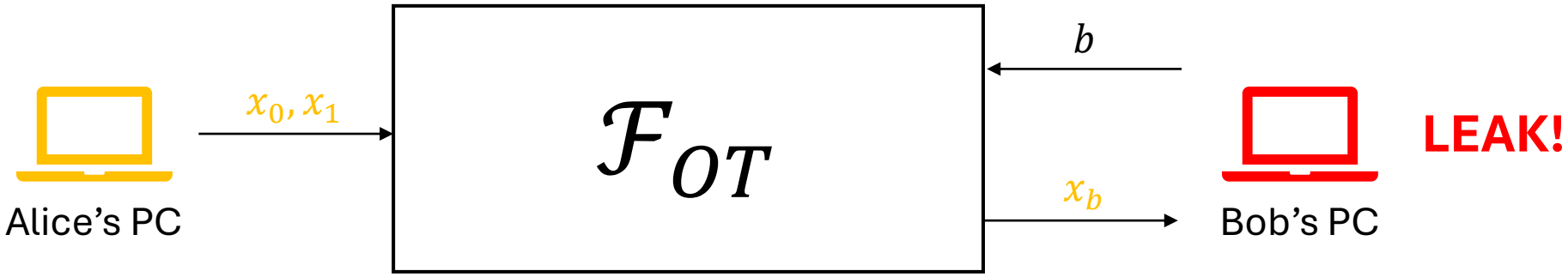
Oblivious Transfer in the presence of subversion



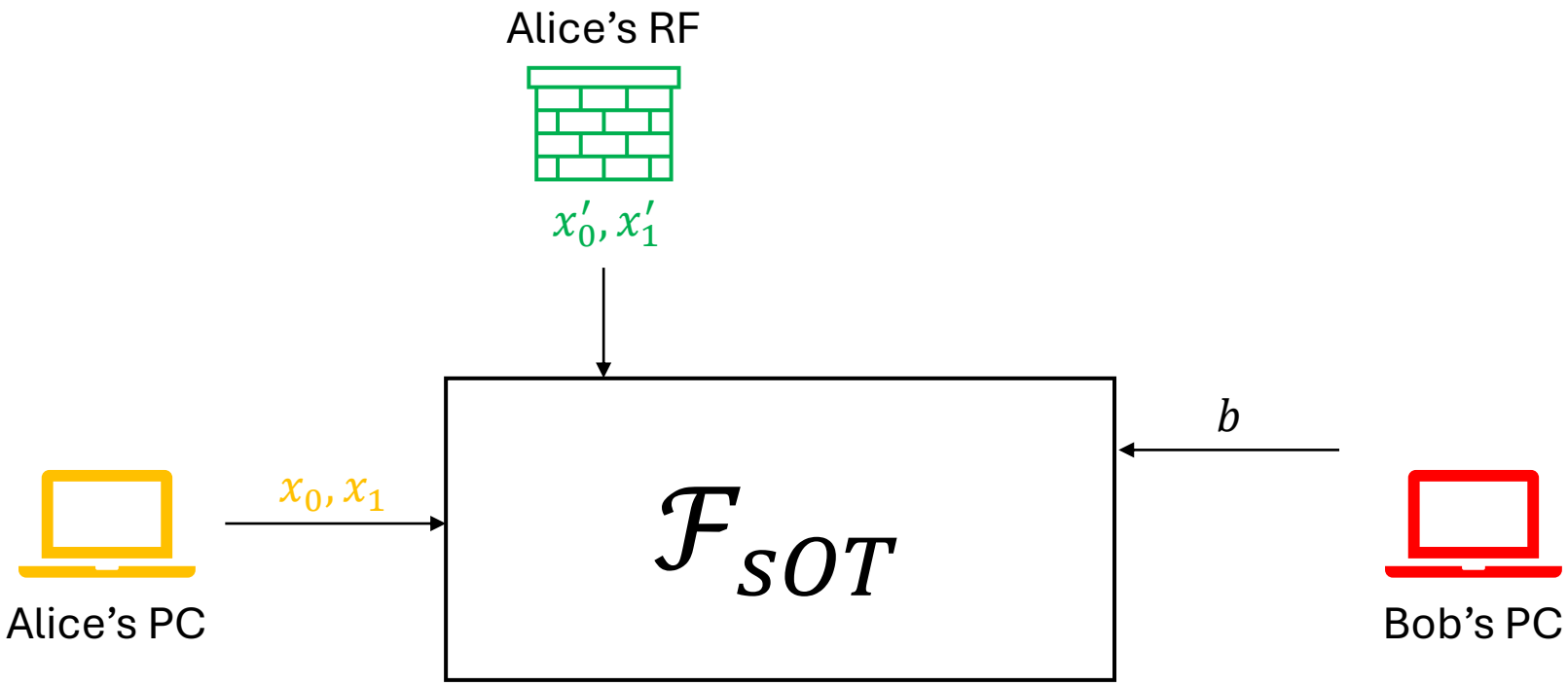
Oblivious Transfer in the presence of subversion



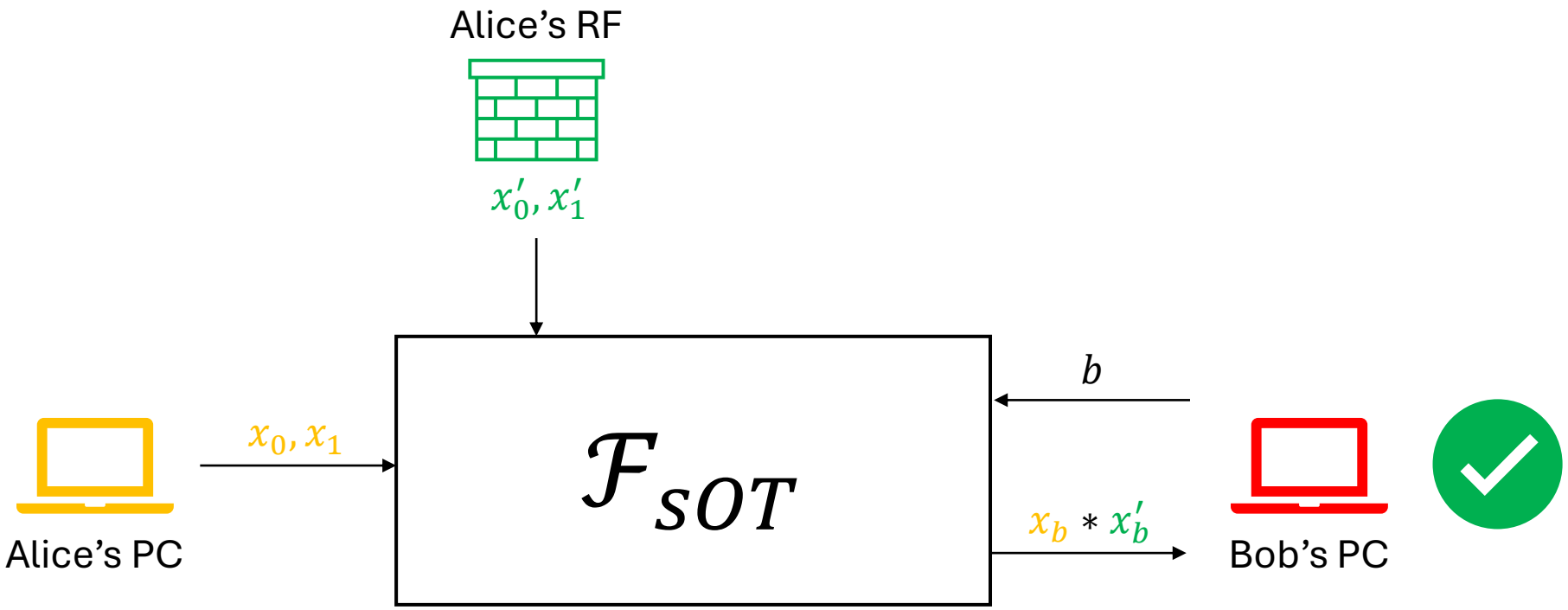
Oblivious Transfer in the presence of subversion



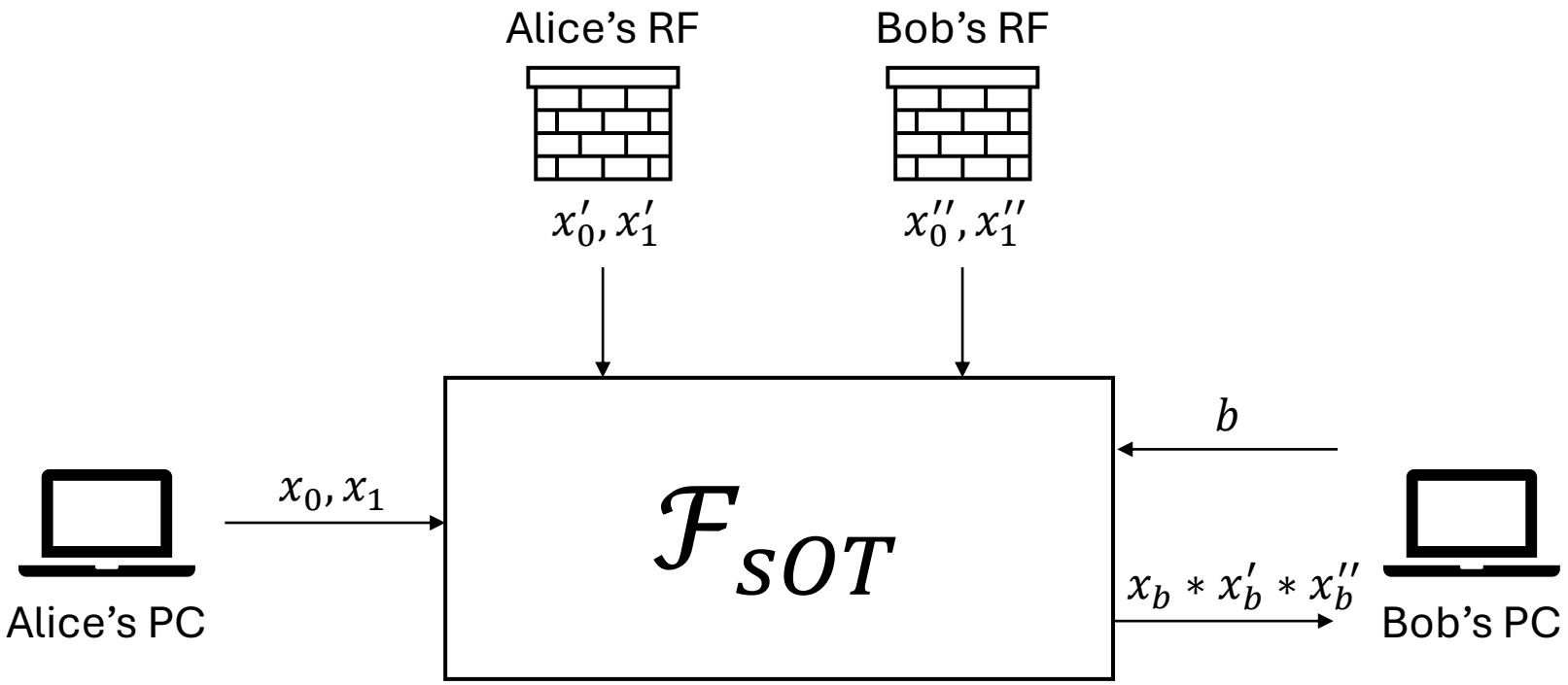
Oblivious Transfer in the presence of subversion



