QuietOT: Lightweight Oblivious Transfer with a Public-Key Setup

INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

Sacha Servan-Schreiber

MIT CSAIL

Joint work with
Geoffroy Couteau, Lalita Devadas, Srinivas Devadas, and Alexander Koch

This talk: Fast oblivious transfer with QuietOT

Overview

This talk: Fast oblivious transfer with QuietOT

Overview

- **Background on oblivious transfer**

This talk: Fast oblivious transfer with QuietOT

Overview

- **Background on oblivious transfer**
- **Background on OT extension**

This talk: Fast oblivious transfer with QuietOT

Overview

- **Background on oblivious transfer**
- **Background on OT extension**
- **QuietOT framework**

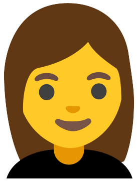
This talk: Fast oblivious transfer with QuietOT

Overview

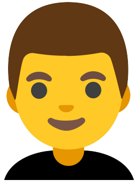
- Background on oblivious transfer
- Background on OT extension
- QuietOT framework
- Evaluation

Oblivious Transfer

Oblivious Transfer

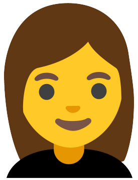


Alice



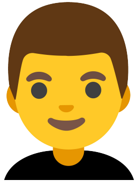
Bob

Oblivious Transfer



Alice

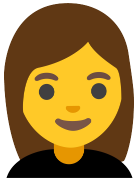
(m_0, m_1)



Bob

b

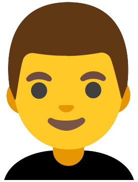
Oblivious Transfer



Alice

(m_0, m_1)

Wants: $\perp$



Bob

b

Wants: m_b

Oblivious Transfer

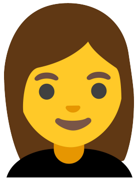
(m_0, m_1)

b

Wants: $\perp$

Wants: m_b

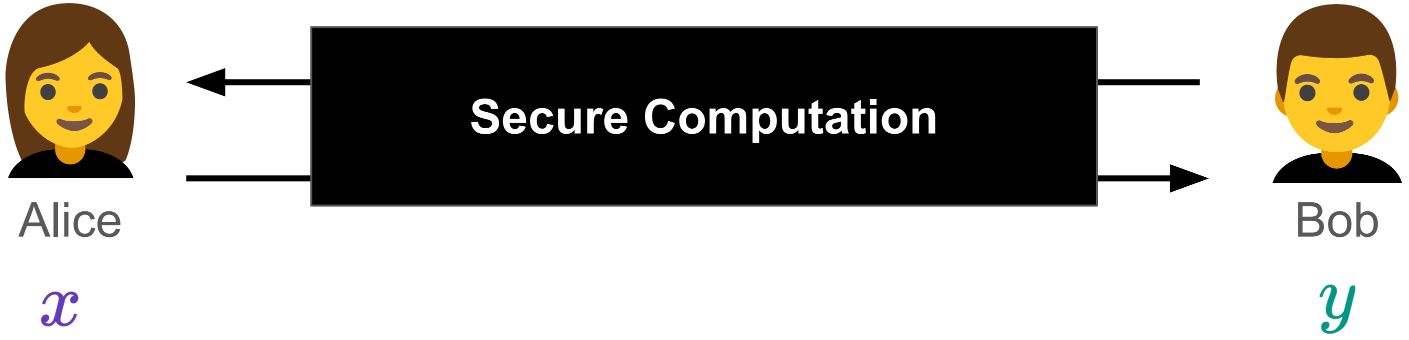
Oblivious Transfer for secure computation



Alice

Bob

Oblivious Transfer for secure computation

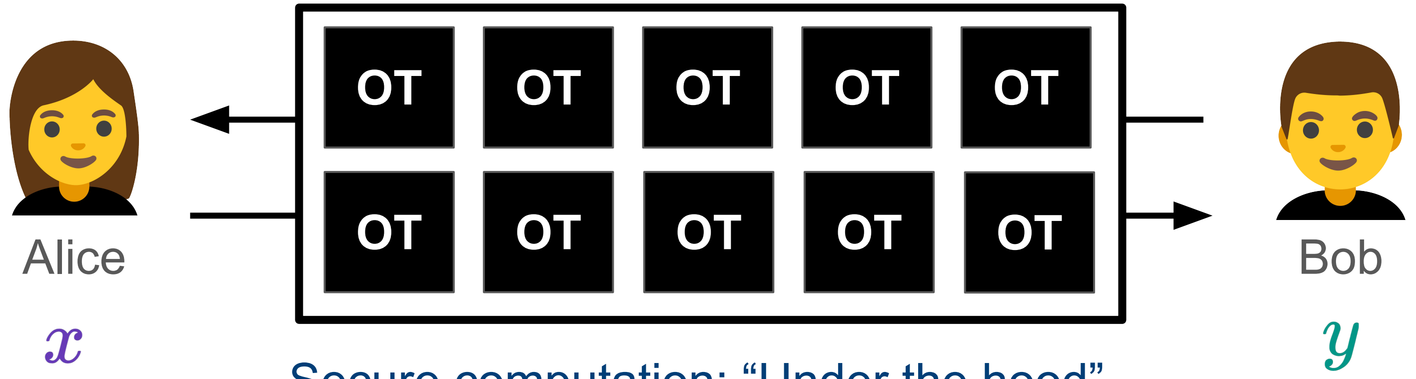


Oblivious Transfer for secure computation

Wants: $\perp$

Wants: $f(x, y)$

Oblivious Transfer for secure computation



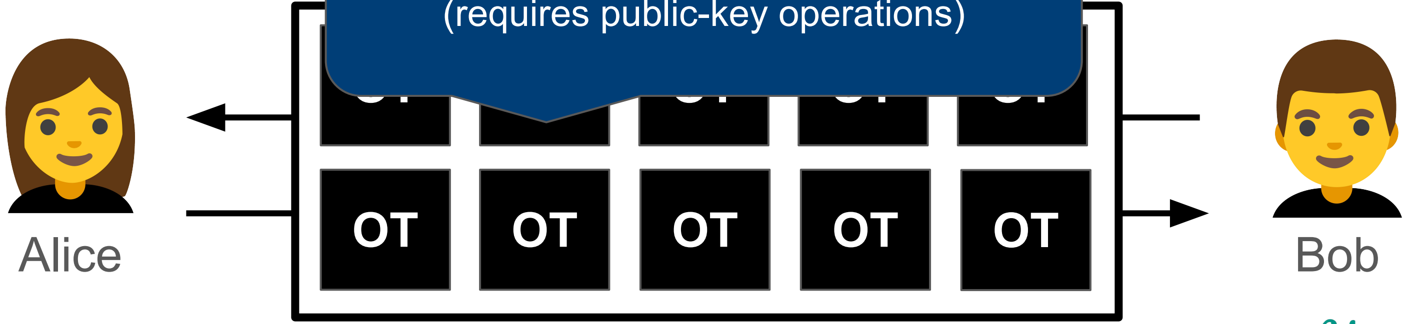
Secure computation: "Under the hood"

Wants: $\perp$

Wants: $f(x, y)$

Oblivious Transfer for secure computation

Each OT is computationally expensive
(requires public-key operations)



x

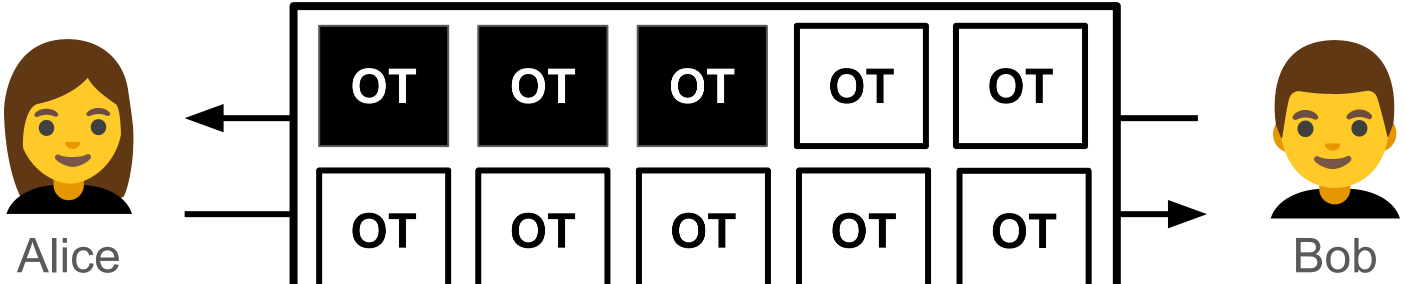
y

Secure computation: “Under the hood”

Wants: $\perp$

Wants: $f\left(x, y\right)$

Oblivious Transfer for secure computation



x

y

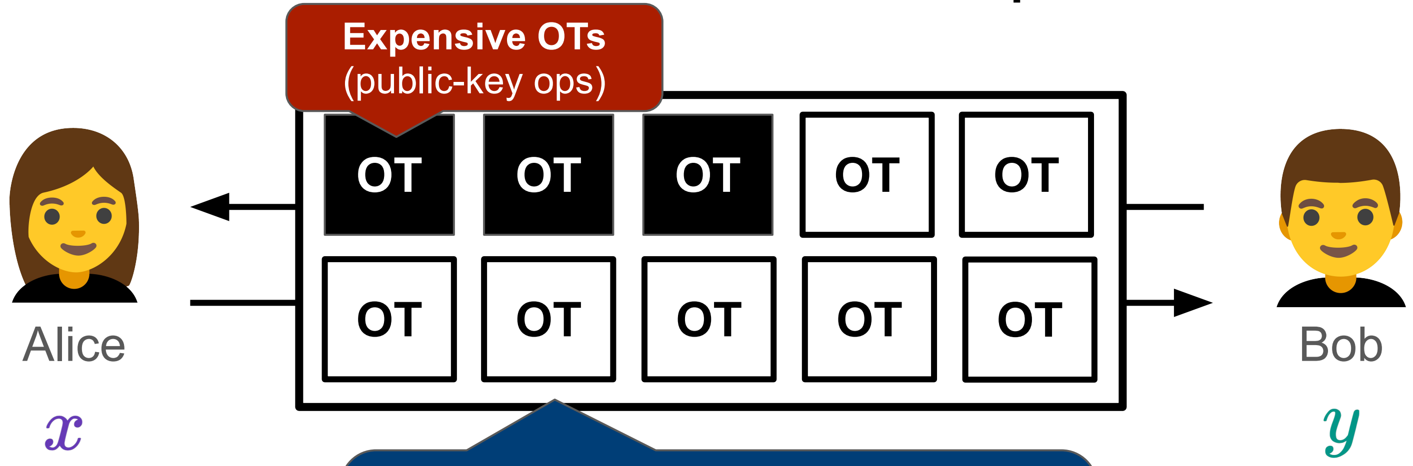
Idea: OT Extension [Beaver'96],[IKNP'03]

1. Compute a small number of OT
2. "Extend" them to many OTs

Wants: $\perp$

Wants: $f(x, y)$

Oblivious Transfer for secure computation



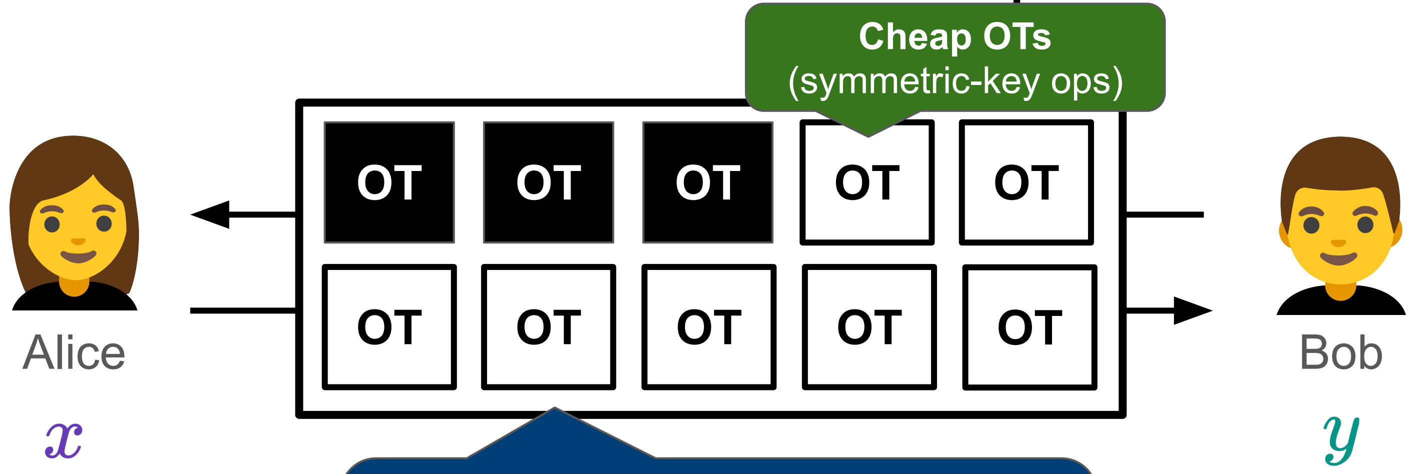
Idea: OT Extension [Beaver'96],[IKNP'03]

1. Compute a small number of OT
2. "Extend" them to many OTs

Wants: $\perp$

Wants: $f(x, y)$

Oblivious Transfer for secure computation



Idea: OT Extension [Beaver'96],[IKNP'03]

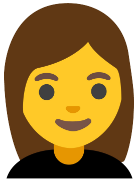
1. Compute a small number of OT
2. "Extend" them to many OTs

Wants: $\perp$

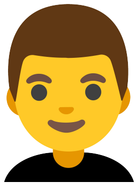
Wants: $f\left(x, y\right)$

OT Extension

OT Extension

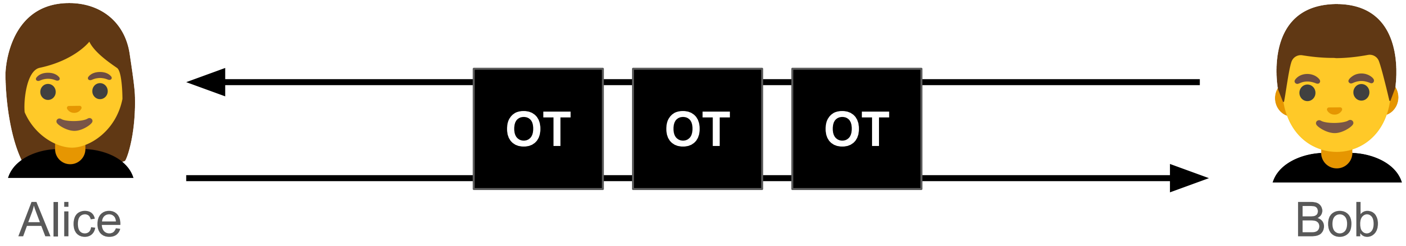


Alice



Bob

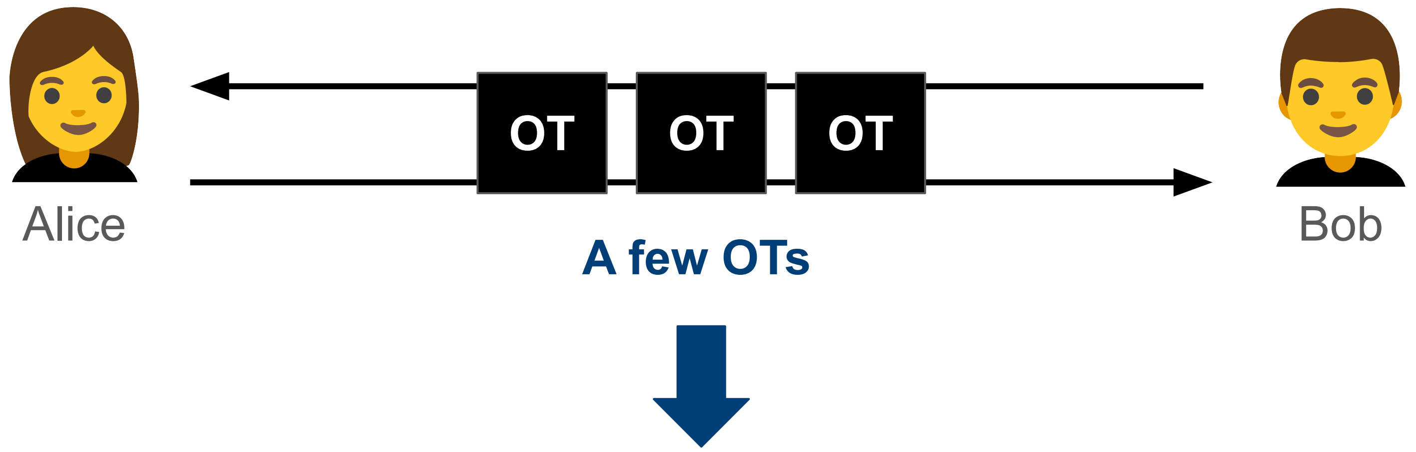
OT Extension



OT Extension

A few OTs

OT Extension



OT Extension

$(a_{1,0}, a_{1,1})$

(a_{1,b_1}, b_1)

