Constrained PRFs for Unbounded Inputs with Short Keys

Hamza Abusalah **Georg Fuchsbauer**

IST AUSTRIA
Institute of Science and Technology

ENS
ÉCOLE NORMALE SUPÉRIEURE

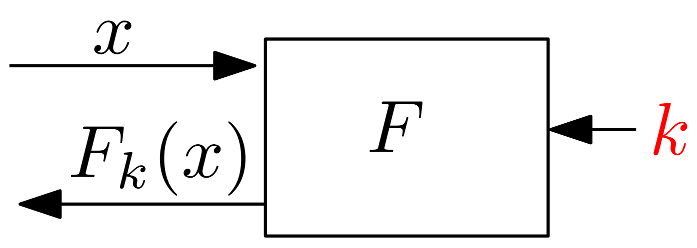
ACNS 2016

Outline

1. Constrained Pseudorandom Functions (CPRFs)
2. Identity-Based Non-interactive Key Exchange
3. Unbounded-Input CPRFs
4. Unbounded-Input CPRFs with Short Keys

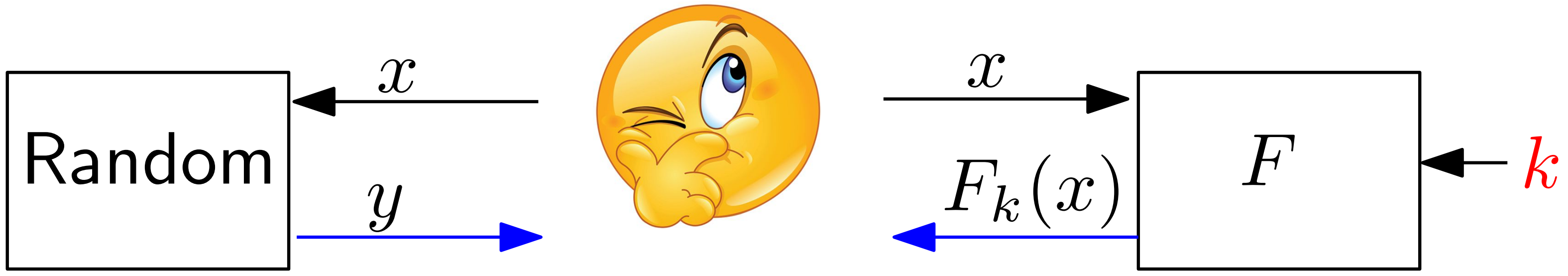
Pseudorandom Functions (PRFs)

[GGM86]



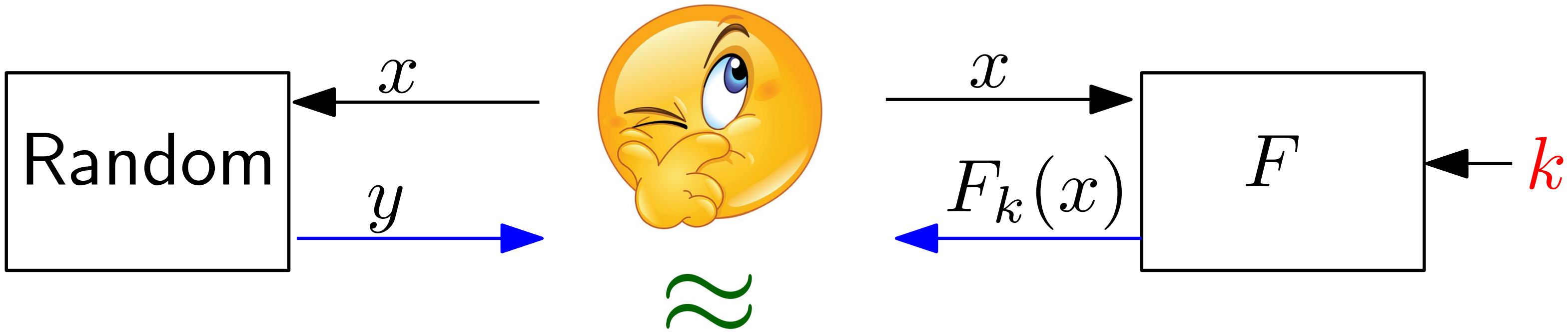
Pseudorandom Functions (PRFs)

[GGM86]