Oblivious Transfer in the presence of subversion



Oblivious Transfer in the presence of subversion



$\mathcal{F}_{OT}$ from Dual-Mode Cryptosystems [PVW08]

Sender $\mathrm{S}(x_0, x_1)$ | **Receiver** $\mathrm{R}(\sigma)$

$\mathrm{CRS} = \mathsf{crs}$

$(sk, pk) \leftarrow_{\$} \mathsf{KeyGen}(\sigma)$

$pk \longleftarrow$

$\mathcal{F}_{OT}$ from Dual-Mode Cryptosystems [PVW08]

Sender $\mathrm{S}(x_0, x_1)$ | **Receiver** $\mathrm{R}(\sigma)$

$\mathrm{CRS} = \mathsf{crs}$

$(sk, pk) \leftarrow_{\$} \mathsf{KeyGen}(\sigma)$

$\xleftarrow{pk}$

$y_b \leftarrow_{\$} \mathsf{Enc}(pk, b, x_b)$

$\mathcal{F}_{OT}$ from Dual-Mode Cryptosystems [PVW08]

Sender $\mathrm{S}(x_0, x_1)$ | **Receiver** $\mathrm{R}(\sigma)$

$\mathrm{CRS} = \mathsf{crs}$

$(sk, pk) \leftarrow_{\$} \mathsf{KeyGen}(\sigma)$

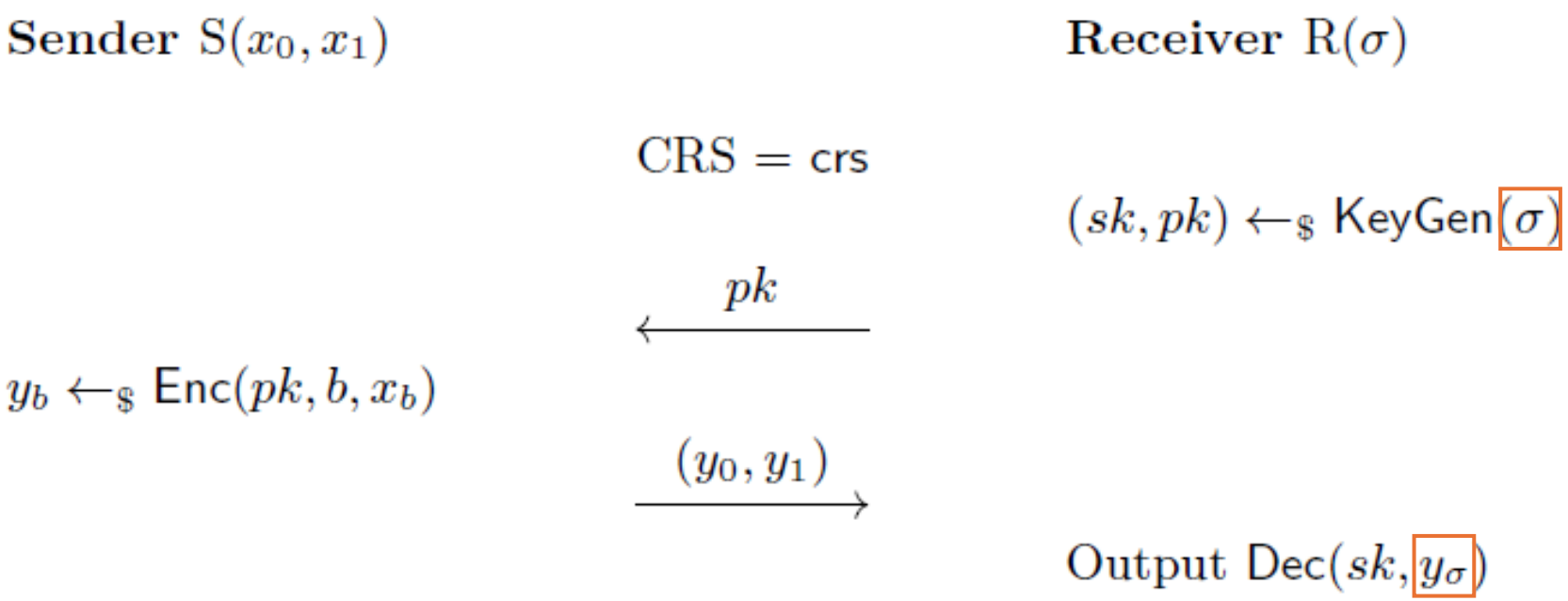
$\xleftarrow{pk}$

$y_b \leftarrow_{\$} \mathsf{Enc}(pk, b, x_b)$

$\xrightarrow{(y_0, y_1)}$

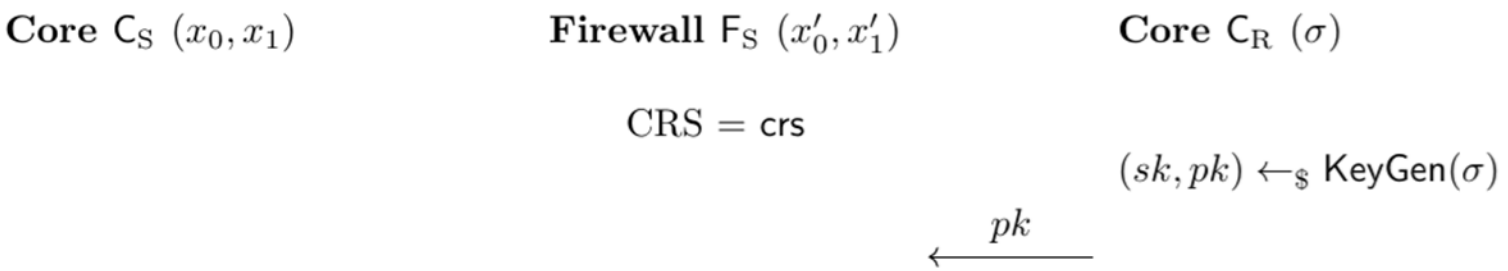
Output $\mathrm{Dec}(sk, y_\sigma)$

$\mathcal{F}_{OT}$ from Dual-Mode Cryptosystems [PVW08]

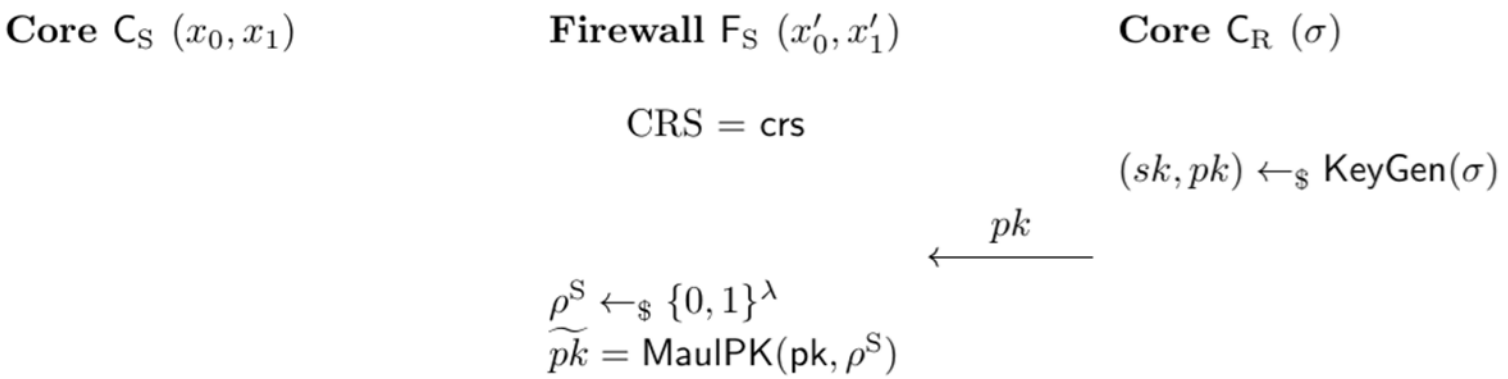


- Only y_b for which $b = \sigma$ can be decrypted.