A few OTs

Many *Pseudo*random OT Correlations

OT Extension

$(a_{1,0}, a_{1,1})$

$(a_{2,0}, a_{2,1})$

A few OTs

Many *Pseudo*random OT Correlations

(a_{1,b_1}, b_1)

(a_{2,b_2}, b_2)

OT Extension

A few OTs

Alice	Bob
$(a_{1,0}, a_{1,1})$	(a_{1,b_1}, b_1)
$(a_{2,0}, a_{2,1})$	(a_{2,b_2}, b_2)
$(a_{3,0}, a_{3,1})$	(a_{3,b_3}, b_3)

Many *Pseudo*random OT Correlations

OT Extension

A few OTs

⬇

Many *Pseudo*random OT Correlations

Alice	Bob
$(a_{1,0}, a_{1,1})$	(a_{1,b_1}, b_1)
$(a_{2,0}, a_{2,1})$	(a_{2,b_2}, b_2)
$(a_{3,0}, a_{3,1})$	(a_{3,b_3}, b_3)
$(a_{4,0}, a_{4,1})$	(a_{4,b_4}, b_4)

OT Extension

A few OTs

⬇

Many *Pseudo*random OT Correlations

Alice:

$(a_{1,0}, a_{1,1})$

$(a_{2,0}, a_{2,1})$

$(a_{3,0}, a_{3,1})$

$(a_{4,0}, a_{4,1})$

$\vdots$

Bob:

(a_{1,b_1}, b_1)

(a_{2,b_2}, b_2)

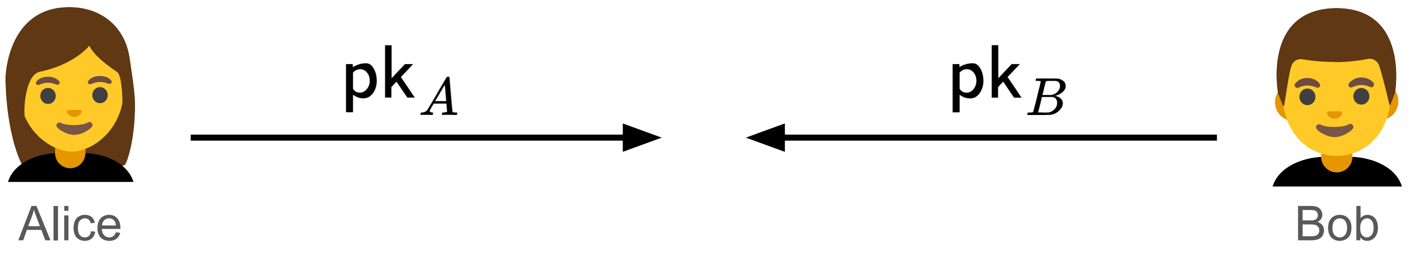
(a_{3,b_3}, b_3)

(a_{4,b_4}, b_4)

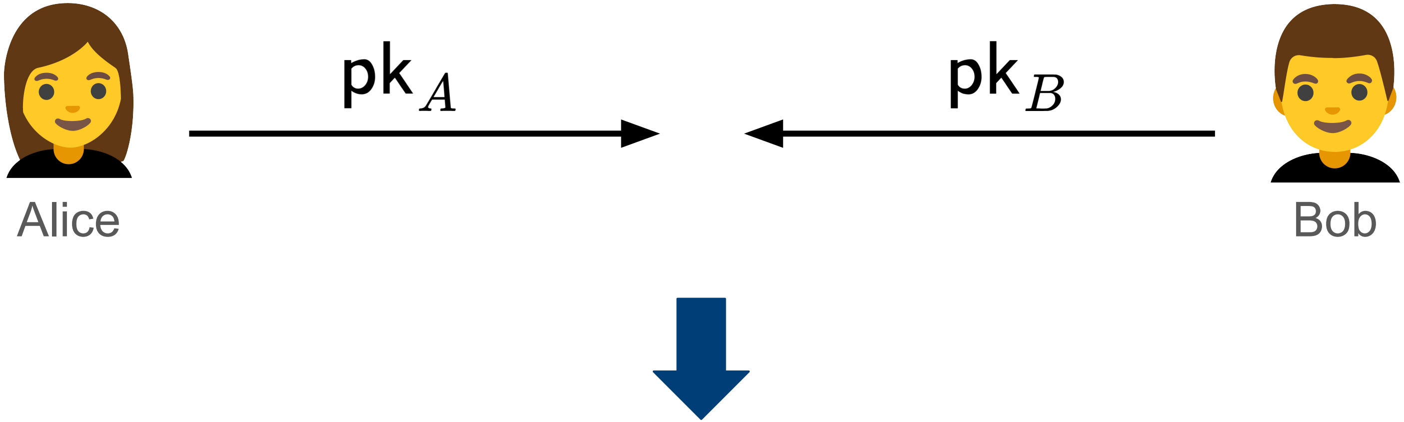
$\vdots$

OT Extension **with a public-key setup**

OT Extension with a public-key setup

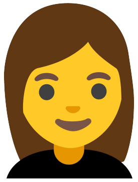


OT Extension **with a public-key setup**



Many *Pseudo*random
OT Correlations

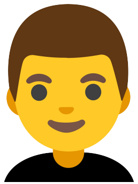
OT Extension with a public-key setup



Alice

$\mathsf{pk}_A \longrightarrow$

$\longleftarrow \mathsf{pk}_B$



Bob

$(a_{1,0}, a_{1,1})$

$(a_{2,0}, a_{2,1})$

$(a_{3,0}, a_{3,1})$

$(a_{4,0}, a_{4,1})$

$\vdots$

Many *Pseudo*random OT Correlations

(a_{1,b_1}, b_1)

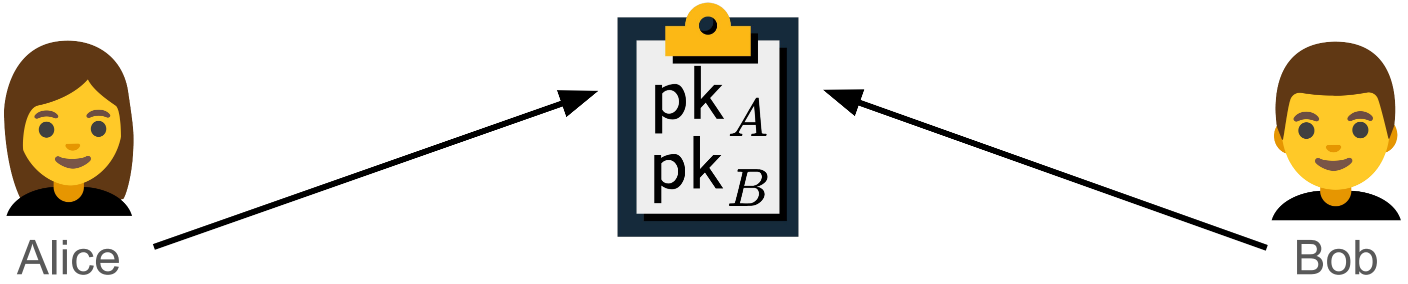
(a_{2,b_2}, b_2)

(a_{3,b_3}, b_3)

(a_{4,b_4}, b_4)

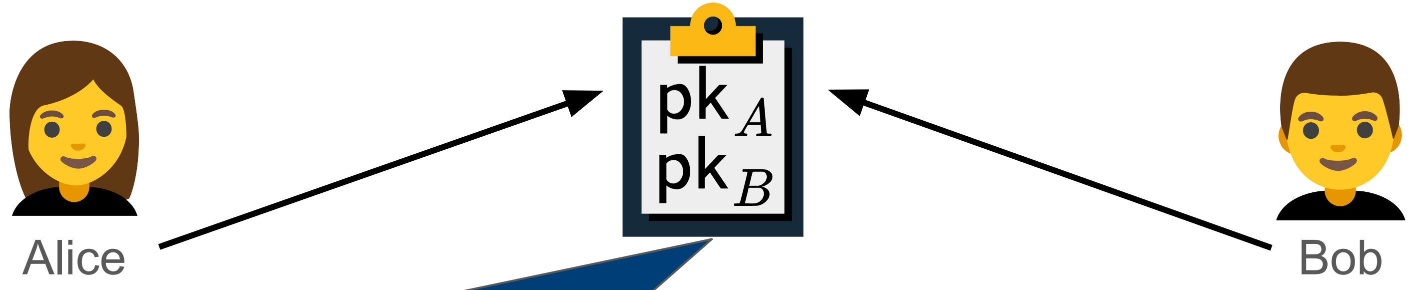
$\vdots$

OT Extension **with a public-key setup**



Alice		Bob
$(a_{1,0}, a_{1,1})$		(a_{1,b_1}, b_1)
$(a_{2,0}, a_{2,1})$		(a_{2,b_2}, b_2)
$(a_{3,0}, a_{3,1})$	**Many *Pseudo*random OT Correlations**	(a_{3,b_3}, b_3)
$(a_{4,0}, a_{4,1})$		(a_{4,b_4}, b_4)
$\vdots$		$\vdots$

OT Extension **with a public-key setup**



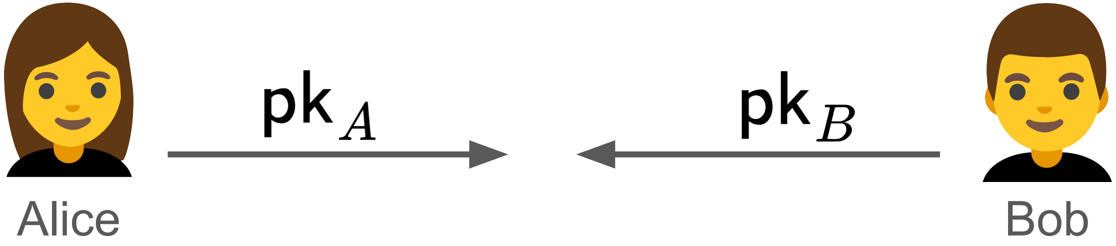
"Diffie–Hellman" for OT extension

Alice	Bob
$(a_{1,0}, a_{1,1})$	(a_{1,b_1}, b_1)
$(a_{2,0}, a_{2,1})$	(a_{2,b_2}, b_2)
$(a_{3,0}, a_{3,1})$	(a_{3,b_3}, b_3)
$(a_{4,0}, a_{4,1})$	(a_{4,b_4}, b_4)
$\vdots$	$\vdots$

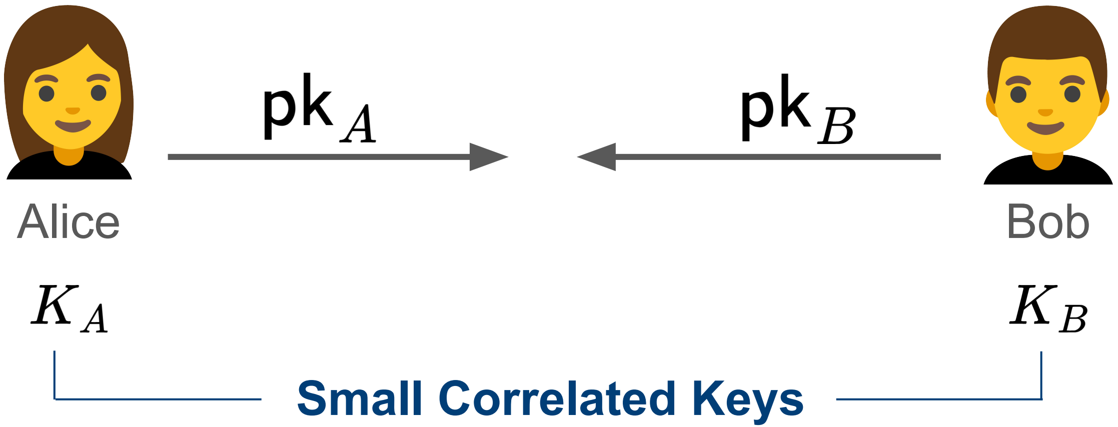
Many *Pseudo*random OT Correlations

QuietOT Framework

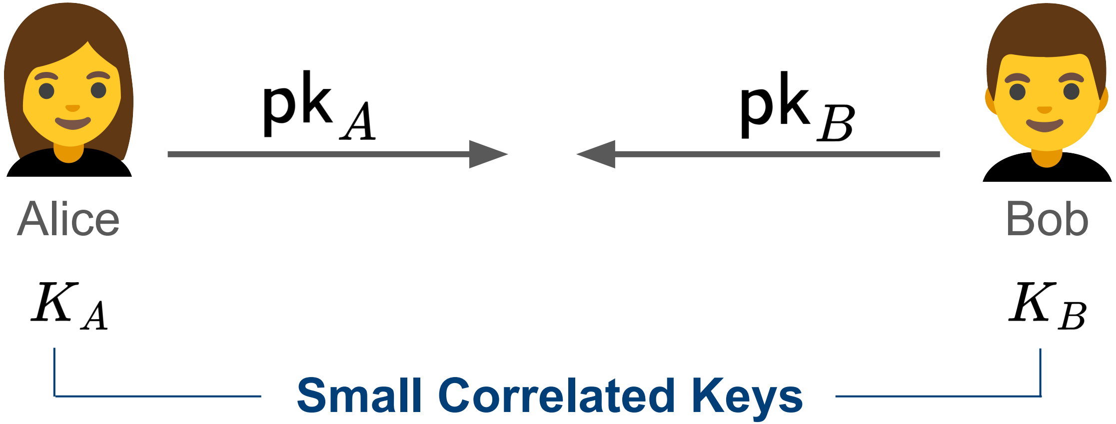
QuietOT: Communication model and syntax



QuietOT: Communication model and syntax



QuietOT: Communication model and syntax



$$((a_{1,0}, a_{1,1}), \ldots, (a_{n,0}, a_{n,1})) \leftarrow \mathsf{Extend}_A(K_A)$$

$$((a_{1,b_1}, b_1), \ldots (a_{n,b_n}, b_n)) \leftarrow \mathsf{Extend}_B(K_B)$$

Locally generate many pseudorandom "OTs"