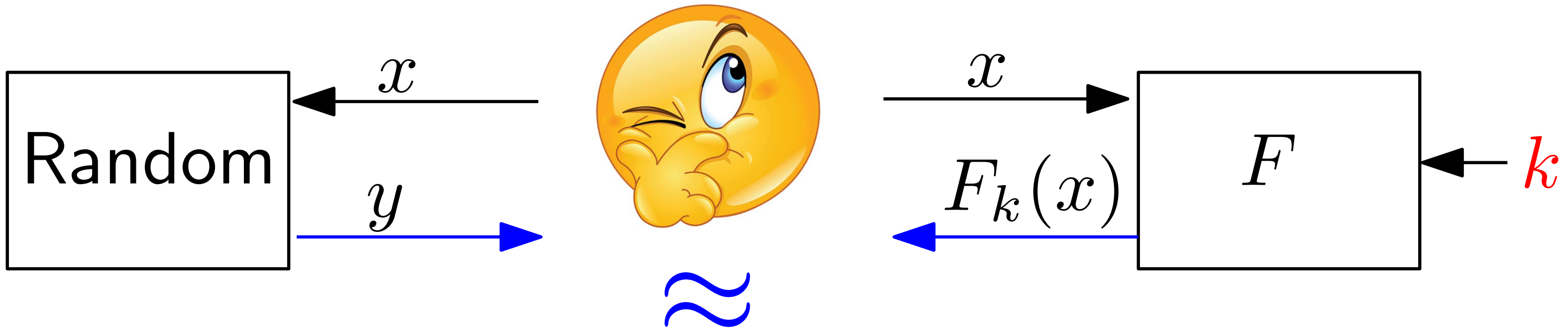
Pseudorandom Functions (PRFs)

[GGM86]



Pseudorandom Functions (PRFs)

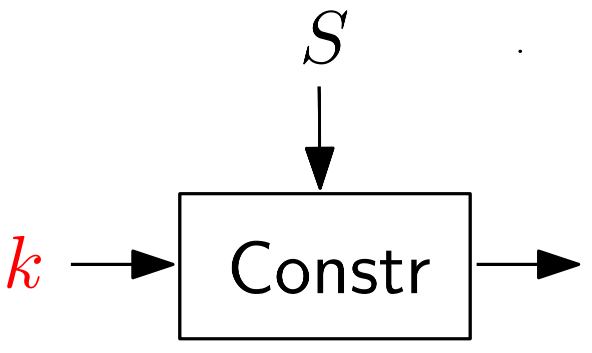
[GGM86]



Unbounded-input PRFs [Goldreich04]: supports $x \in \{0,1\}^*$

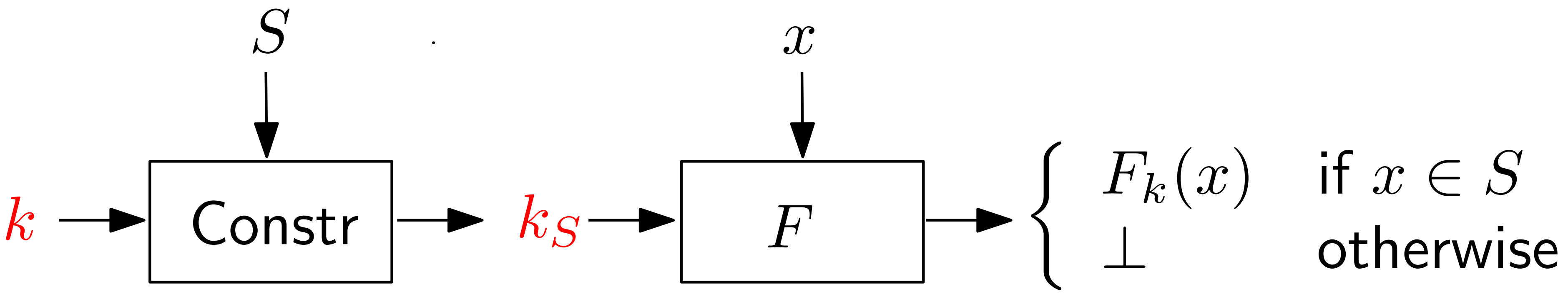
Constrained Pseudorandom Functions (CPRFs)

[BW13],[KPTZ13],[BGI14]