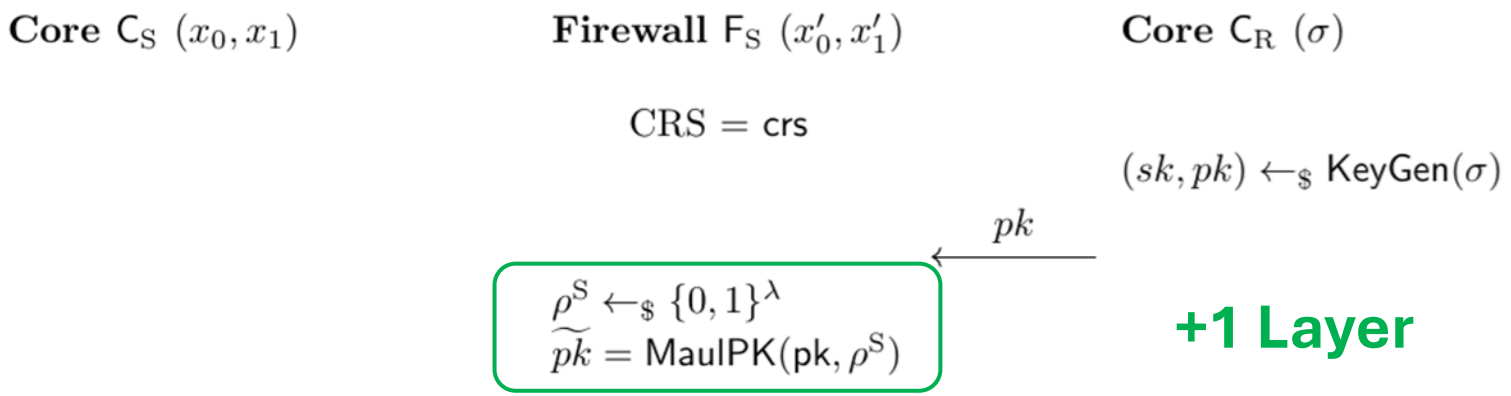
Instantiating $\mathcal{F}_{sOT}$



Instantiating $\mathcal{F}_{sOT}$

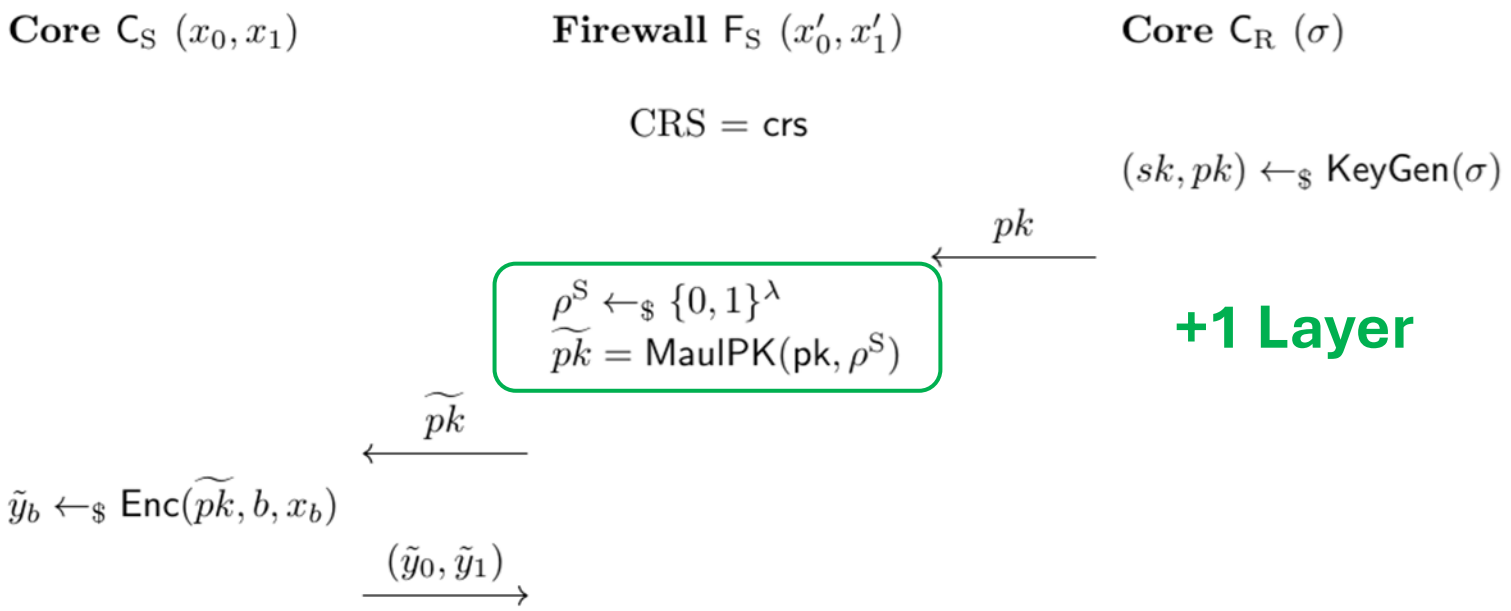


Instantiating $\mathcal{F}_{sOT}$

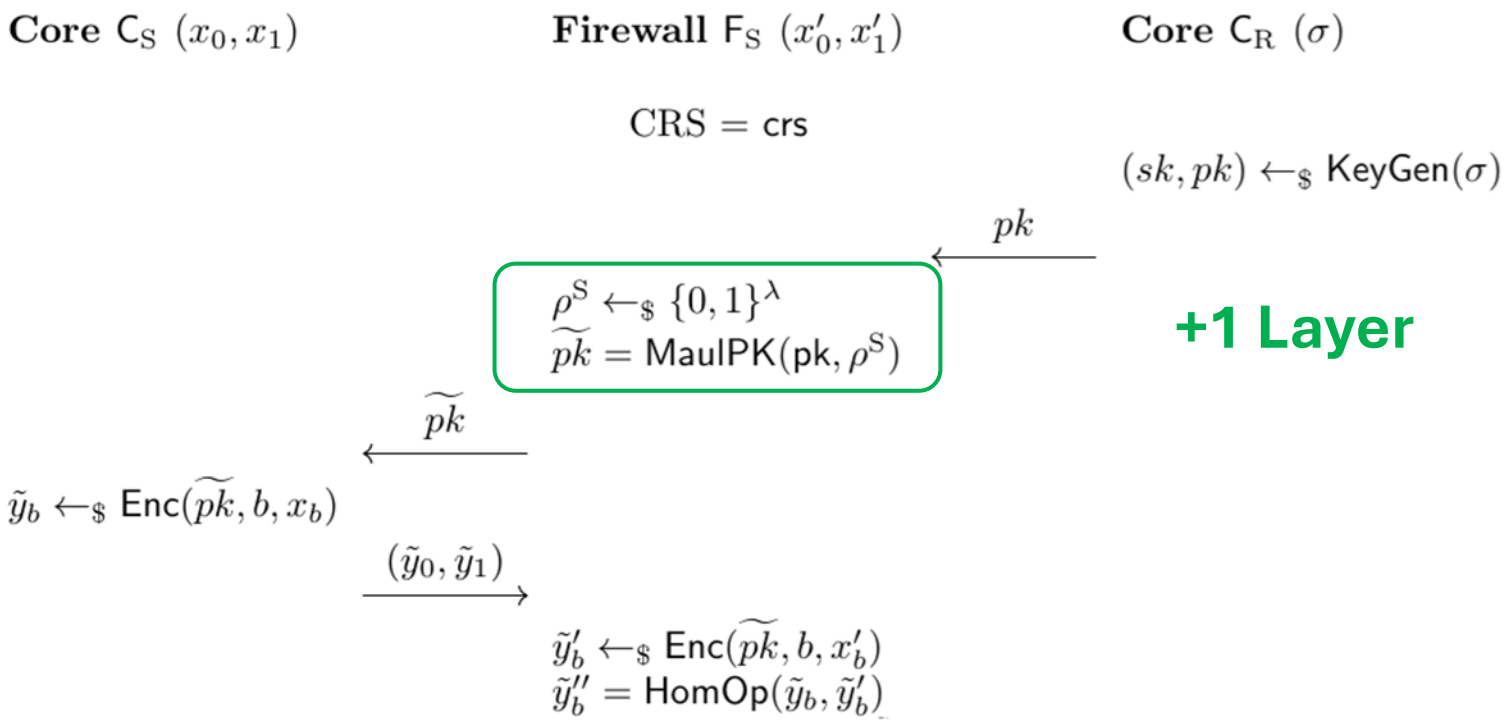


+1 Layer

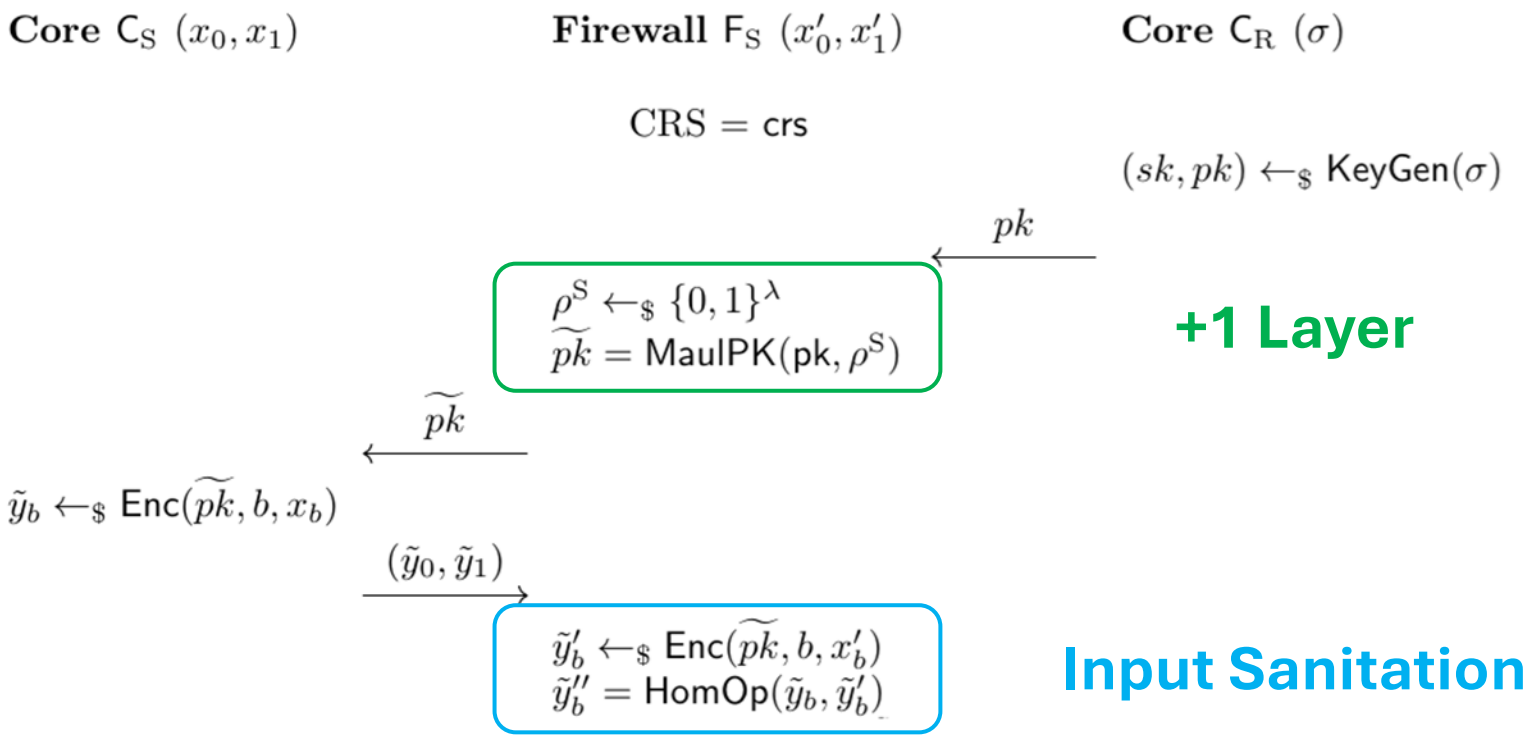
Instantiating $\mathcal{F}_{sOT}$



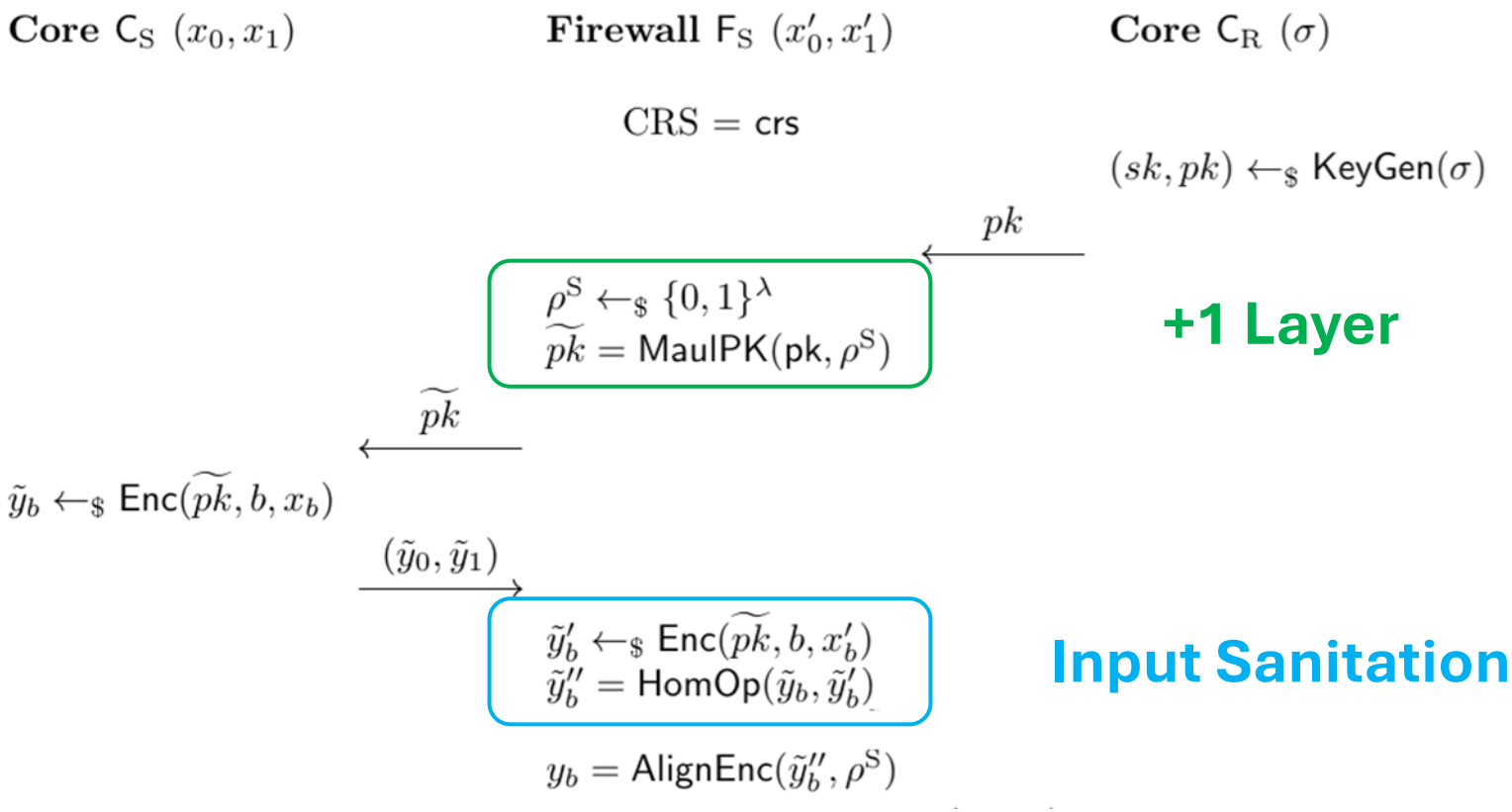
Instantiating $\mathcal{F}_{sOT}$



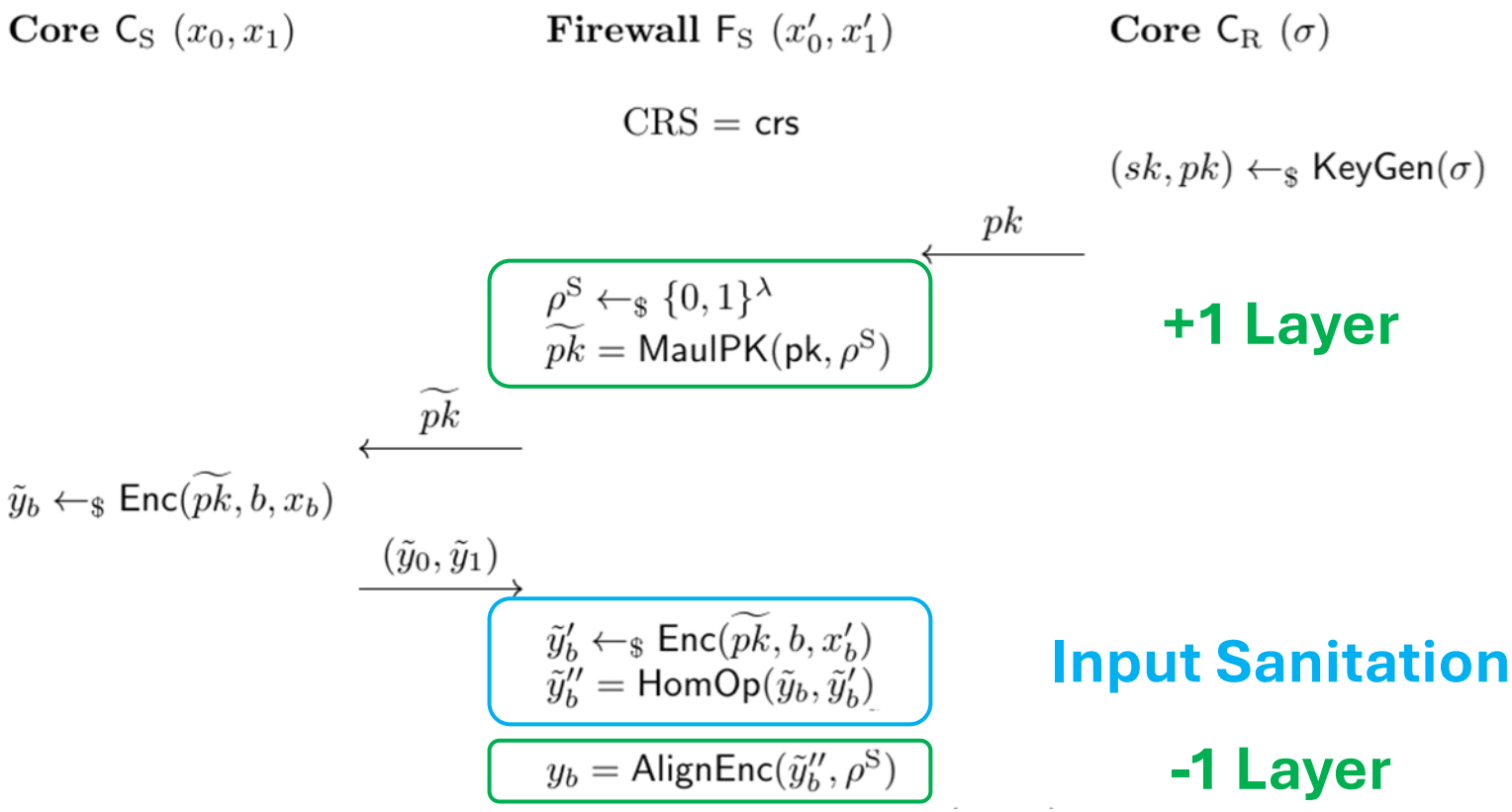
Instantiating $\mathcal{F}_{sOT}$