QuietOT: The main ingredients

QuietOT: The main ingredients

CPRF for Inner-Products

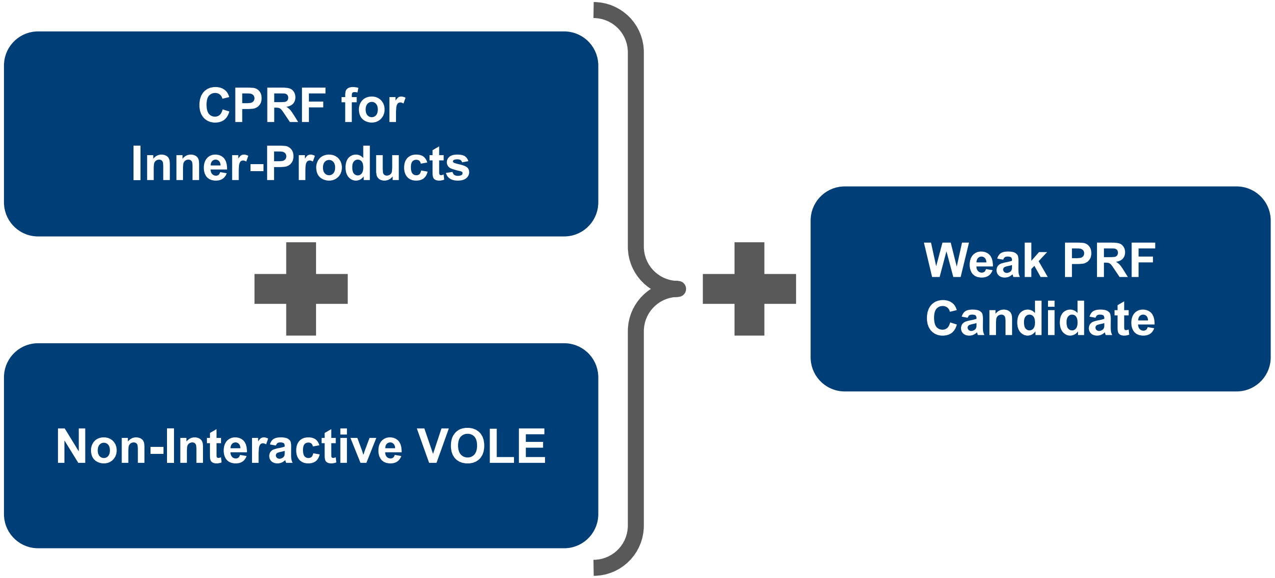
QuietOT: The main ingredients

CPRF for Inner-Products

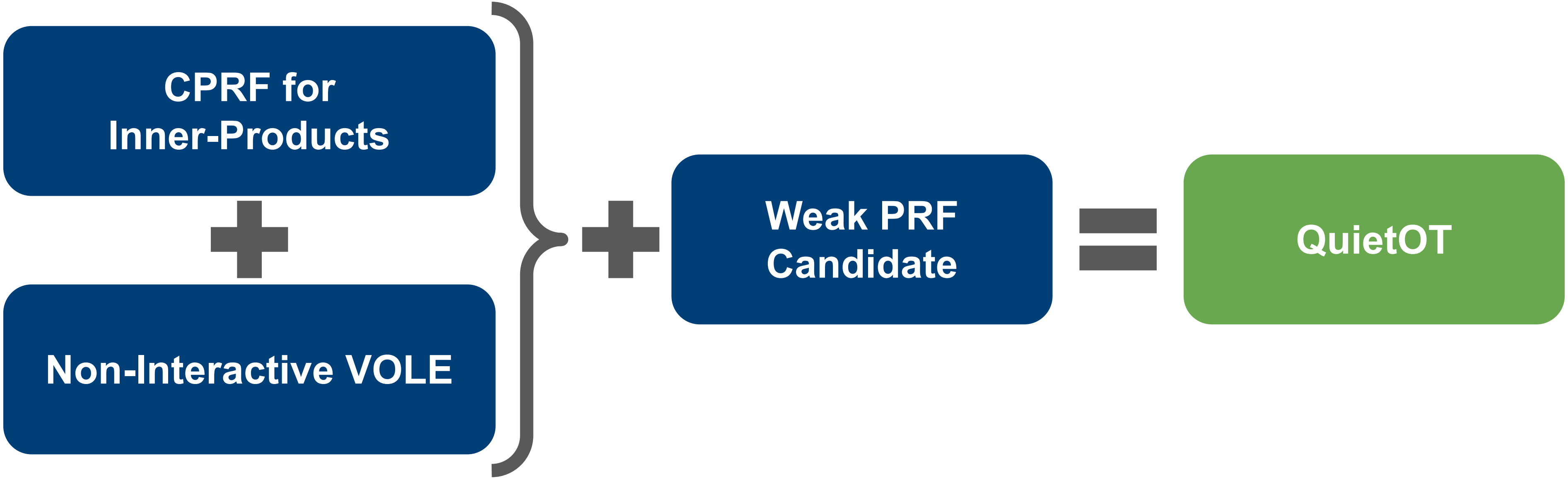
+

Non-Interactive VOLE

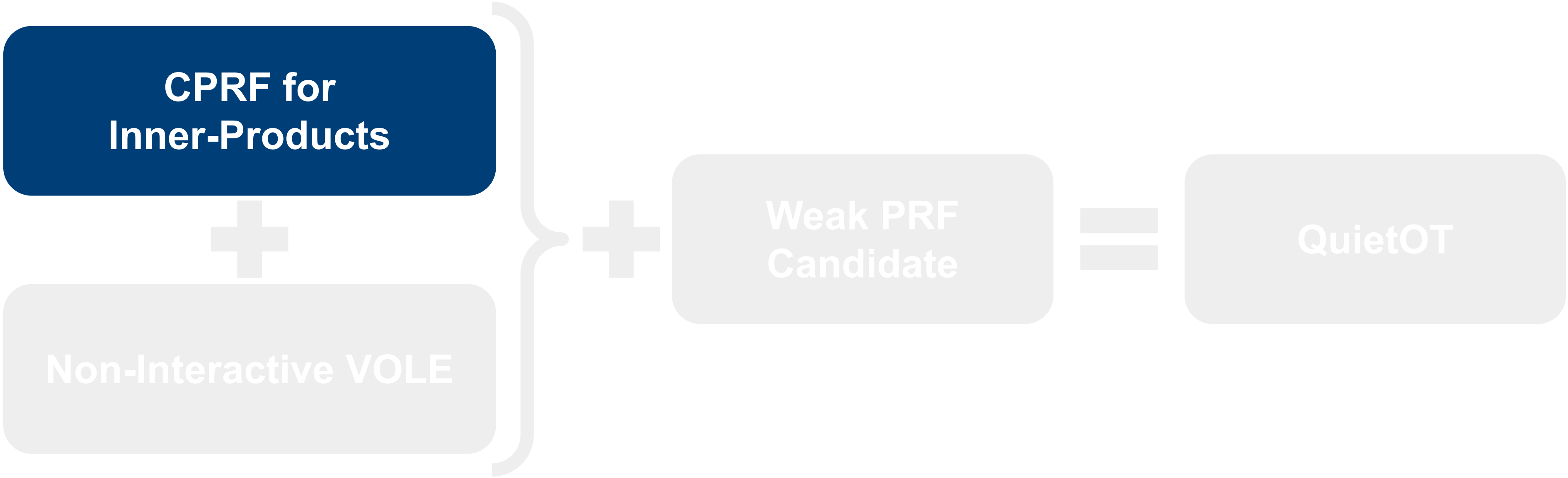
QuietOT: The main ingredients



QuietOT: The main ingredients



QuietOT: The main ingredients



Ingredient I: CPRF for Inner-Products

Constrained Pseudorandom Function (CPRF) [BW'13][KPTZ'13][BGI'14]

CPRFs have an additional **constrain** functionality for a class of circuits C:

Ingredient I: CPRF for Inner-Products

Constrained Pseudorandom Function (CPRF) [BW'13][KPTZ'13][BGI'14]

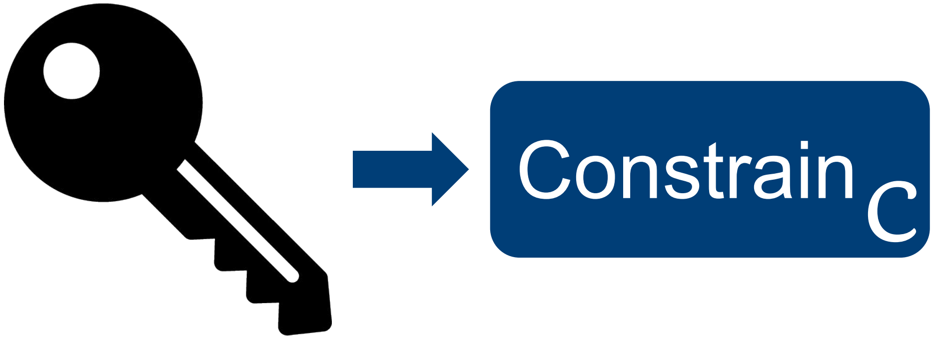
CPRFs have an additional **constrain** functionality for a class of circuits C:

Master PRF Key

Ingredient I: CPRF for Inner-Products

Constrained Pseudorandom Function (CPRF) [BW'13][KPTZ'13][BGI'14]

CPRFs have an additional **constrain** functionality for a class of circuits C:



Master PRF Key

Ingredient I: CPRF for Inner-Products

Constrained Pseudorandom Function (CPRF) [BW’13][KPTZ’13][BGI’14]

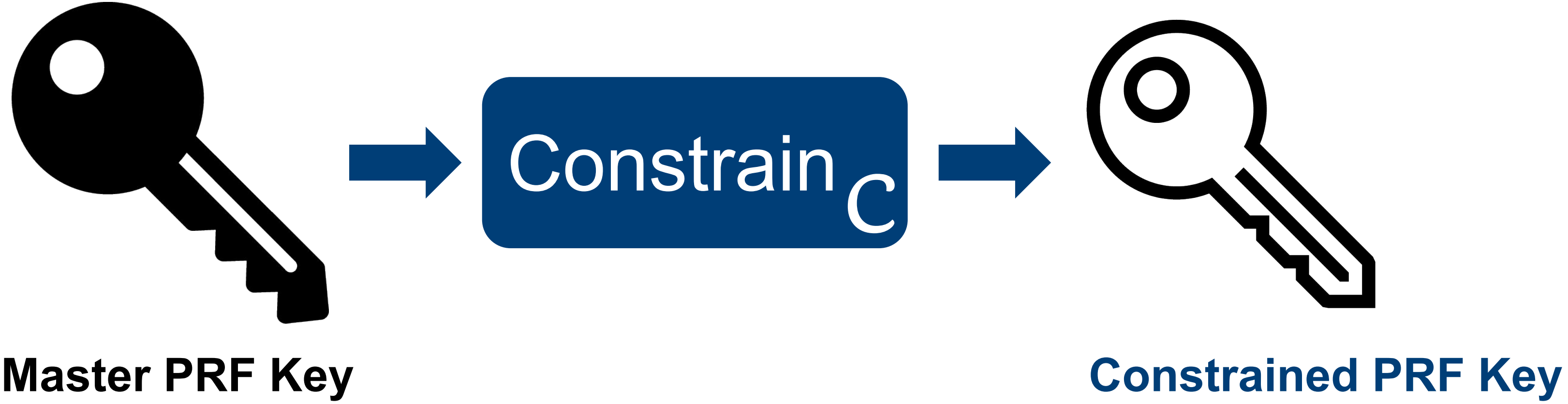
CPRFs have an additional **constrain** functionality for a class of circuits C:

Master PRF Key **Constrained PRF Key**

Ingredient I: CPRF for Inner-Products

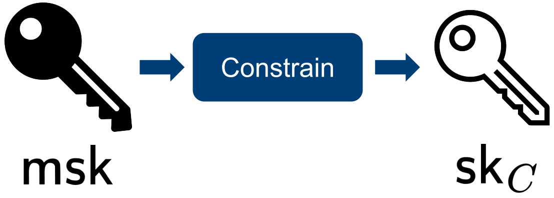
Constrained Pseudorandom Function (CPRF) [BW'13][KPTZ'13][BGI'14]

CPRFs have an additional **constrain** functionality for a class of circuits C:

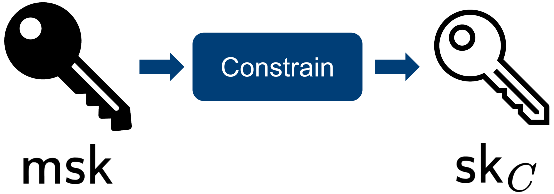


Constrained key can be used to evaluate PRF$(\;, x)$ for all $x \in \mathcal{X}$ where $C(x) = 0$

Ingredient I: CPRF for Inner-Products

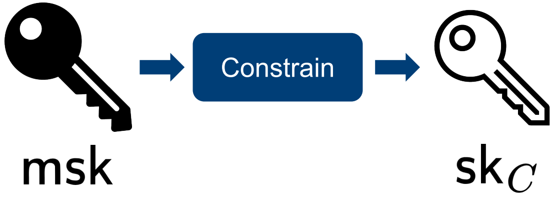


Ingredient I: CPRF for Inner-Products



$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

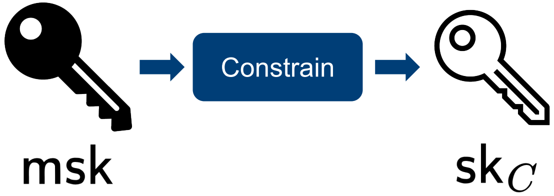
Ingredient I: CPRF for Inner-Products



$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

Correctness: If $C(x) = 0$ then $\mathsf{PRF}(\mathsf{msk}, x) = \mathsf{PRF}(\mathsf{sk}_C, x)$

Ingredient I: CPRF for Inner-Products

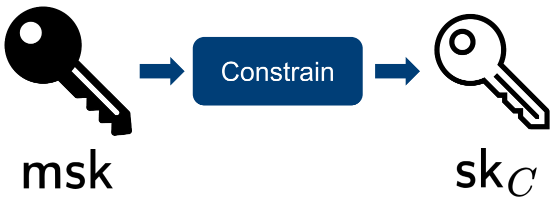


$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

Correctness: If $C(x) = 0$ then $\mathsf{PRF}(\mathsf{msk}, x) = \mathsf{PRF}(\mathsf{sk}_C, x)$

Pseudorandomness: If $C(x) \neq 0$ then $\mathsf{PRF}(\mathsf{msk}, x)$ is pseudorandom given sk_C

Ingredient I: CPRF for Inner-Products



$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

Correctness: If $C(x) = 0$ then $\mathsf{PRF}(\mathsf{msk}, x) = \mathsf{PRF}(\mathsf{sk}_C, x)$

Pseudorandomness: If $C(x) \neq 0$ then $\mathsf{PRF}(\mathsf{msk}, x)$ is pseudorandom given sk_C

No Hiding: C is **not** hidden given sk_C

Ingredient I: CPRF for Inner-Products

For efficiency: Use CPRF with inner-product predicate

Ingredient I: CPRF for Inner-Products

For efficiency: Use CPRF with inner-product predicate

$$C(\mathbf{x}) = \langle \mathbf{z}, \mathbf{x} \rangle \in \mathcal{R} \ \text{where}\, \mathbf{z}, \mathbf{x} \in \mathcal{R}^{\ell}$$