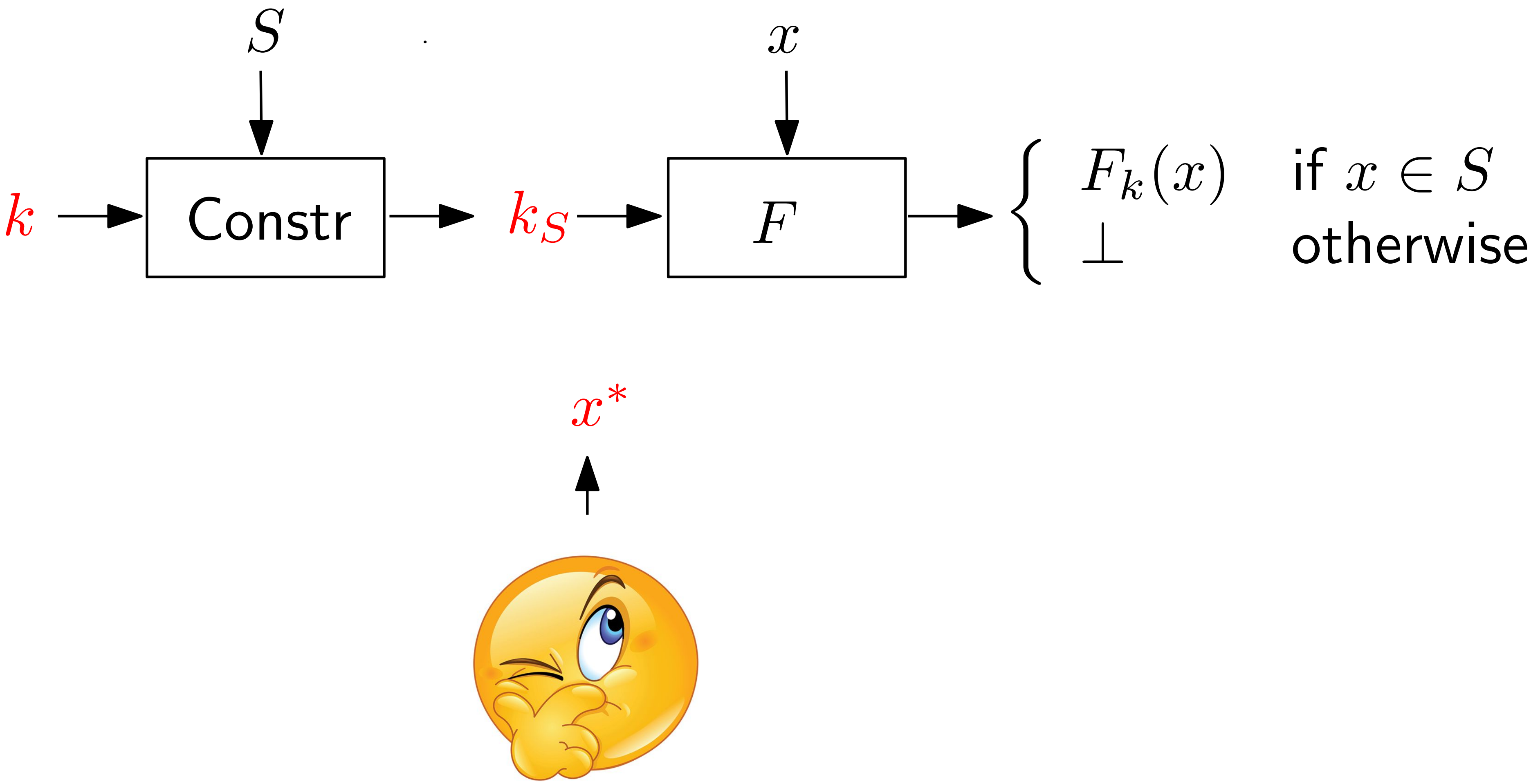
Constrained Pseudorandom Functions (CPRFs)

[BW13],[KPTZ13],[BGI14]