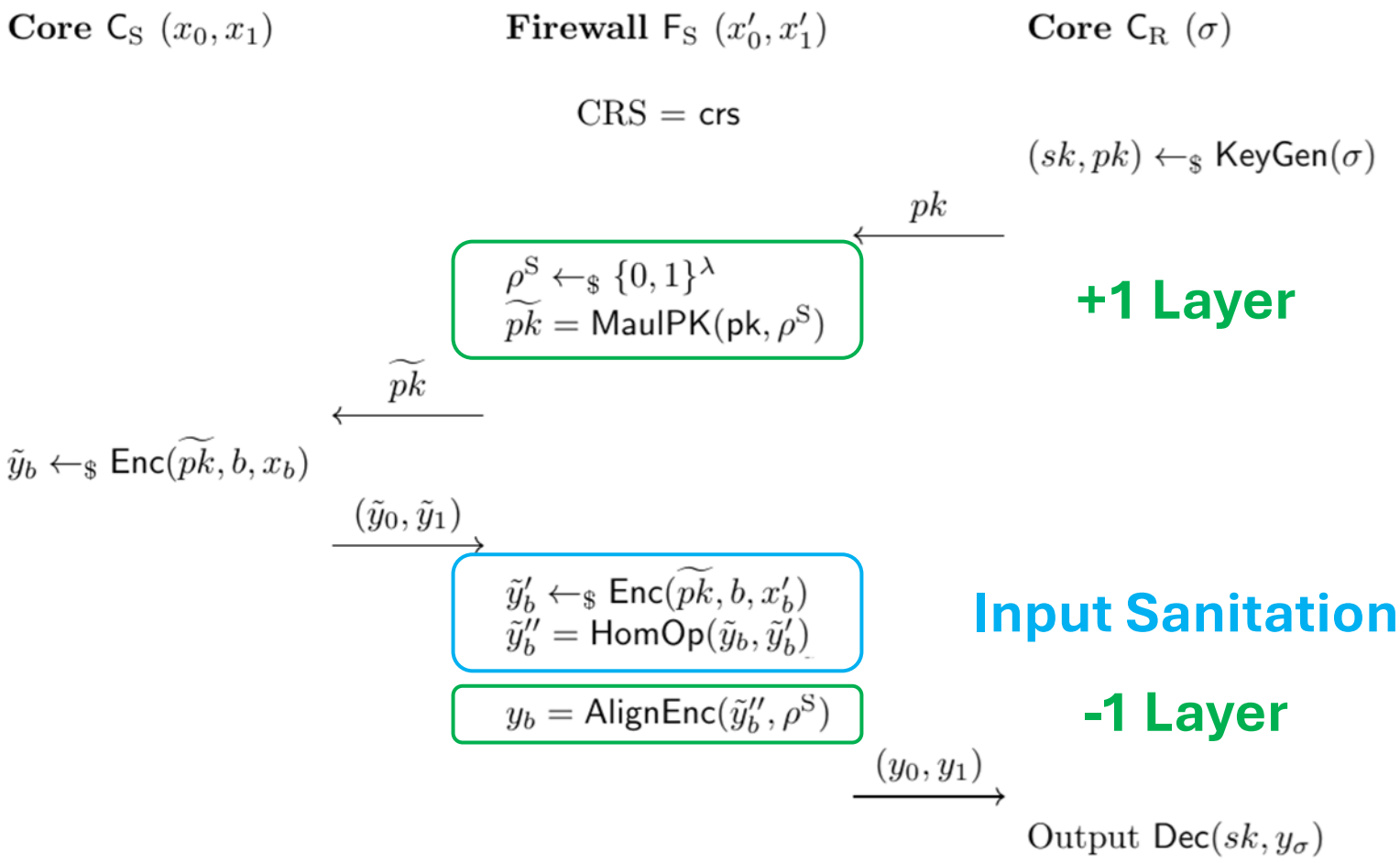
Instantiating $\mathcal{F}_{sOT}$



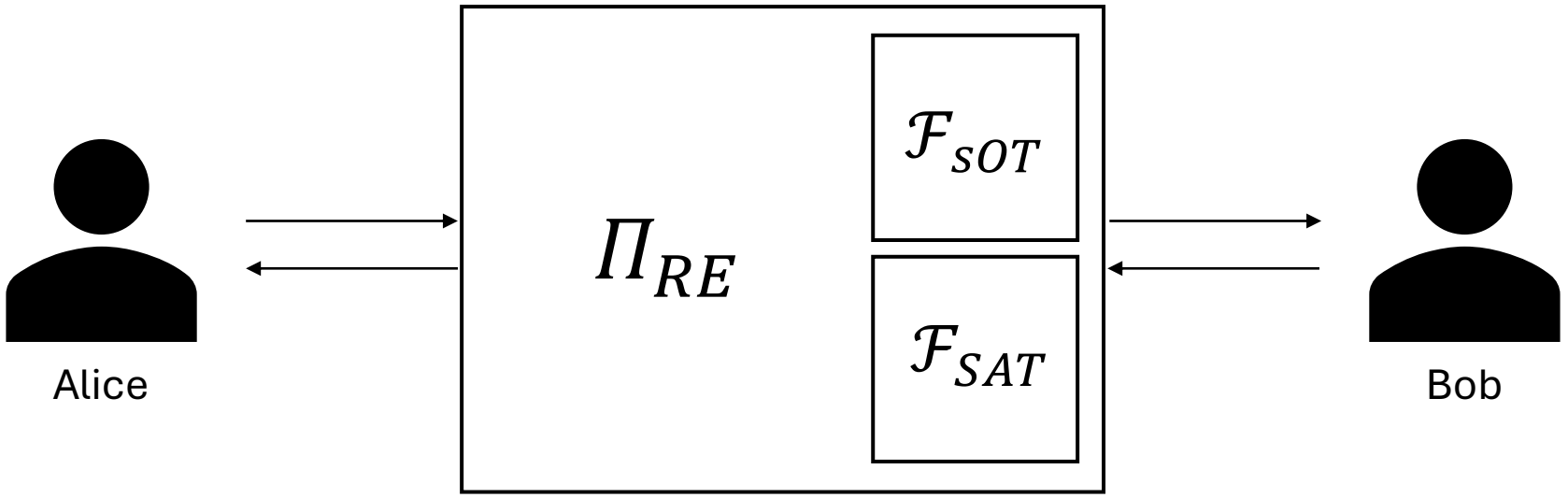
Instantiating $\mathcal{F}_{sOT}$



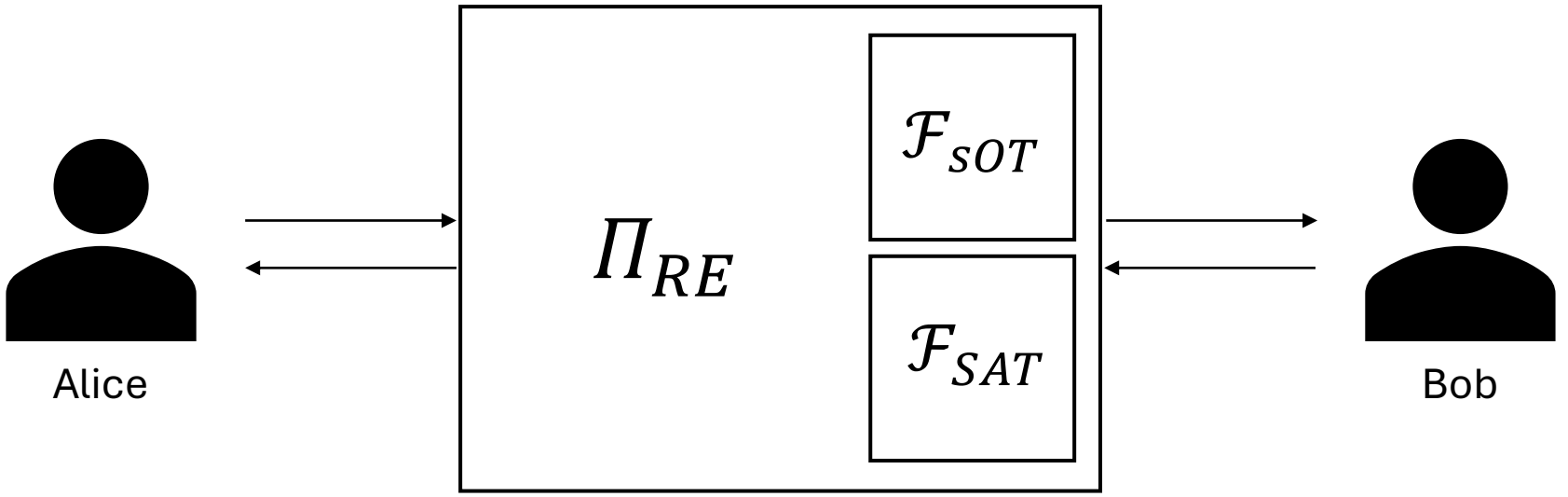
Instantiating $\mathcal{F}_{sOT}$



Randomized Equality from $\mathcal{F}_{sOT}$

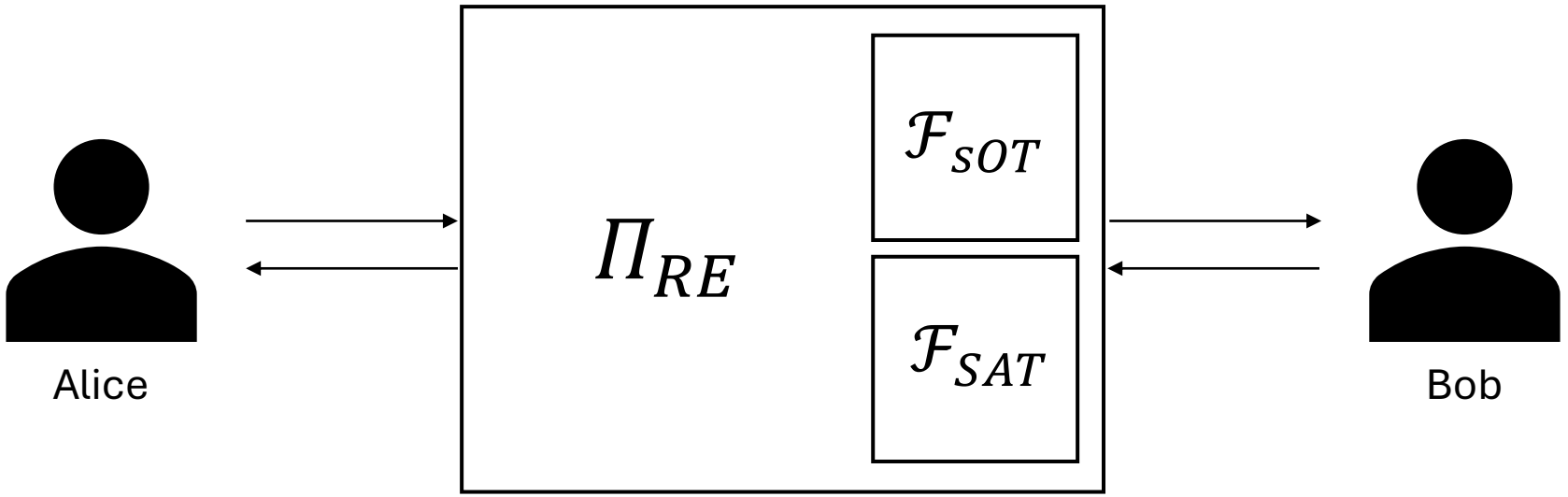


Randomized Equality from $\mathcal{F}_{sOT}$



- Given $\boldsymbol{\mathcal{F}_{sOT}}$, we realize Randomized Equality in the srUC framework.

Randomized Equality from $\mathcal{F}_{sOT}$