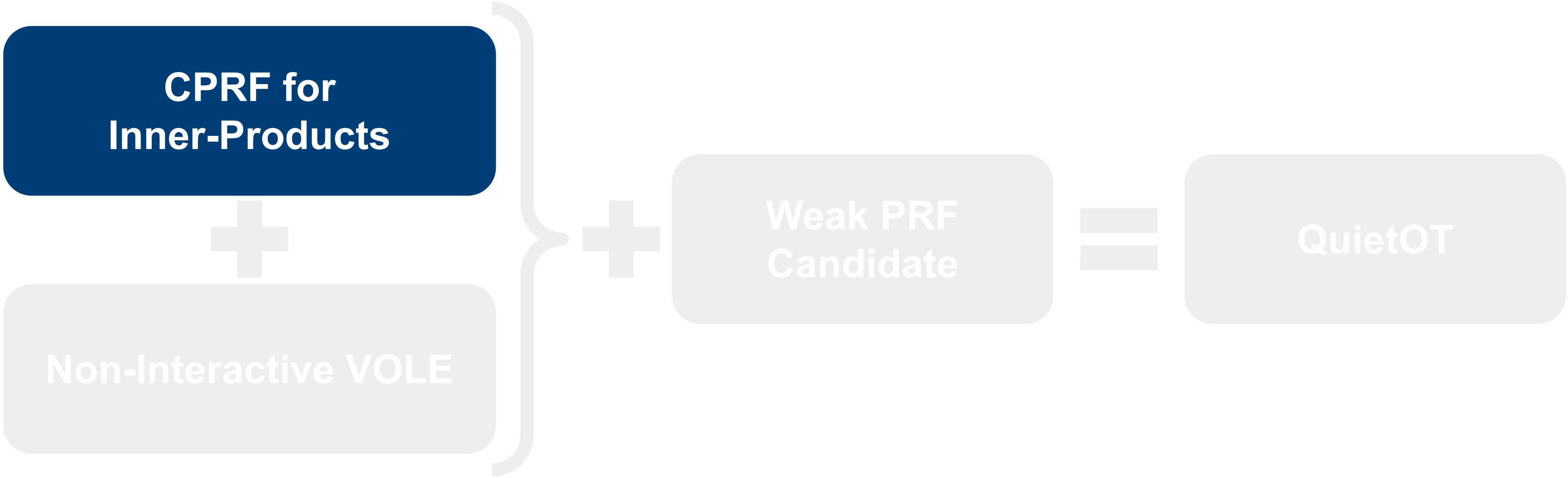
Ingredient I: CPRF for Inner-Products

For efficiency: Use CPRF with inner-product predicate

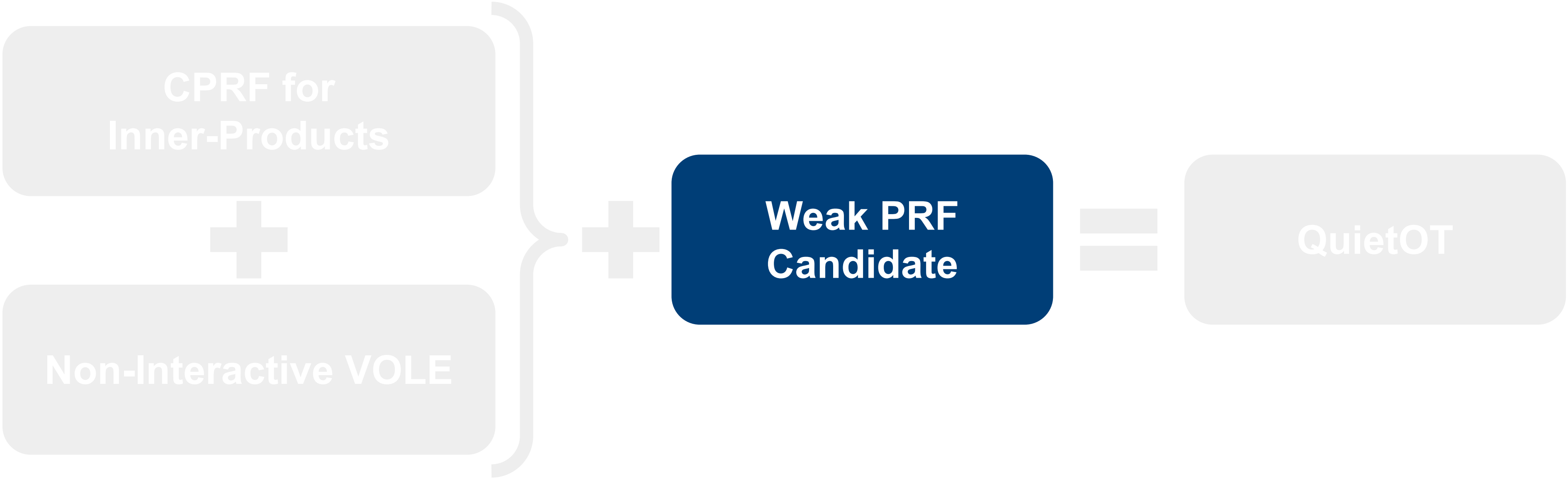
$$C(\mathbf{x}) = \langle \mathbf{z}, \mathbf{x} \rangle \in \mathcal{R} \text{ where } \mathbf{z}, \mathbf{x} \in \mathcal{R}^{\ell}$$

We use the random-oracle based CPRF construction of [S'24]

QuietOT: The main ingredients



QuietOT: The main ingredients



Ingredient II: "Inner-Product Membership" PRF

BIPSW weak PRF candidate: [BIPSW'18]

Ingredient II: "Inner-Product Membership" PRF

BIPSW weak PRF candidate: [BIPSW'18]

$$f_{\mathbf{k}}(\mathbf{x}) := \lceil \langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 \rfloor_2$$

Ingredient II: "Inner-Product Membership" PRF

BIPSW weak PRF candidate: [BIPSW'18]

$$f_{\mathbf{k}}(\mathbf{x}) := \lceil \underbrace{\langle \mathbf{k}, \mathbf{x} \rangle \bmod 6}_{\text{Just an inner product}} \rfloor_2$$

Ingredient II: "Inner-Product Membership" PRF

BIPSW weak PRF candidate: [BIPSW'18]

$$f_{\mathbf{k}}(\mathbf{x}) := \underbrace{\lceil \langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 \rfloor_2}_{\text{Cannot be evaluated as an inner product}}$$

Ingredient II: “Inner-Product Membership” PRF

BIPSW weak PRF candidate: [BIPSW’18]

$$f_{\mathbf{k}}(\mathbf{x}) := \lceil \langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 \rfloor_2$$

Ingredient II: "Inner-Product Membership" PRF

BIPSW weak PRF candidate: [BIPSW'18]

$$f_{\mathbf{k}}(\mathbf{x}) := \lceil \langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 \rfloor_2$$

$$f_{\mathbf{k}}(\mathbf{x}) = 0 \iff \langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 \in \{0, 1, 2\}$$

Ingredient II: “Inner-Product Membership” PRF

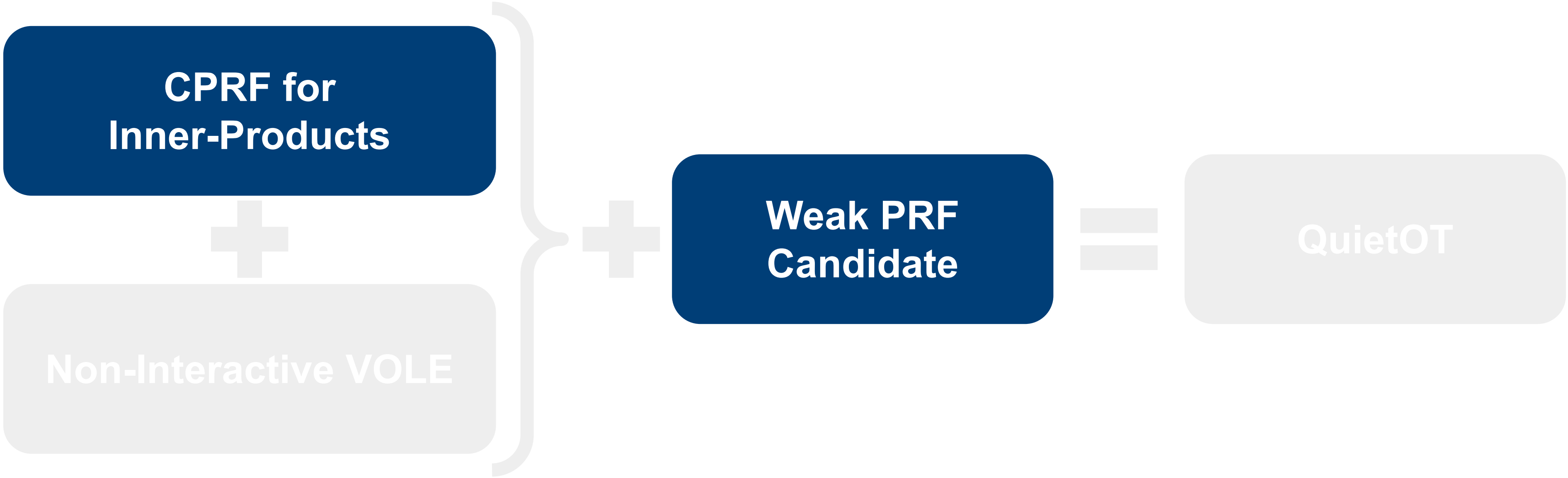
BIPSW weak PRF candidate: [BIPSW’18]

$$f_{\mathbf{k}}(\mathbf{x}) := \lceil \langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 \rfloor_2$$

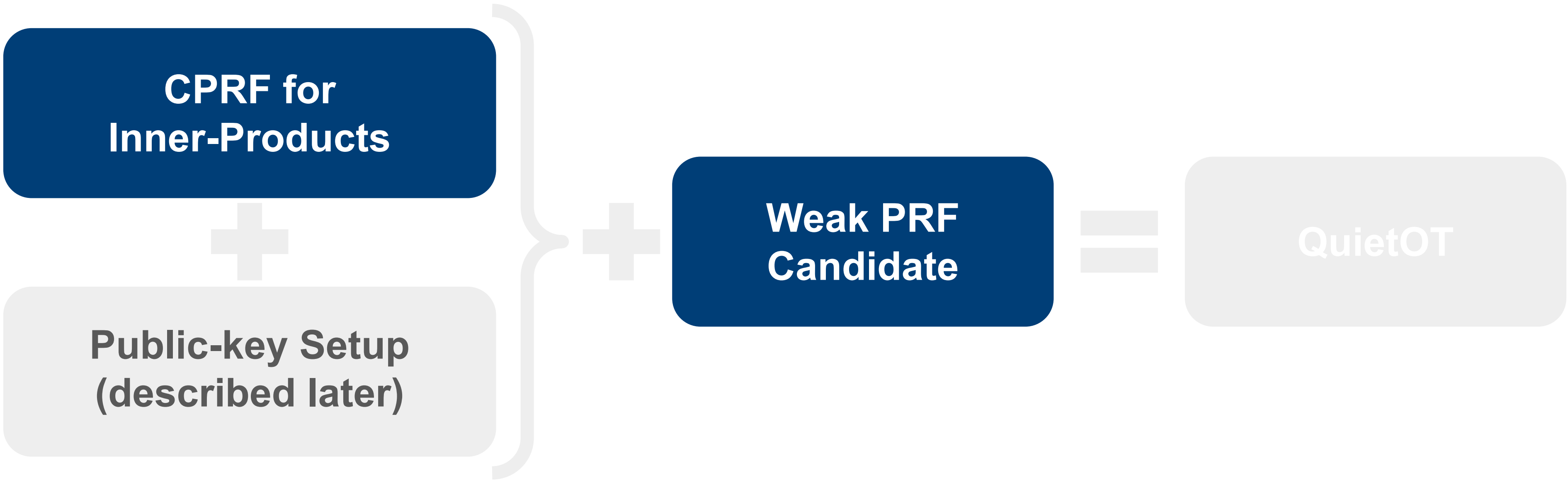
$$f_{\mathbf{k}}(\mathbf{x}) = 0 \iff \langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 \in \{0, 1, 2\}$$

$$f_{\mathbf{k}}(\mathbf{x}) = 1 \iff \langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 \in \{3, 4, 5\}$$

QuietOT: The main ingredients



QuietOT: The main ingredients

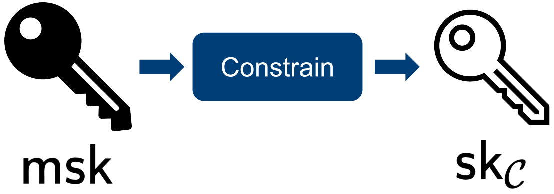


Putting things together

Idea: Use a PRF as the constraint predicate

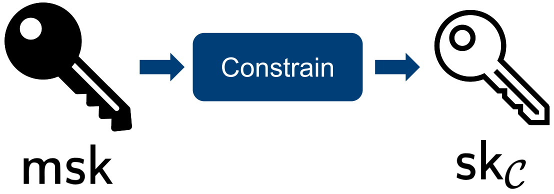
Inspired by previous constructions building OT extension, in particular [BCMPR'24]

Idea: Use a PRF as the constraint predicate



$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

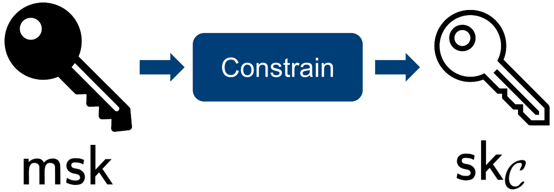
Idea: Use a PRF as the constraint predicate



$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

$$f_k(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

Idea: Use a PRF as the constraint predicate

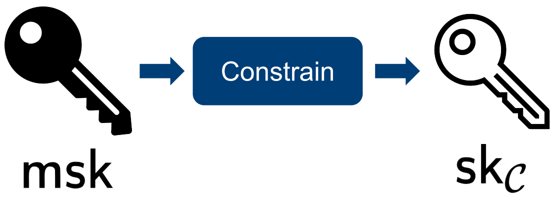


$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

A PRF

$$f_k(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

Idea: Use a PRF as the constraint predicate



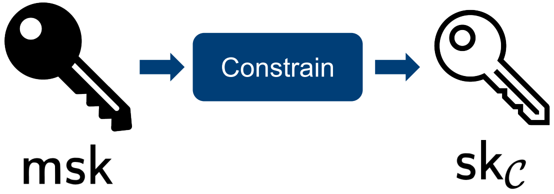
$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

$$f_k(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

A PRF

A *pseudorandom* subset of the CPRF domain is authorized.

Idea: Use a PRF as the constraint predicate



$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

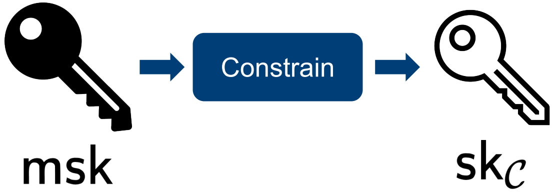
A PRF

$$f_k(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

A *pseudorandom* subset of the CPRF domain is authorized.

$$f_k(0) = 0$$

Idea: Use a PRF as the constraint predicate



$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

A PRF

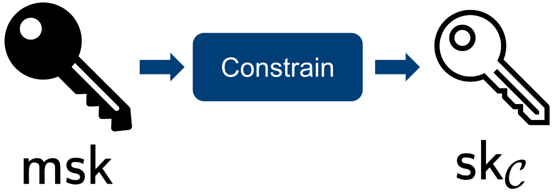
$$f_k(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

A *pseudorandom* subset of the CPRF domain is authorized.

$$f_k(0) = 0$$

$$f_k(1) = 1$$

Idea: Use a PRF as the constraint predicate



$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

A PRF

$$f_k(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

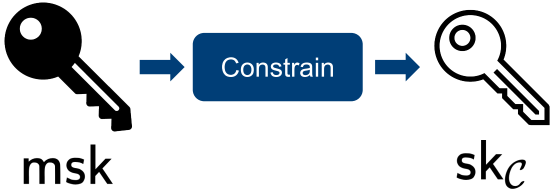
A *pseudorandom* subset of the CPRF domain is authorized.