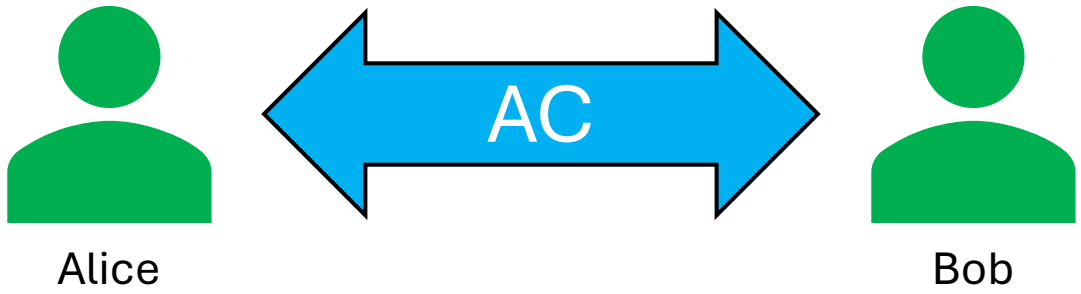


- Given $\mathcal{F}_{sOT}$, we realize Randomized Equality in the srUC framework.
- Similar protocol structure to **[CDVW12]** - details on the paper!

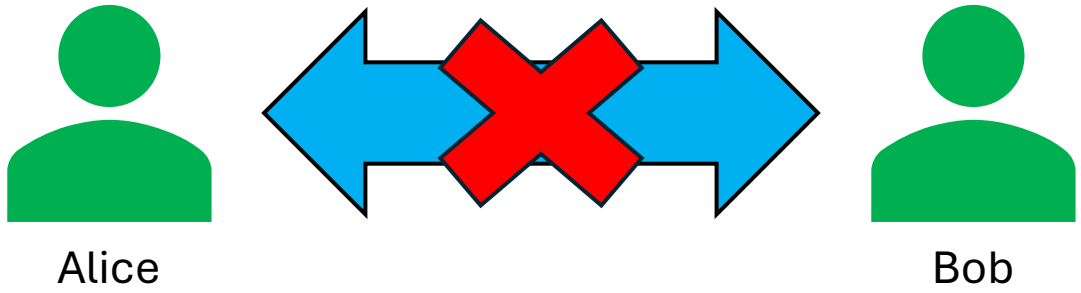
Bird's eye view

Intro to subversion	$\mathcal{F}_{RE}$ from srOT
$\mathcal{F}_{RE}$ to $\mathcal{F}_{PAKE}$	Wrapping up

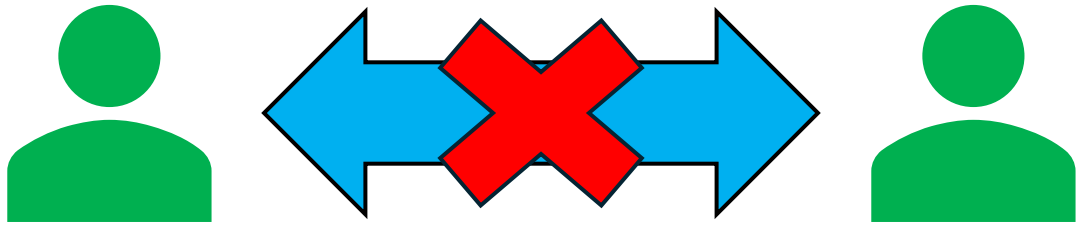
Removing authenticated channels in standard UC



Removing authenticated channels in standard UC

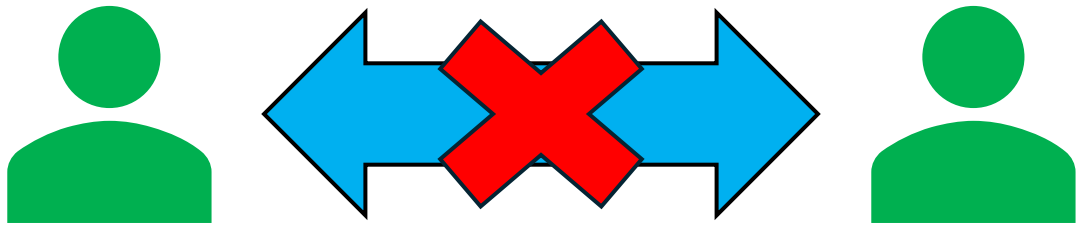


Removing authenticated channels in standard UC



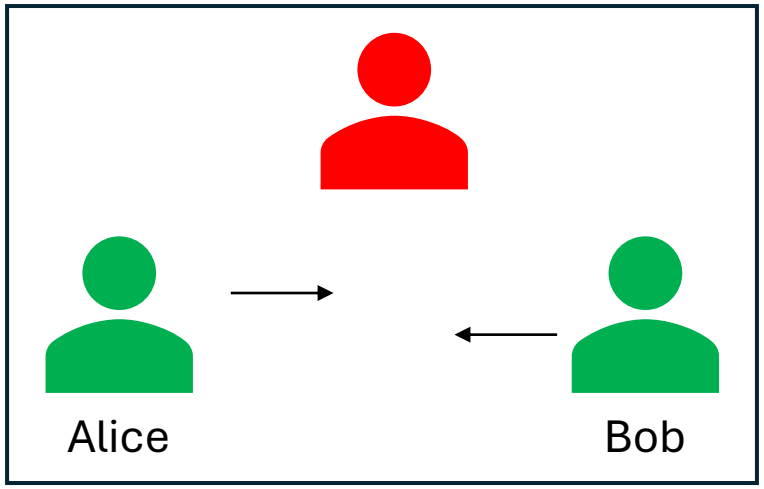
- **Idea**: leverage on **split** functionalities [**BCL+05**] (CRYPTO'05, JoC'07).

Removing authenticated channels in standard UC

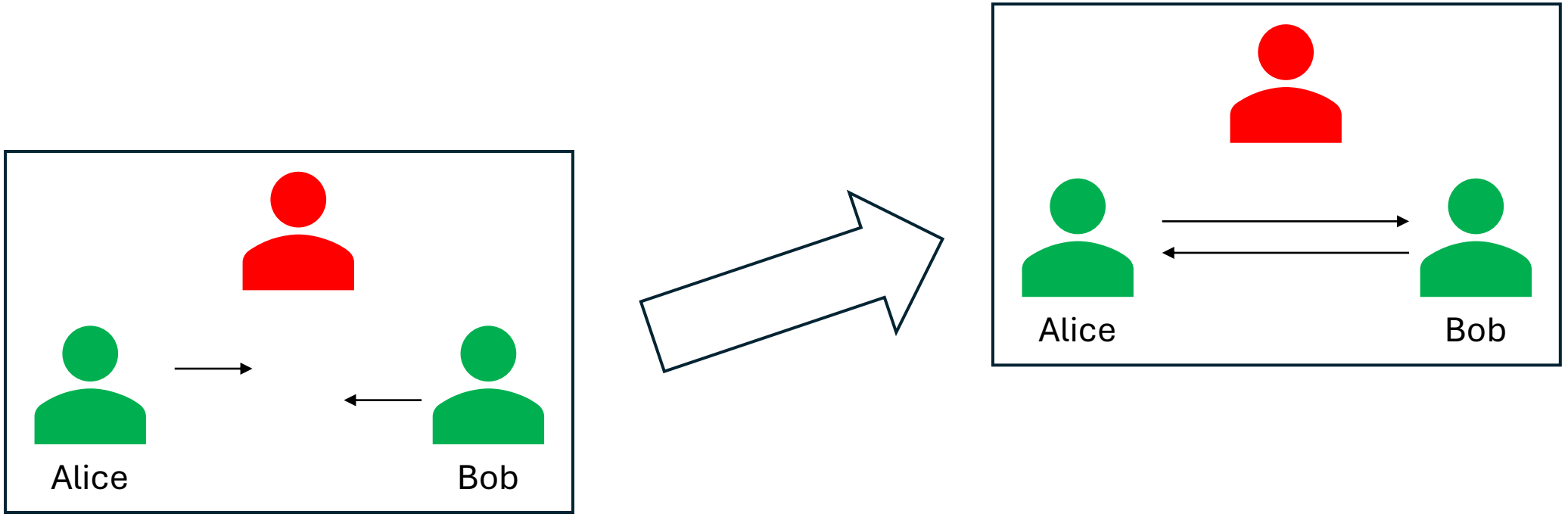


- **Idea**: leverage on **split** functionalities [**BCL+05**] (CRYPTO'05, JoC'07).
- The **worse** the adversary can do is partition parties before the protocol run.