$f_k(0) = 0$ ✓

$f_k(1) = 1$ ✗

$f_k(2) = 1$ ✗

Idea: Use a PRF as the constraint predicate



$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

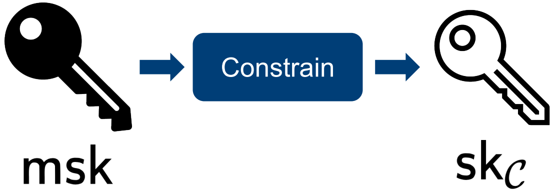
A PRF

A *pseudorandom* subset of the CPRF domain is authorized.

$$f_k(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

$f_k(0) = 0$	$f_k(1) = 1$	$f_k(2) = 1$	$f_k(3) = 0$
✓	✗	✗	✓

Idea: Use a PRF as the constraint predicate



$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

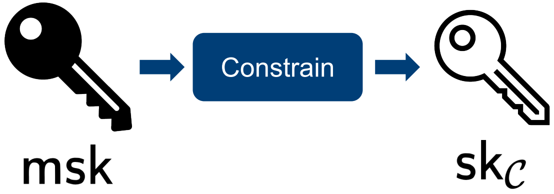
A PRF

A *pseudorandom* subset of the CPRF domain is authorized.

$$f_k(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

$f_k(0) = 0$	$f_k(1) = 1$	$f_k(2) = 1$	$f_k(3) = 0$	$f_k(4) = 0$
✓	✗	✗	✓	✓

Idea: Use BIPSW weak PRF as the predicate

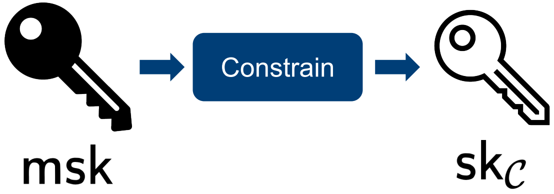


$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

BIPSW weak PRF

$$\lceil \langle \mathbf{k}, \mathbf{x} \rangle \rfloor_2 = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

Idea: Use BIPSW weak PRF as the predicate



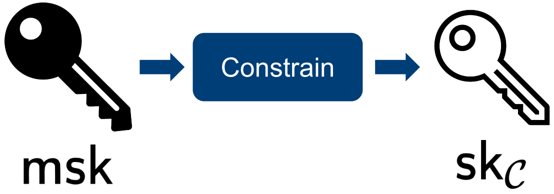
$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

BIPSW weak PRF

$$\lceil \langle \mathbf{k}, \mathbf{x} \rangle \rfloor_2 = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

Problem: inner-product predicate isn't powerful enough

Idea: Use BIPSW weak PRF as the predicate



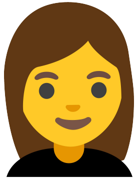
$$C(x) = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

BIPSW weak PRF →

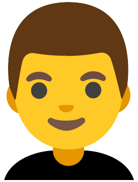
$$\lceil \langle \mathbf{k}, \mathbf{x} \rangle \rfloor_2 = \begin{cases} 0 & \text{authorized} \\ 1 & \text{unauthorized} \end{cases}$$

Solution: use many constrained PRFs!

OT Extension using CPRF + BIPSW weak PRF

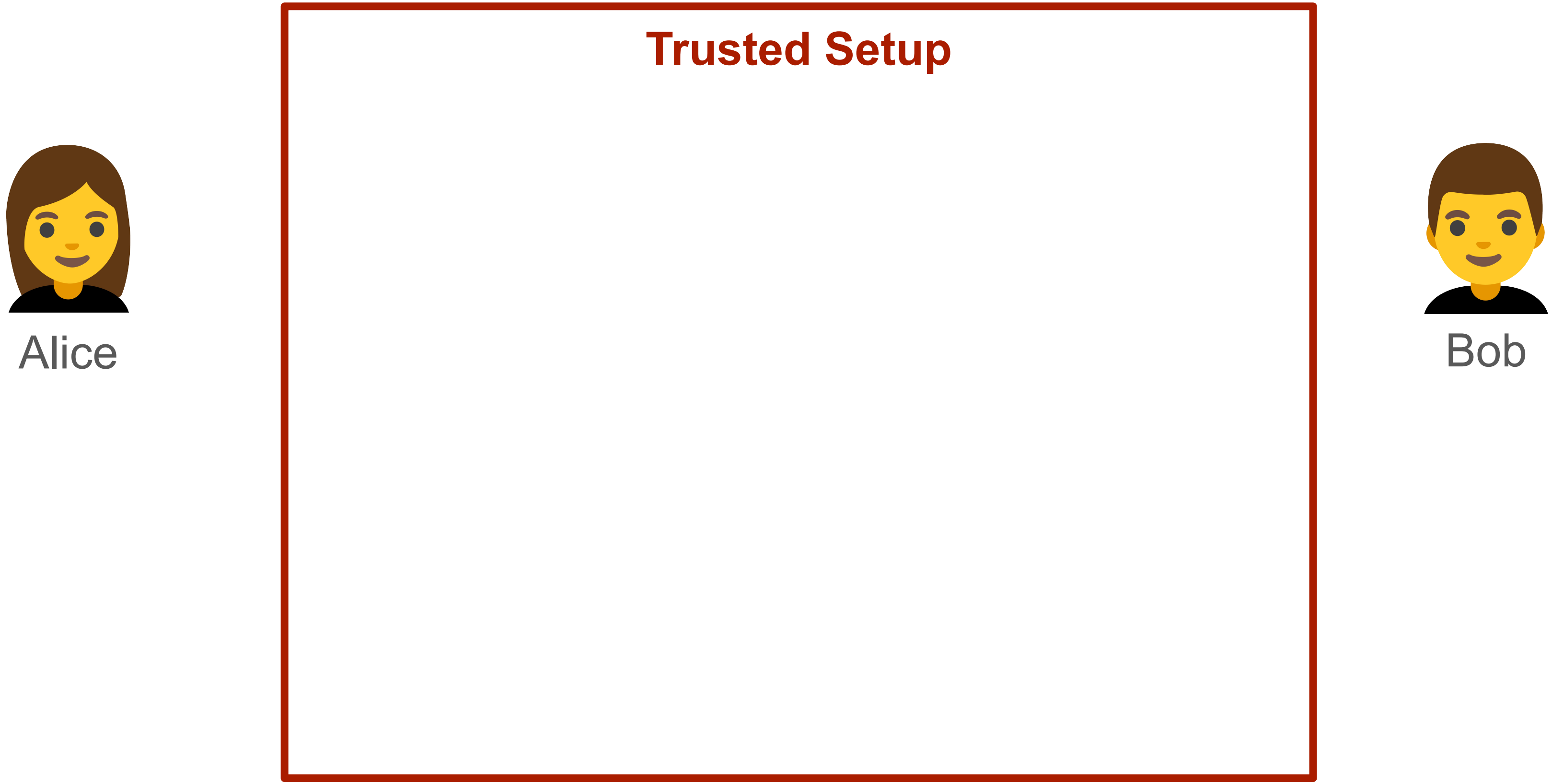


Alice

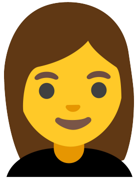


Bob

OT Extension using CPRF + BIPSW weak PRF



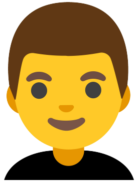
OT Extension using CPRF + BIPSW weak PRF