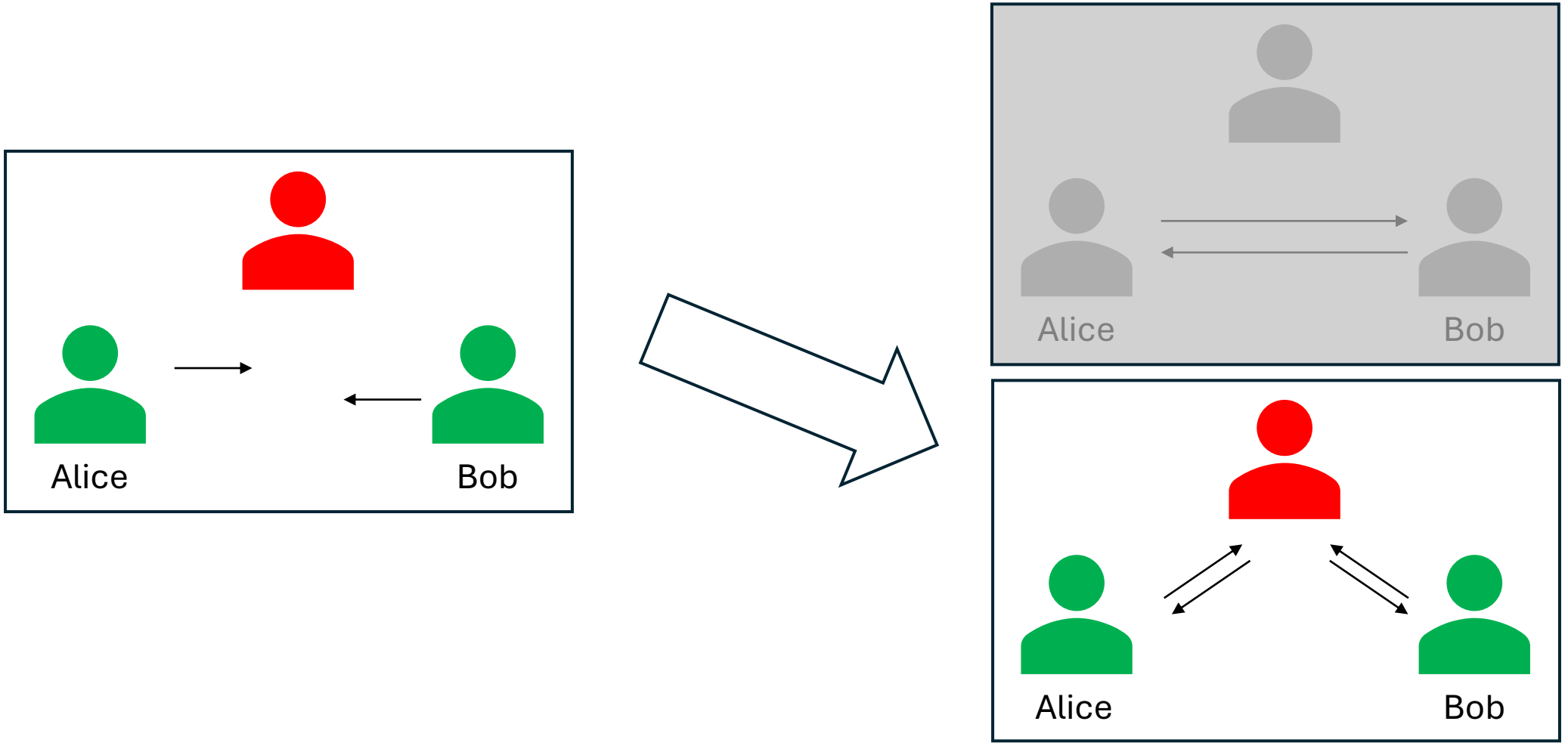
Split authentication in standard UC



Split authentication in standard UC



Split authentication in standard UC



Split authentication in standard UC [BCL+05]

Client P_i

$(\mathsf{sk}_i, \mathsf{vk}_i) \leftarrow_{\$} \mathsf{KeyGen}(1^\lambda)$

$\mathsf{vk}_i \longrightarrow$

$\longleftarrow \mathsf{vk}_j$

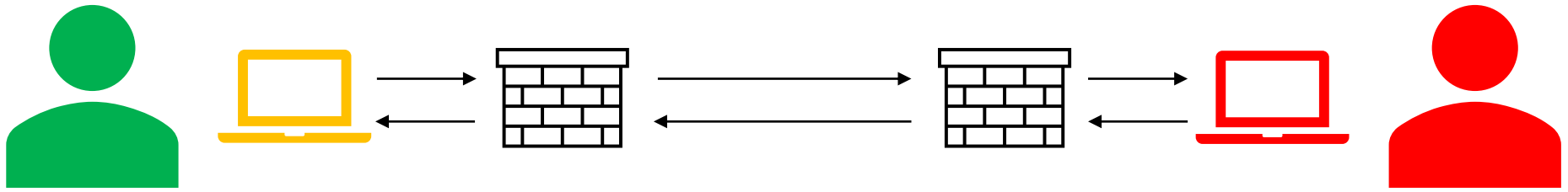
$\mathsf{sid}_i = (\mathsf{vk}_i, \mathsf{vk}_j)$
$\sigma_i = \mathsf{Sign}_{\mathsf{sk}_i}(\mathsf{sid}_i)$

$(\sigma_i, \mathsf{sid}_i) \longrightarrow$

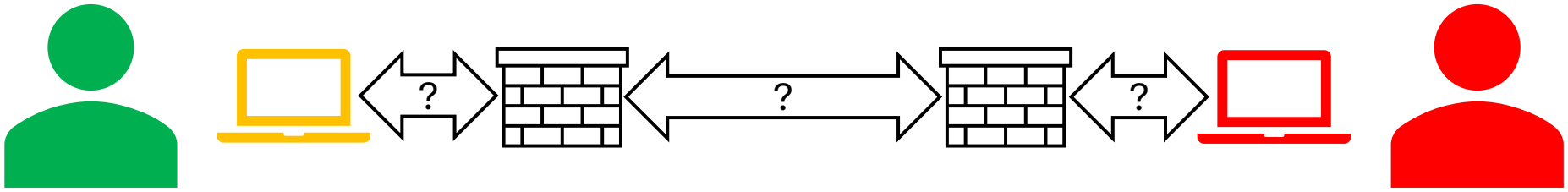
$\longleftarrow (\sigma_j, \mathsf{sid}_j)$

If $\mathsf{sid}_i \neq \mathsf{sid}_j$, abort.
If $\mathsf{Vrfy}_{\mathsf{vk}_j}(\mathsf{sid}_j, \sigma_j) = 0$, abort.

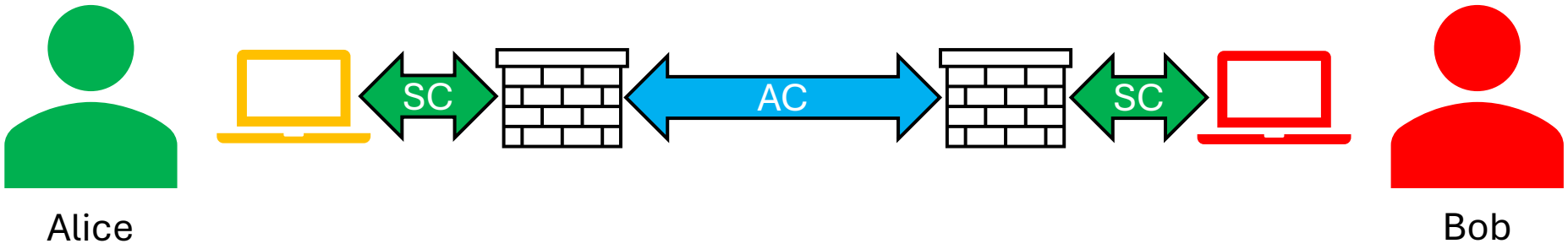
Backbone of communication in srUC



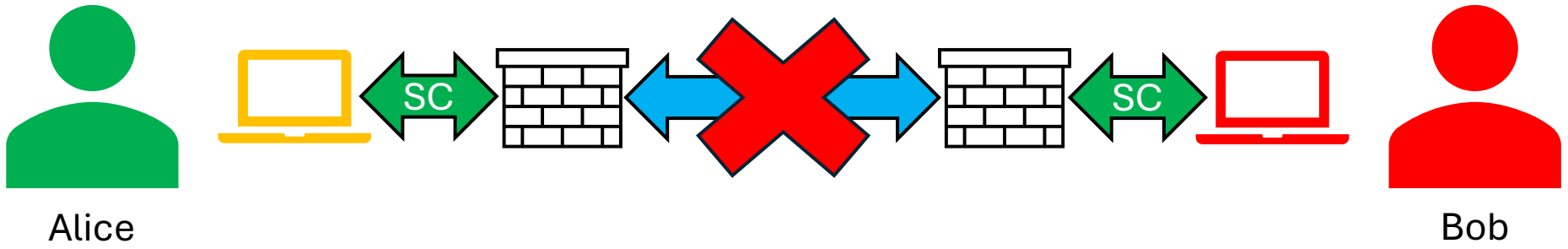
Backbone of communication in srUC



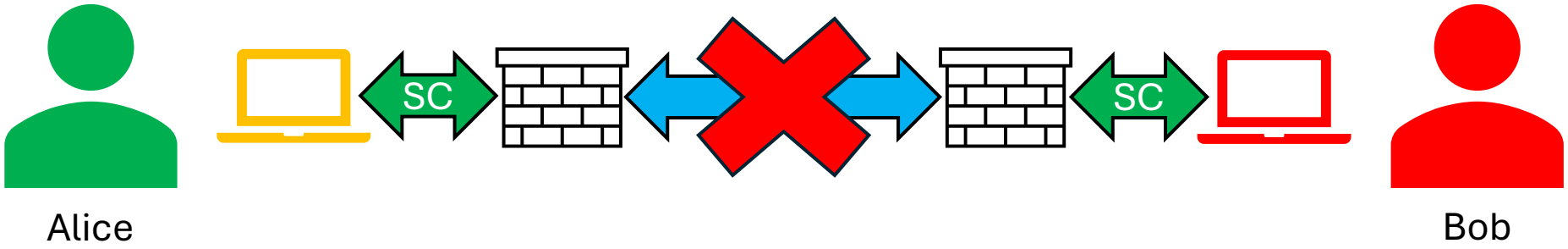
Backbone of communication in srUC ($\mathcal{F}_{SAT}$)