Alice

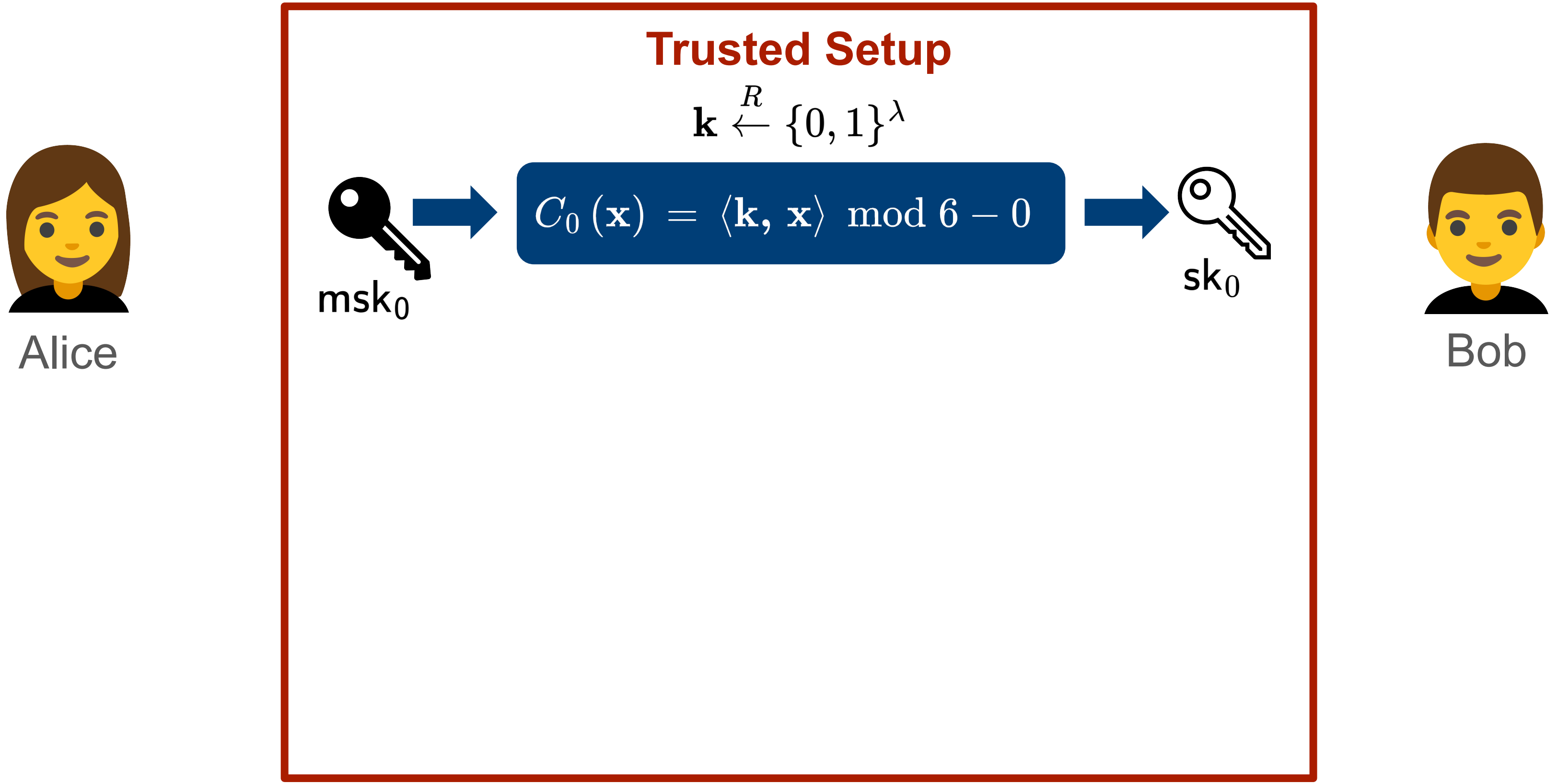
Trusted Setup

$$\mathbf{k} \xleftarrow{R} \{0,1\}^\lambda$$

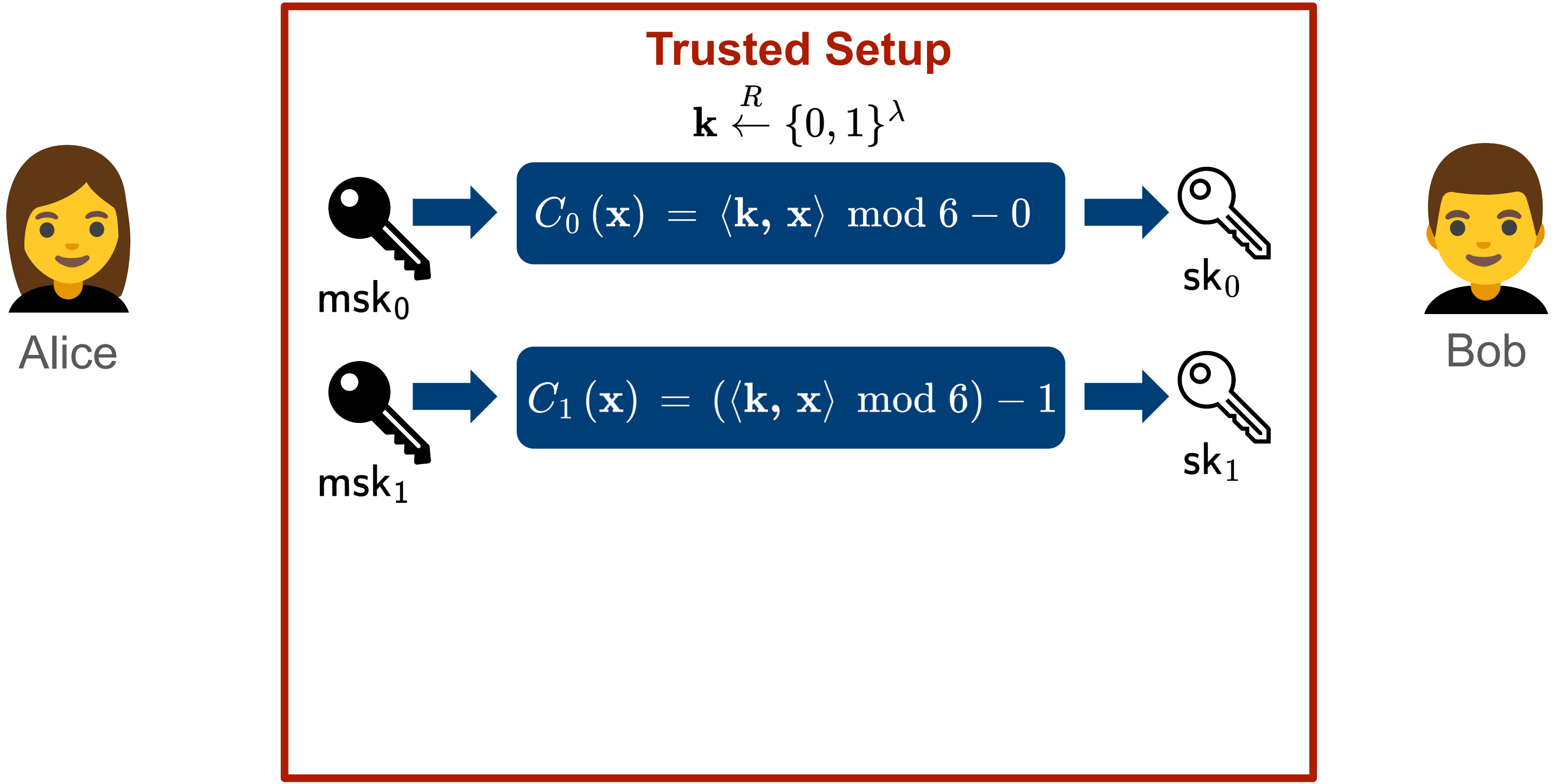


Bob

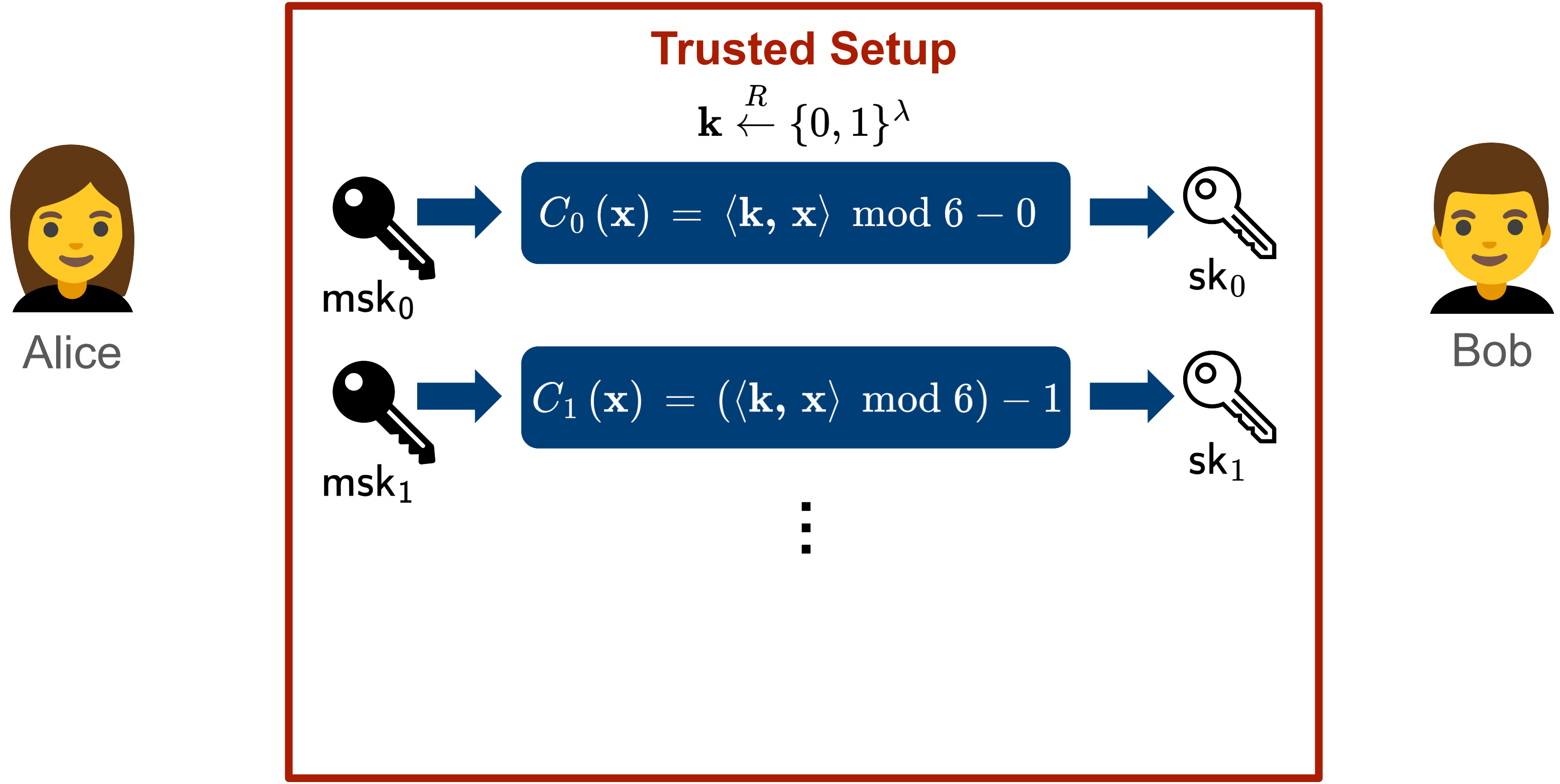
OT Extension using CPRF + BIPSW weak PRF



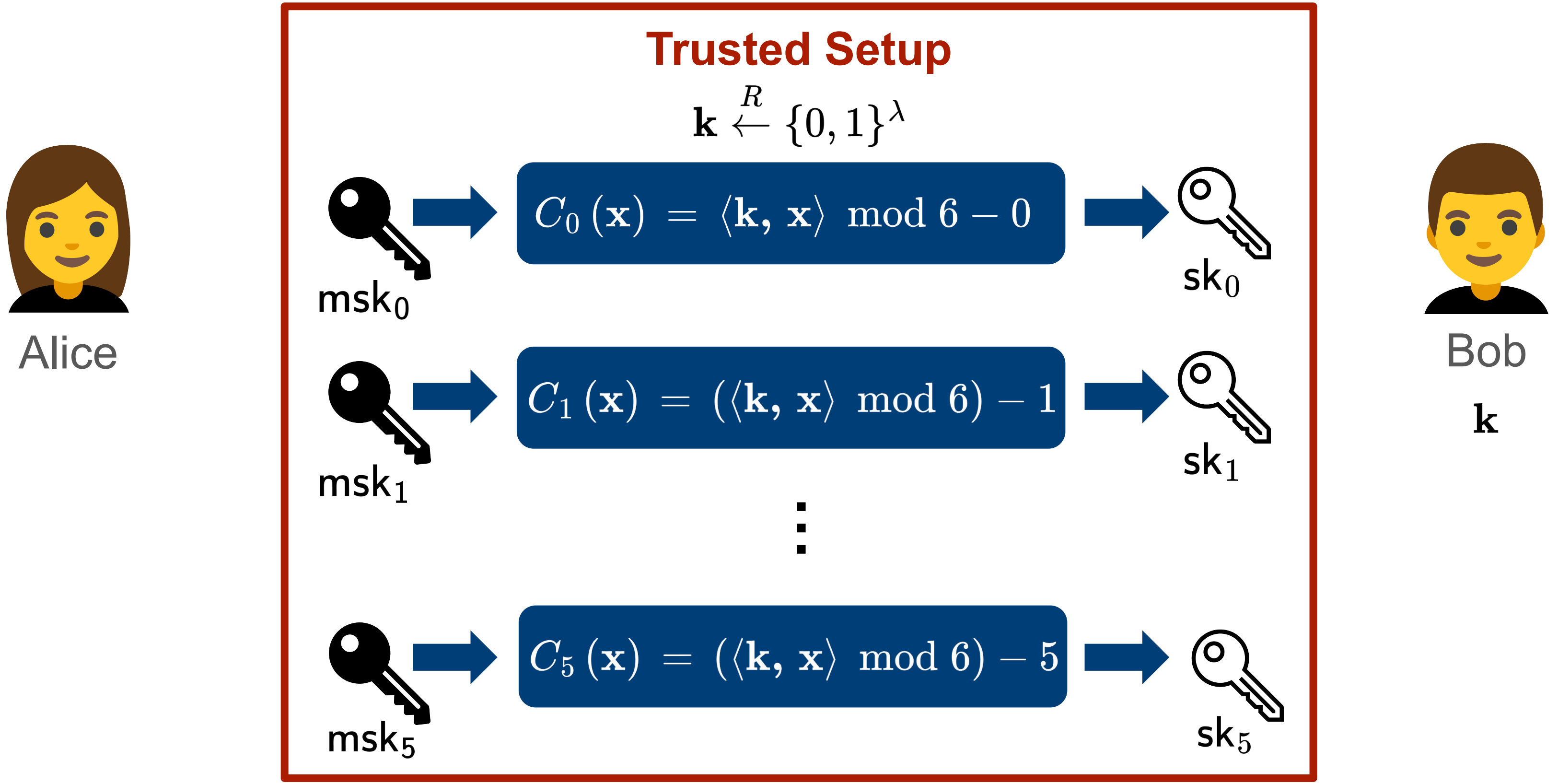
OT Extension using CPRF + BIPSW weak PRF



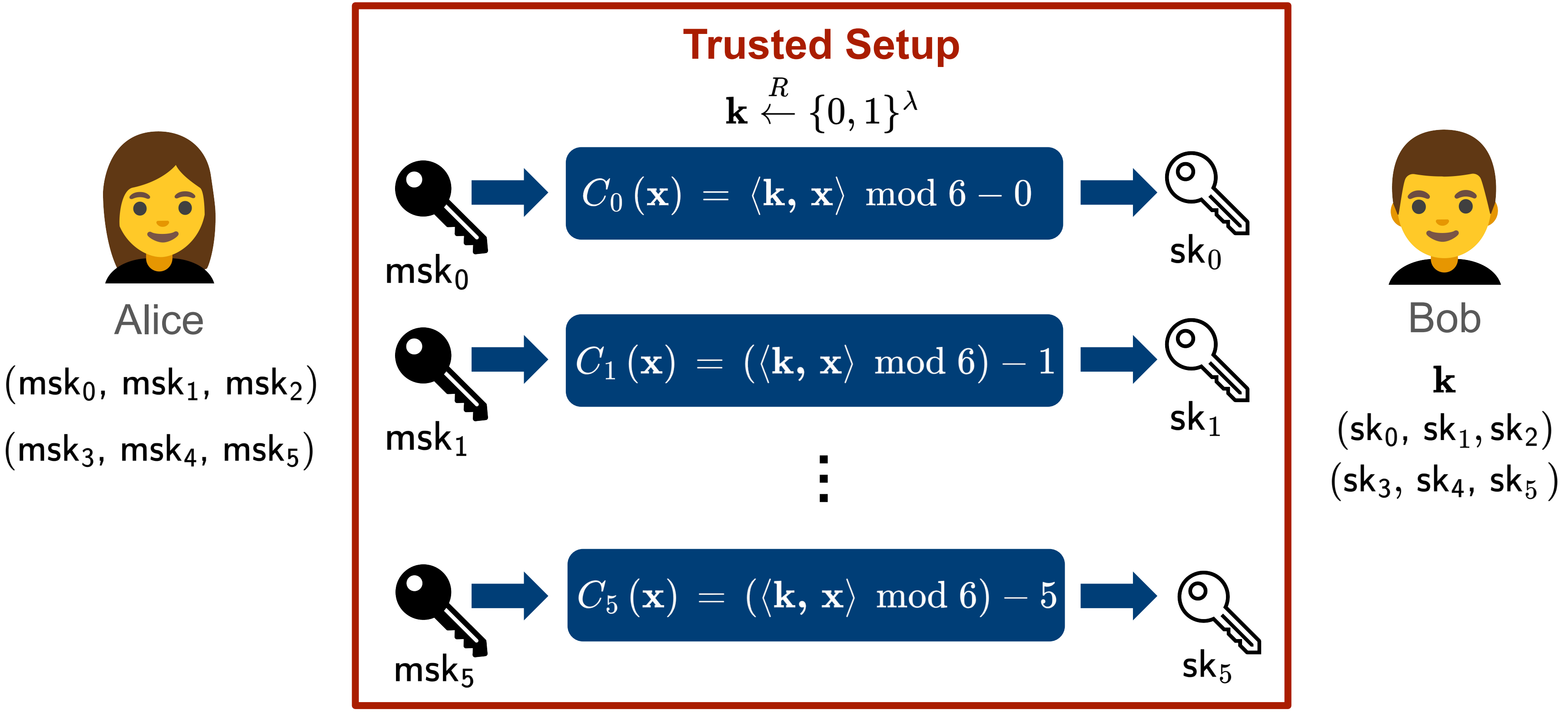
OT Extension using CPRF + BIPSW weak PRF



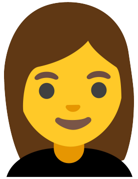
OT Extension using CPRF + BIPSW weak PRF



OT Extension using CPRF + BIPSW weak PRF



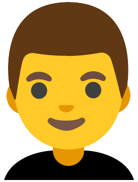
OT Extension using CPRF + BIPSW weak PRF



Alice

$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$



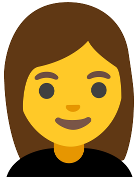
Bob

BIPSW weak PRF Key → $\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1, \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5\ )$

OT Extension using CPRF + BIPSW weak PRF

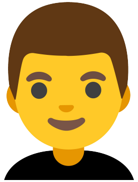


Alice

$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$

Constrained CPRF keys for each possible inner product value {0,1,2,3,4,5} →



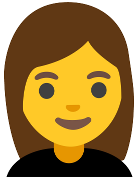
Bob

$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1, \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5\ )$

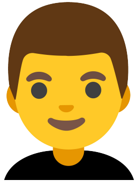
OT Extension using CPRF + BIPSW weak PRF



$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$

← **Master CPRF keys**



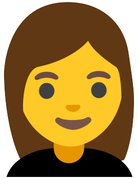
$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1,\mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5\ )$

OT Extension using CPRF + BIPSW weak PRF

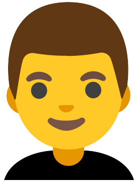
For a public, random input $\mathbf{x}$:



Alice

$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$



Bob

$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1, \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5\ )$

OT Extension using CPRF + BIPSW weak PRF

For a public, random input $\mathbf{x}$:

Alice

$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$

$$a_0 := \mathsf{PRF}(\mathsf{msk}_0,\ \mathbf{x})$$
$$a_1 := \mathsf{PRF}(\mathsf{msk}_1,\ \mathbf{x})$$
$$a_2 := \mathsf{PRF}(\mathsf{msk}_2,\ \mathbf{x})$$
$$a_3 := \mathsf{PRF}(\mathsf{msk}_3,\ \mathbf{x})$$
$$a_4 := \mathsf{PRF}(\mathsf{msk}_4,\ \mathbf{x})$$
$$a_5 := \mathsf{PRF}(\mathsf{msk}_5,\ \mathbf{x})$$

Bob

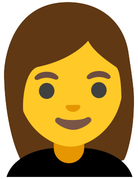
$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1, \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5)$

OT Extension using CPRF + BIPSW weak PRF

For a public, random input $\mathbf{x}$:



Alice

$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$

$$a_0 := \mathsf{PRF}(\mathsf{msk}_0, \mathbf{x})$$
$$a_1 := \mathsf{PRF}(\mathsf{msk}_1, \mathbf{x})$$
$$a_2 := \mathsf{PRF}(\mathsf{msk}_2, \mathbf{x})$$
$$a_3 := \mathsf{PRF}(\mathsf{msk}_3, \mathbf{x})$$
$$a_4 := \mathsf{PRF}(\mathsf{msk}_4, \mathbf{x})$$
$$a_5 := \mathsf{PRF}(\mathsf{msk}_5, \mathbf{x})$$

$$\langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 = 4$$

Bob

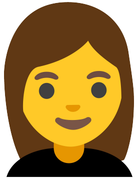
$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1, \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5)$

OT Extension using CPRF + BIPSW weak PRF

For a public, random input $\mathbf{x}$:



Alice

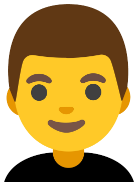
$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$

$$a_0 := \mathsf{PRF}(\mathsf{msk}_0,\ \mathbf{x})$$
$$a_1 := \mathsf{PRF}(\mathsf{msk}_1,\ \mathbf{x})$$
$$a_2 := \mathsf{PRF}(\mathsf{msk}_2,\ \mathbf{x})$$
$$a_3 := \mathsf{PRF}(\mathsf{msk}_3,\ \mathbf{x})$$
$$a_4 := \mathsf{PRF}(\mathsf{msk}_4,\ \mathbf{x})$$
$$a_5 := \mathsf{PRF}(\mathsf{msk}_5,\ \mathbf{x})$$

$$\langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 = 4$$

$$a_4 := \mathsf{PRF}(\mathsf{sk}_4,\ \mathbf{x})$$



Bob

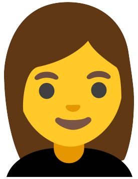
$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1, \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5)$

OT Extension using CPRF + BIPSW weak PRF

For a public, random input $\mathbf{x}$:



Alice

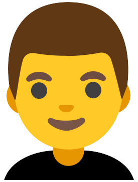
$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$

$$a_0 := \mathsf{PRF}(\mathsf{msk}_0,\ \mathbf{x})$$
$$a_1 := \mathsf{PRF}(\mathsf{msk}_1,\ \mathbf{x})$$
$$a_2 := \mathsf{PRF}(\mathsf{msk}_2,\ \mathbf{x})$$
$$a_3 := \mathsf{PRF}(\mathsf{msk}_3,\ \mathbf{x})$$
$$a_4 := \mathsf{PRF}(\mathsf{msk}_4,\ \mathbf{x})$$
$$a_5 := \mathsf{PRF}(\mathsf{msk}_5,\ \mathbf{x})$$

$$\langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 = 4$$

$$a_4 := \mathsf{PRF}(\mathsf{sk}_4,\ \mathbf{x})$$



Bob

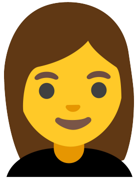
$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1,\ \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5)$

OT Extension using CPRF + BIPSW weak PRF

For a public, random input $\mathbf{x}$:



Alice

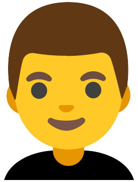
$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$

$$a_0 := \mathsf{PRF}\left(\mathsf{msk}_0,\ \mathbf{x}\right)$$
$$a_1 := \mathsf{PRF}\left(\mathsf{msk}_1,\ \mathbf{x}\right)$$
$$a_2 := \mathsf{PRF}\left(\mathsf{msk}_2,\ \mathbf{x}\right)$$
$$a_3 := \mathsf{PRF}\left(\mathsf{msk}_3,\ \mathbf{x}\right)$$
$$a_4 := \mathsf{PRF}\left(\mathsf{msk}_4,\ \mathbf{x}\right)$$
$$a_5 := \mathsf{PRF}\left(\mathsf{msk}_5,\ \mathbf{x}\right)$$

$$\langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 = 1$$

$$a_1 := \mathsf{PRF}\left(\mathsf{sk}_1,\ \mathbf{x}\right)$$



Bob

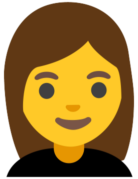
$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1, \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5)$

OT Extension using CPRF + BIPSW weak PRF

For a public, random input $\mathbf{x}$:



Alice

$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$

$a_0 := \mathsf{PRF}(\mathsf{msk}_0, \mathbf{x})$

$a_1 := \mathsf{PRF}(\mathsf{msk}_1, \mathbf{x})$

$a_2 := \mathsf{PRF}(\mathsf{msk}_2, \mathbf{x})$

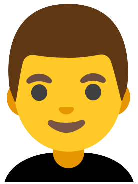
$a_3 := \mathsf{PRF}(\mathsf{msk}_3, \mathbf{x})$

$a_4 := \mathsf{PRF}(\mathsf{msk}_4, \mathbf{x})$

$a_5 := \mathsf{PRF}(\mathsf{msk}_5, \mathbf{x})$

$\langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 = 1$

$a_1 := \mathsf{PRF}(\mathsf{sk}_1, \mathbf{x})$



Bob

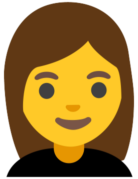
$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1, \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5)$

OT Extension using CPRF + BIPSW weak PRF

For a public, random input $\mathbf{x}$:



Alice

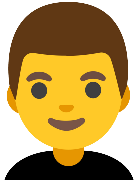
$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$

$$a_0 := \mathsf{PRF}(\mathsf{msk}_0,\ \mathbf{x})$$
$$a_1 := \mathsf{PRF}(\mathsf{msk}_1,\ \mathbf{x})$$
$$a_2 := \mathsf{PRF}(\mathsf{msk}_2,\ \mathbf{x})$$
$$a_3 := \mathsf{PRF}(\mathsf{msk}_3,\ \mathbf{x})$$
$$a_4 := \mathsf{PRF}(\mathsf{msk}_4,\ \mathbf{x})$$
$$a_5 := \mathsf{PRF}(\mathsf{msk}_5,\ \mathbf{x})$$

$$\langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 = 2$$

$$a_2 := \mathsf{PRF}(\mathsf{sk}_2,\ \mathbf{x})$$



Bob

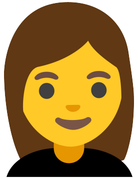
$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1, \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5)$

OT Extension using CPRF + BIPSW weak PRF

For a public, random input $\mathbf{x}$:



Alice

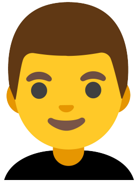
$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$

$$a_0 := \mathsf{PRF}\left(\mathsf{msk}_0,\ \mathbf{x}\right)$$
$$a_1 := \mathsf{PRF}\left(\mathsf{msk}_1,\ \mathbf{x}\right)$$
$$a_2 := \mathsf{PRF}\left(\mathsf{msk}_2,\ \mathbf{x}\right)$$
$$a_3 := \mathsf{PRF}\left(\mathsf{msk}_3,\ \mathbf{x}\right)$$
$$a_4 := \mathsf{PRF}\left(\mathsf{msk}_4,\ \mathbf{x}\right)$$
$$a_5 := \mathsf{PRF}\left(\mathsf{msk}_5,\ \mathbf{x}\right)$$

$$\langle \mathbf{k}, \mathbf{x} \rangle \bmod 6 = 2$$

$$a_2 := \mathsf{PRF}\left(\mathsf{sk}_2,\ \mathbf{x}\right)$$



Bob

$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1, \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5)$

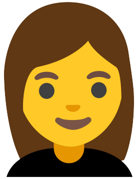
Random-bit OT

Random-bit OT

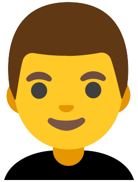
Theorem [Beaver'96]: If there exists a *random-bit* OT protocol with R rounds, then there exists a *chosen-bit* OT protocol with R rounds.

Getting random-bit OT

In random-bit OT, Bob does not get to set the "choice bit" b



Alice

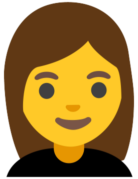


Bob

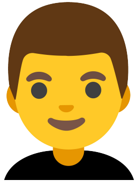
(m_0, m_1)

Getting random-bit OT

In random-bit OT, Bob does not get to set the "choice bit" b



Alice



Bob

$(m_0,\ m_1)$

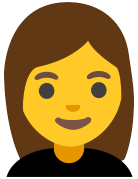
(a_0, a_1, a_2)
(a_3, a_4, a_5)

← **Computed using the CPRFs** →

$(c,\ a_c)$

Getting random-bit OT

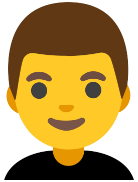
In random-bit OT, Bob does not get to set the "choice bit" b



Alice

$$(a_0 \oplus m_0, a_1 \oplus m_0, a_2 \oplus m_0)$$
$$(a_3 \oplus m_1, a_4 \oplus m_1, a_5 \oplus m_1)$$

$\longrightarrow$



Bob

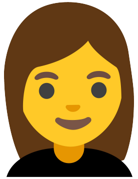
(m_0, m_1)

(a_0, a_1, a_2)
(a_3, a_4, a_5)

(c, a_c)

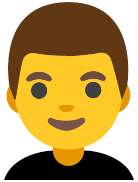
Getting random-bit OT

In random-bit OT, Bob does not get to set the "choice bit" b



Alice

$$(a_0 \oplus m_0, a_1 \oplus m_0, a_2 \oplus m_0)$$
$$(a_3 \oplus m_1, a_4 \oplus m_1, a_5 \oplus m_1)$$



Bob

(m_0, m_1)

(a_0, a_1, a_2)
(a_3, a_4, a_5)

(c, a_c)

$$m_b := (m_b \oplus a_c) \oplus a_c$$

Getting random-bit OT

In random-bit OT, Bob does not get to set the "choice bit" b

$$(a_0 \oplus m_0, a_1 \oplus m_0, a_2 \oplus m_0)$$
$$(a_3 \oplus m_1, a_4 \oplus m_1, a_5 \oplus m_1)$$

Alice

Bob

$(m_0,\ m_1)$

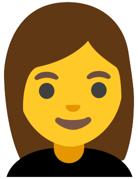
(a_0, a_1, a_2)
(a_3, a_4, a_5)

$(2,\ a_2)$

$$m_0 := (m_0 \oplus a_2) \oplus a_2$$

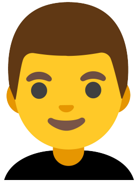
Getting random-bit OT

In random-bit OT, Bob does not get to set the "choice bit" b



Alice

$$(a_0 \oplus m_0, a_1 \oplus m_0, a_2 \oplus m_0)$$
$$(a_3 \oplus m_1, a_4 \oplus m_1, a_5 \oplus m_1)$$



Bob

(m_0, m_1)

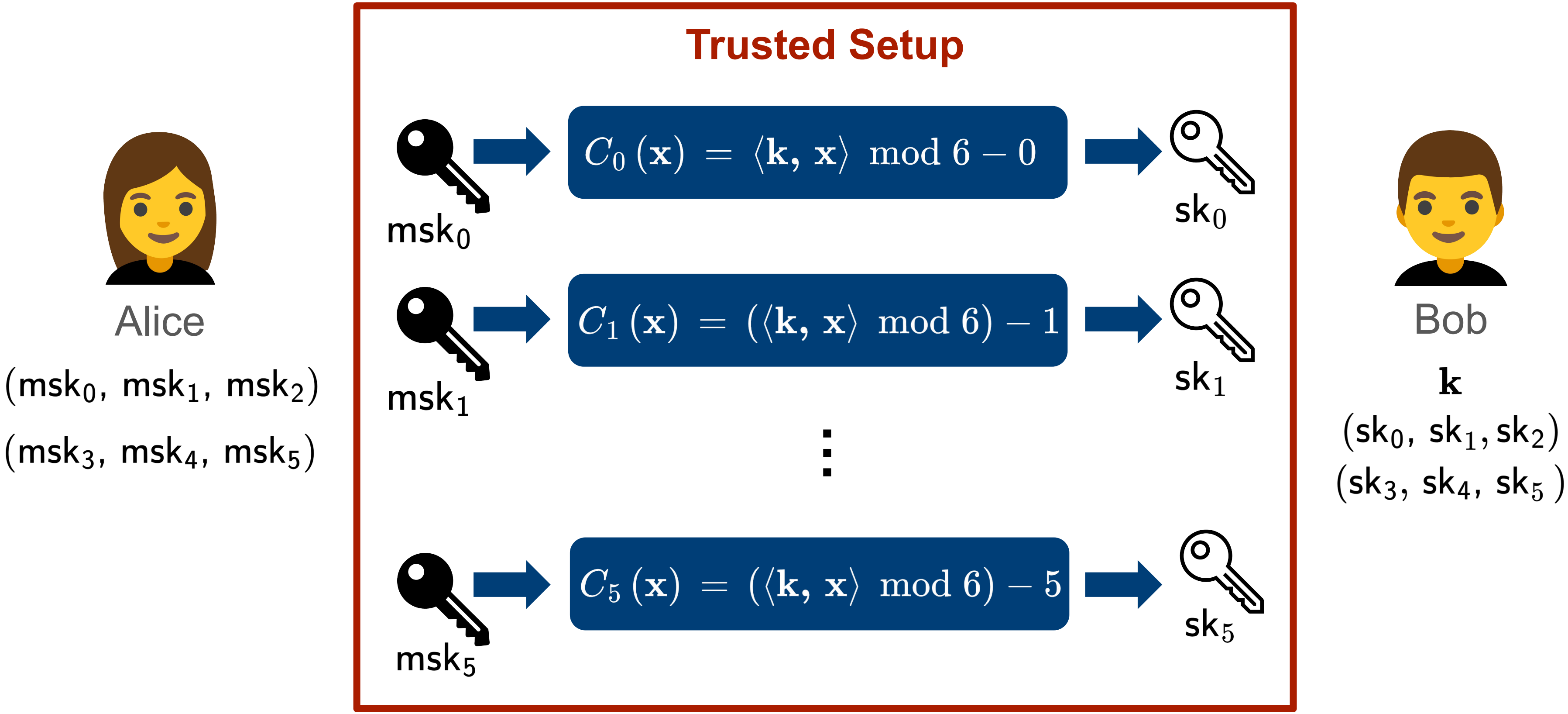
(a_0, a_1, a_2)
(a_3, a_4, a_5)

$(3, a_3)$

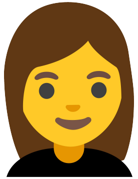
$$m_1 := (m_1 \oplus a_3) \oplus a_3$$

Public-key Setup

Obtaining a public-key setup



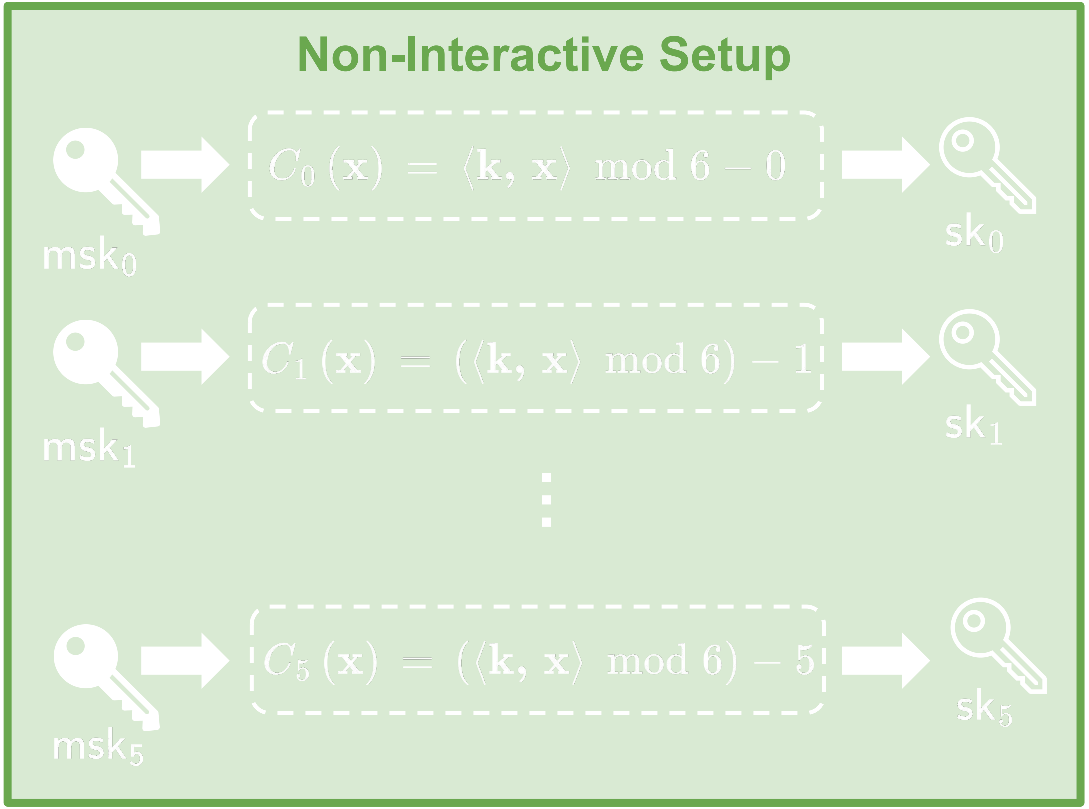
Obtaining a public-key setup

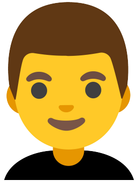


Alice

$(\mathsf{msk}_0,\ \mathsf{msk}_1,\ \mathsf{msk}_2)$

$(\mathsf{msk}_3,\ \mathsf{msk}_4,\ \mathsf{msk}_5)$





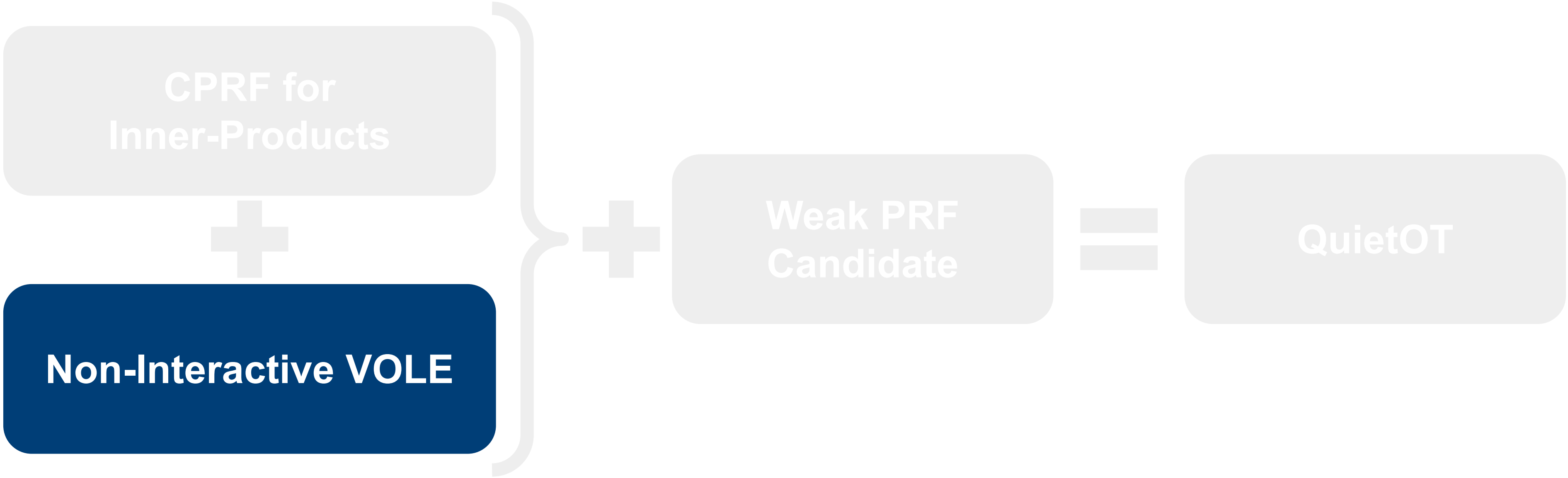
Bob

$\mathbf{k}$

$(\mathsf{sk}_0,\ \mathsf{sk}_1, \mathsf{sk}_2)$

$(\mathsf{sk}_3,\ \mathsf{sk}_4,\ \mathsf{sk}_5\ )$

Obtaining a public-key setup



Obtaining a public-key setup

Constrained PRF for Inner-Product Predicates [S'24]

Let $H : \mathcal{R} \times \mathcal{R}^\ell \to \mathcal{Y}$ be a random oracle (RO).