Backbone of communication in srUC (split $\mathcal{F}_{SAT}$)

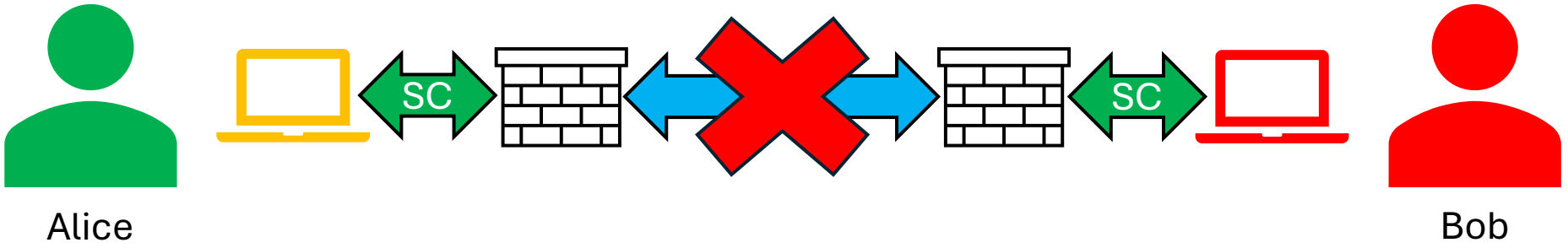


Backbone of communication in srUC (split $\mathcal{F}_{SAT}$)



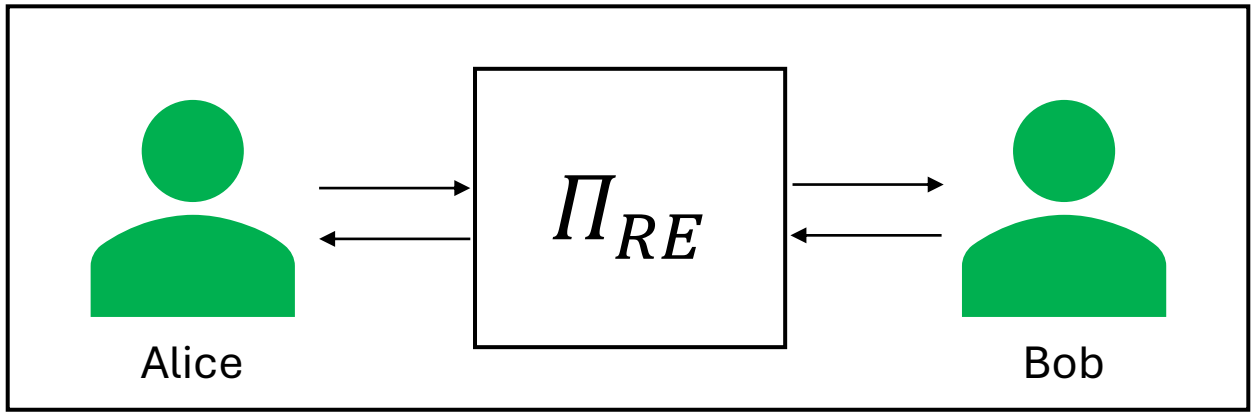
- Proper instantiation by sanitizing the protocol for split-auth shown before.

Backbone of communication in srUC (split $\mathcal{F}_{SAT}$)



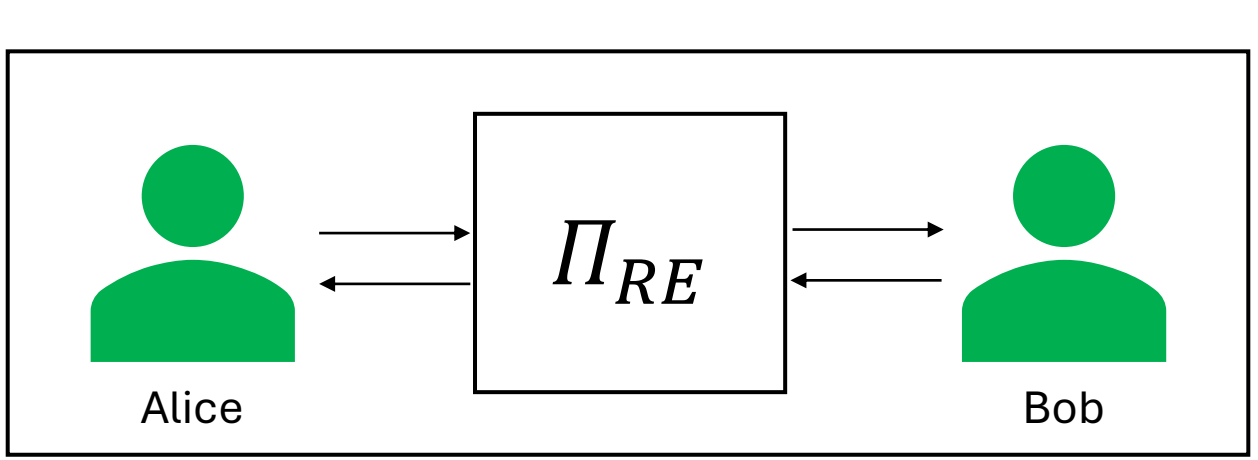
- Proper instantiation by sanitizing the protocol for split-auth shown before.
- Improving on an open problem: $\mathcal{F}_{SAT}$ was never instantiated in [**CMNV22**]!

From Π_{RE} to Π_{PAKE}



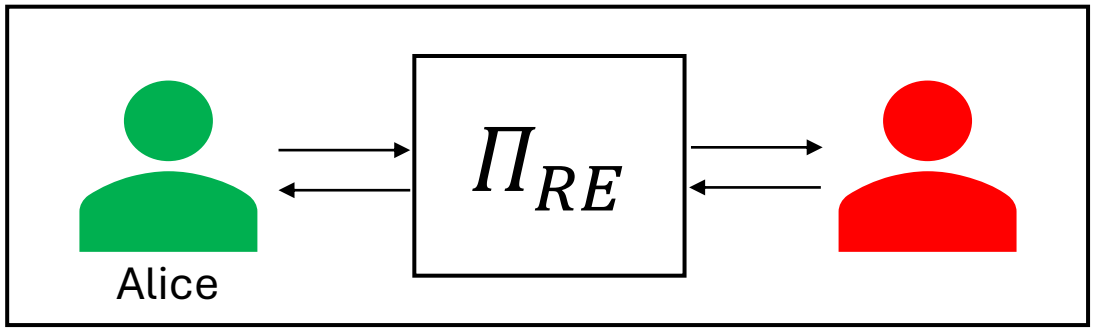
AUTH

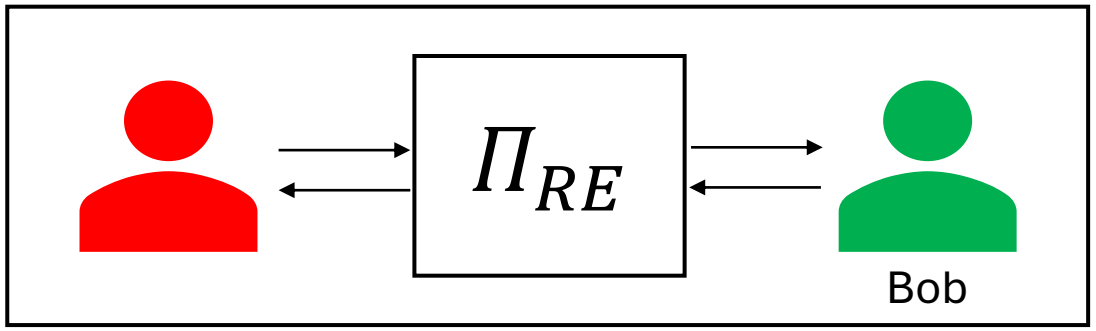
From Π_{RE} to Π_{PAKE}



AUTH

UNAUTH





Bird's eye view

Intro to subversion	$\mathcal{F}_{RE}$ from srOT
$\mathcal{F}_{RE}$ to $\mathcal{F}_{PAKE}$	**Wrapping up**

Wrapping up

- Introduction of the **first** PAKE protocol in the srUC framework.

Wrapping up

- Introduction of the **first** PAKE protocol in the srUC framework.
- Several new **sanitation-friendly** primitives.

Wrapping up

- Introduction of the **first** PAKE protocol in the srUC framework.
- Several new **sanitation-friendly** primitives.
- Extension of the srUC framework to the **unauthenticated** setting.

Wrapping up

- Introduction of the **first** PAKE protocol in the srUC framework.
- Several new **sanitation-friendly** primitives.
- Extension of the srUC framework to the **unauthenticated** setting.
- **Open**: more efficient PAKE, other functionalities, new primitives – the sky's the limit!

Thanks!