Obtaining a public-key setup

Constrained PRF for Inner-Product Predicates [S'24]

Let $H : \mathcal{R} \times \mathcal{R}^\ell \to \mathcal{Y}$ be a random oracle (RO).

KeyGen(1^λ **):**

1. $\mathbf{z_0} \xleftarrow{R} \mathcal{R}^\ell$
2. Return msk $:= \mathbf{z_0}$

Constrain(msk, $\mathbf{k}$ **):**

1. $\Delta \xleftarrow{R} \mathcal{R}$
2. $\mathbf{z_1} := \mathbf{z_0} - \Delta \mathbf{k}$
3. Return $\mathsf{sk}_{\mathbf{z}} := \mathbf{z_1}$

Eval(msk, $\mathbf{x}$ **):**

1. $k := \langle \mathbf{z_0}, \mathbf{x} \rangle$
2. Return $H(k, \mathbf{x})$

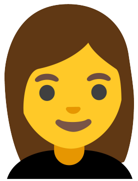
CEval($\mathsf{sk}_{\mathbf{z}}$, $\mathbf{x}$ **):**

1. $k := \langle \mathbf{z_1}, \mathbf{x} \rangle$
2. Output $H(k, \mathbf{x})$

Obtaining a public-key setup

Constrained PRF for Inner-Product Predicates [S'24]

Let $H : \mathcal{R} \times \mathcal{R}^\ell \to \mathcal{Y}$ be a random oracle (RO).



Alice

KeyGen(1^λ):

1. $\mathbf{z_0} \xleftarrow{R} \mathcal{R}^\ell$
2. Return msk $:= \mathbf{z}_0$

Constrain(msk , $\mathbf{k}$):

1. $\Delta \xleftarrow{R} \mathcal{R}$
2. $\mathbf{z_1} := \mathbf{z_0} - \Delta\mathbf{k}$
3. Return $\mathsf{sk}_{\mathbf{z}} := \mathbf{z}_1$

Bob

$\Delta \xleftarrow{R} \mathcal{R}$

$\mathbf{k} \in \mathcal{R}^\ell$

Obtaining a public-key setup

Constrained PRF for Inner-Product Predicates [S'24]

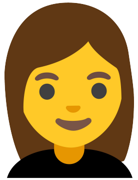
Let $H : \mathcal{R} \times \mathcal{R}^\ell \rightarrow \mathcal{Y}$ be a random oracle (RO).

KeyGen(1^λ):

1. $\mathbf{z_0} \xleftarrow{R} \mathcal{R}^\ell$
2. Return msk $:= \mathbf{z_0}$

Constrain(msk, $\mathbf{k}$):

1. $\Delta \xleftarrow{R} \mathcal{R}$
2. $\mathbf{z_1} := \mathbf{z_0} - \Delta\mathbf{k}$
3. Return $\mathsf{sk}_\mathbf{z} := \mathbf{z_1}$



Alice

$\Delta \xleftarrow{R} \mathcal{R}$

Bob

$\mathbf{k} \in \mathcal{R}^\ell$

Non-Interactive VOLE

Obtaining a public-key setup

Constrained PRF for Inner-Product Predicates [S'24]

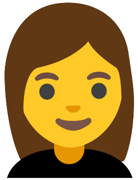
Let $H : \mathcal{R} \times \mathcal{R}^\ell \to \mathcal{Y}$ be a random oracle (RO).

KeyGen(1^λ **):**

1. $\mathbf{z_0} \xleftarrow{R} \mathcal{R}^\ell$
2. Return msk $:= \mathbf{z_0}$

Constrain(msk, $\mathbf{k}$ **):**

1. $\Delta \xleftarrow{R} \mathcal{R}$
2. $\mathbf{z_1} := \mathbf{z_0} - \Delta\mathbf{k}$
3. Return $\mathsf{sk}_\mathbf{z} := \mathbf{z_1}$


Alice


Bob

$\Delta \xleftarrow{R} \mathcal{R}$ → **Non-Interactive VOLE** ← $\mathbf{k} \in \mathcal{R}^\ell$

Obtaining a public-key setup

Constrained PRF for Inner-Product Predicates [S'24]

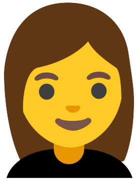
Let $H : \mathcal{R} \times \mathcal{R}^\ell \to \mathcal{Y}$ be a random oracle (RO).

KeyGen(1^λ):

1. $\mathbf{z_0} \xleftarrow{R} \mathcal{R}^\ell$
2. Return msk $:= \mathbf{z_0}$

Constrain(msk , $\mathbf{k}$):

1. $\Delta \xleftarrow{R} \mathcal{R}$
2. $\mathbf{z_1} := \mathbf{z_0} - \Delta\mathbf{k}$
3. Return $\mathsf{sk}_{\mathbf{z}} := \mathbf{z_1}$



Alice

Bob

$\Delta \xleftarrow{R} \mathcal{R}$ → **Non-Interactive VOLE** ← $\mathbf{k} \in \mathcal{R}^\ell$

$\mathbf{z_0}$ ← **Non-Interactive VOLE** → $\mathbf{z_1}$

Obtaining a public-key setup

Constrained PRF for Inner-Product Predicates [S'24]

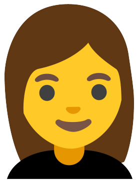
Let $H : \mathcal{R} \times \mathcal{R}^\ell \to \mathcal{Y}$ be a random oracle (RO).

KeyGen(1^λ):

1. $\mathbf{z}_0 \xleftarrow{R} \mathcal{R}^\ell$
2. Return msk $:= \mathbf{z}_0$

Constrain(msk , $\mathbf{k}$):

1. $\Delta \xleftarrow{R} \mathcal{R}$
2. $\mathbf{z}_1 := \mathbf{z}_0 - \Delta\mathbf{k}$
3. Return $\mathsf{sk}_{\mathbf{z}} := \mathbf{z}_1$



Alice

Bob

$\Delta \xleftarrow{R} \mathcal{R}$ → **Non-Interactive VOLE** ← $\mathbf{k} \in \mathcal{R}^\ell$

$\mathbf{z}_0$ ← **Non-Interactive VOLE** → $\mathbf{z}_1$

$$\mathbf{z}_0 - \mathbf{z}_1 = \Delta \cdot \mathbf{k}$$

Obtaining a public-key setup

Constrained PRF for Inner-Product Predicates [S'24]

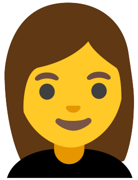
Let $H : \mathcal{R} \times \mathcal{R}^\ell \rightarrow \mathcal{Y}$ be a random oracle (RO).

KeyGen(

1. $\mathbf{z}_0$

2. Retu

Previously known from group-based assumptions (e.g., [OSY'21]).
We give the RingLWE construction of Non-Interactive VOLE



Alice

Bob

$\Delta \xleftarrow{R} \mathcal{R}$ → **Non-Interactive VOLE** ← $\mathbf{k} \in \mathcal{R}^\ell$

$\mathbf{z}_0$ ← → $\mathbf{z}_1$

$$\mathbf{z}_0 - \mathbf{z}_1 = \Delta \cdot \mathbf{k}$$

Implementation and Evaluation

Implementation and Evaluation

Artifact Badges: Available, Functional, and Reproduced.

https://github.com/sachaservan/QuietOT

Protocol	OT/s	Bits/OT

Protocol	OT/s	Bits/OT
SoftSpokenOT	44,443,000	32

Protocol	OT/s	Bits/OT
SoftSpokenOT	44,443,000	32
SoftSpokenOT	76,000	8

Protocol	OT/s	Bits/OT
SoftSpokenOT	44,443,000	32
SoftSpokenOT	76,000	8
RRT	6,856,000	3

Protocol	OT/s	Bits/OT
SoftSpokenOT	44,443,000	32
SoftSpokenOT	76,000	8
RRT	6,856,000	3
$\mathbb{F}_4$OLEAGE	25,000,000	3

Protocol	OT/s	Bits/OT	Sender PK Size	Receiver PK Size
SoftSpokenOT	44,443,000	32	No Public Key Setup	
SoftSpokenOT	76,000	8		
RRT	6,856,000	3		
$\mathbb{F}_4$OLEAGE	25,000,000	3		

<table>
<tr><th>Protocol</th><th>OT/s</th><th>Bits/OT</th><th>Sender PK Size</th><th>Receiver PK Size</th></tr>
<tr><td>SoftSpokenOT</td><td>44,443,000</td><td>32</td><td rowspan="4" colspan="2">No Public Key Setup</td></tr>
<tr><td>SoftSpokenOT</td><td>76,000</td><td>8</td></tr>
<tr><td>RRT</td><td>6,856,000</td><td>3</td></tr>
<tr><td>$\mathbb{F}_4$OLEAGE</td><td>25,000,000</td><td>3</td></tr>
<tr><td>OSY</td><td>1</td><td>3</td><td>50 KB</td><td>1 KB</td></tr>
</table>

Protocol	OT/s	Bits/OT	Sender PK Size	Receiver PK Size
SoftSpokenOT	44,443,000	32	No Public Key Setup	
SoftSpokenOT	76,000	8		
RRT	6,856,000	3		
$\mathbb{F}_4$OLEAGE	25,000,000	3		
OSY	1	3	50 KB	1 KB
BCMPR	12,000	3	63 KB	72 KB

Protocol	OT/s	Bits/OT	Sender PK Size	Receiver PK Size
SoftSpokenOT	44,443,000	32	No Public Key Setup	
SoftSpokenOT	76,000	8		
RRT	6,856,000	3		
$\mathbb{F}_4$OLEAGE	25,000,000	3		
OSY	1	3	50 KB	1 KB
BCMPR	12,000	3	63 KB	72 KB
QuietOT	**561,000**	**7**	**5.4 MB**	**84 KB**

Protocol	OT/s	Bits/OT	Sender PK Size	Receiver PK Size
SoftSpokenOT	44,443,000	32	No Public Key Setup	
SoftSpokenOT	76,000	8		
RRT	6,856,000	3		
$\mathbb{F}_4$OLEAGE	25,000,000	3		

OSY	1	3	50 KB	1 KB
BCMPR	12,000	3	63 KB	72 KB
QuietOT	**561,000**	**7**	**5.4 MB**	**84 KB**
QuietOT (AVX3)	**1,265,000**	**7**	**5.4 MB**	**84 KB**

Protocol	OT/s	Bits/OT	Sender PK Size	Receiver PK Size
SoftSpokenOT	44,443,000	32	No Public Key Setup	
SoftSpokenOT	76,000	8		
RRT	6,856,000	3		
$\mathbb{F}_4$OLEAGE	25,000,000	3		
OSY	1	3		
BCMPR	12,000	3		
QuietOT	**561,000**	**7**	**5.4 MB**	**84 KB**
QuietOT (AVX3)	**1,265,000**	**7**	**5.4 MB**	**84 KB**

For context, average website is 2MB in size

A lot more in the paper!

A lot more in the paper!

- Another construction using a different weak PRF candidate

A lot more in the paper!

- Another construction using a different weak PRF candidate
- A “nearly-black-box” two-round OT extension in the ROM

A lot more in the paper!

- Another construction using a different weak PRF candidate
- A “nearly-black-box” two-round OT extension in the ROM
- A RingLWE-based Non-Interactive VOLE protocol

A lot more in the paper!

- Another construction using a different weak PRF candidate
- A “nearly-black-box” two-round OT extension in the ROM
- A RingLWE-based Non-Interactive VOLE protocol
- Pre-computability and other nice features

A lot more in the paper!

- Another construction using a different weak PRF candidate
- A “nearly-black-box” two-round OT extension in the ROM
- A RingLWE-based Non-Interactive VOLE protocol
- Pre-computability and other nice features

Open questions:

A lot more in the paper!

- Another construction using a different weak PRF candidate
- A "nearly-black-box" two-round OT extension in the ROM
- A RingLWE-based Non-Interactive VOLE protocol
- Pre-computability and other nice features

Open questions:

- Malicious security?

A lot more in the paper!

- Another construction using a different weak PRF candidate
- A “nearly-black-box” two-round OT extension in the ROM
- A RingLWE-based Non-Interactive VOLE protocol
- Pre-computability and other nice features

Open questions:

- Malicious security?
- Can we reach 50,000,000 OT/s with a public key setup?

Thank you!

Email: 3s@mit.edu
ePrint: ia.cr/**2024/1079**

QuietOT: Lightweight Oblivious Transfer with a Public-Key Setup

Geoffroy Couteau[1,2], Lalita Devadas[3], Srinivas Devadas[3], Alexander Koch[1,2], and Sacha Servan-Schreiber[3]

[1] CNRS
[2] IRIF, Université Paris Cité

References

[Beaver'95]: D. Beaver. "Precomputing oblivious transfer." CRYPTO 1995.

[IKNP'03]: Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. "Extending Oblivious Transfers Efficiently." CRYPTO 2003.

[BW'13]: D. Boneh and B. Waters. "Constrained Pseudorandom Functions and Their Applications." ASIACRYPT 2013.

[BGI'14]: E. Boyle, S. Goldwasser, and I. Ivan. "Functional Signatures and Pseudorandom Functions." PKC 2014.

[KPTZ'13]: A. Kiayias, S. Papadopoulos, N. Triandopoulos, and T. Zacharias. "Delegatable Pseudorandom Functions and Applications." CCS 2013.

[BIPSW'18]: D. Boneh, Y. Ishai, A. Passelègue, A. Sahai, and D. J. Wu. "Exploring Crypto Dark Matter: New Simple PRF Candidates and Their Applications." TCC 2018.

[OSY'21]: C. Orlandi, P. Scholl, and S. Yakoubov. "The Rise of Paillier: Homomorphic Secret Sharing and Public-Key Silent OT." EUROCRYPT 2021.

[Roy'22]: L. Roy. "SoftSpokenOT: Quieter OT Extension from Small-Field Silent VOLE in the Minicrypt Model." CRYPTO 2022.

[RRT'23]: S. Raghuraman, P. Rindal, and T. Tanguy. "Expand-Convolute Codes for Pseudorandom Correlation Generators from LPN." CRYPTO 2023.

[BCMPR'24]: D. Bui, G. Couteau, P. Meyer, A. Passelègue, and M. Riahinia. "Fast Public-Key Silent OT and More from Constrained Naor-Reingold." EUROCRYPT 2024.

[S'24]: S. Servan-Schreiber. "Constrained Pseudorandom Functions for Inner-Product Predicates from Weaker Assumptions." ASIACRYPT 2024.