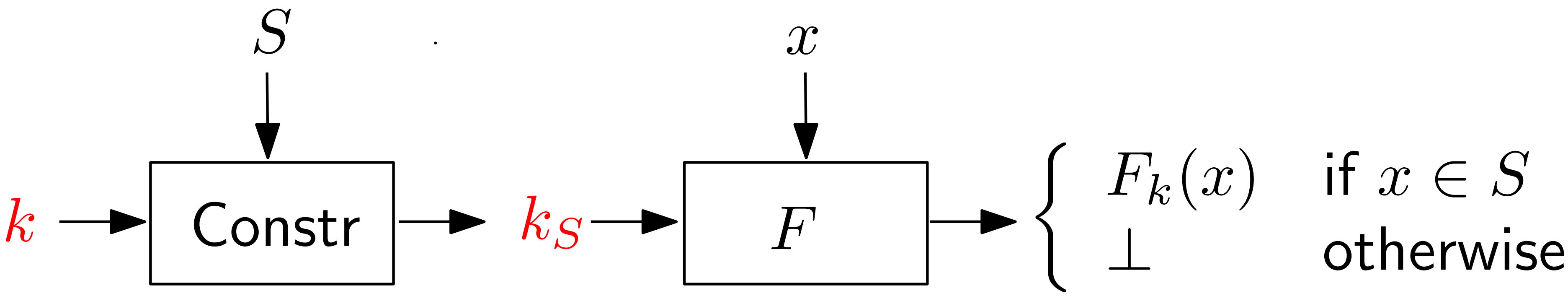
Constrained Pseudorandom Functions (CPRFs)

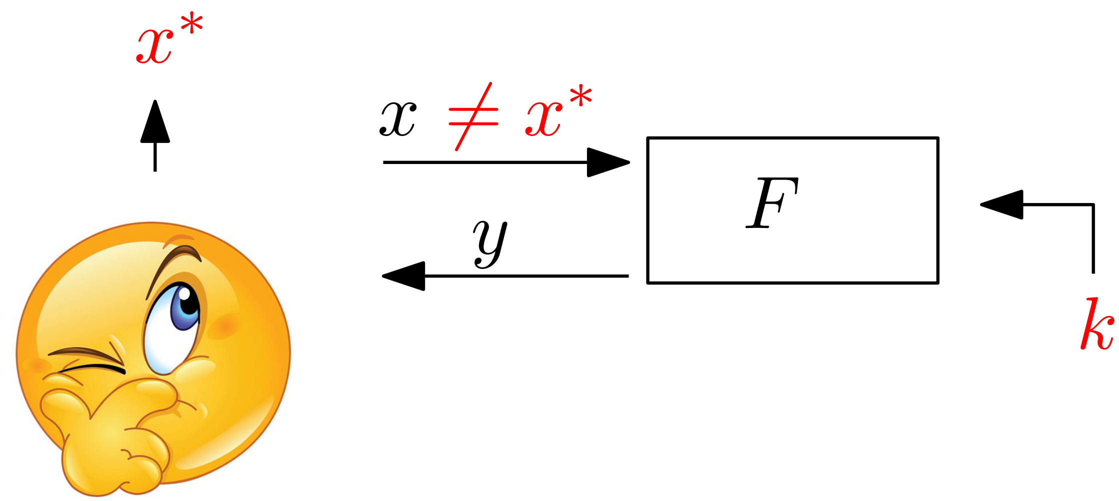
[BW13],[KPTZ13],[BGI14]



Constrained Pseudorandom Functions (CPRFs)

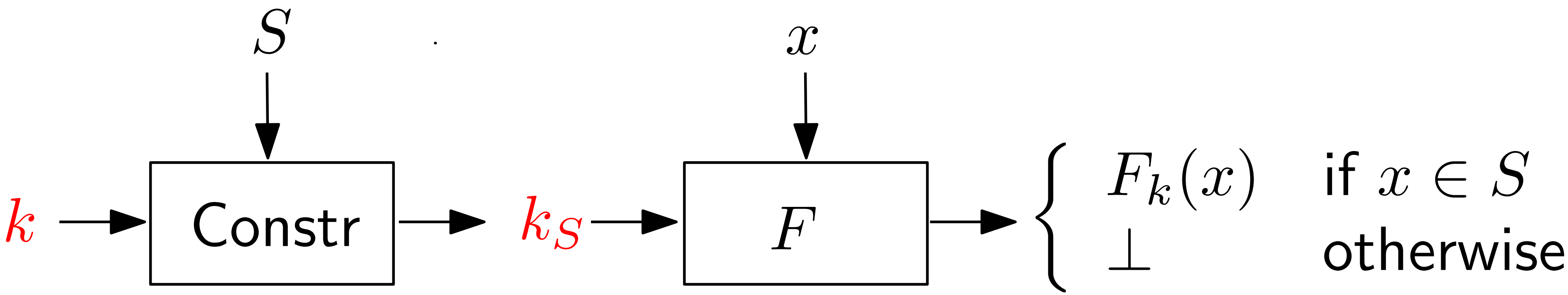
[BW13],[KPTZ13],[BGI14]

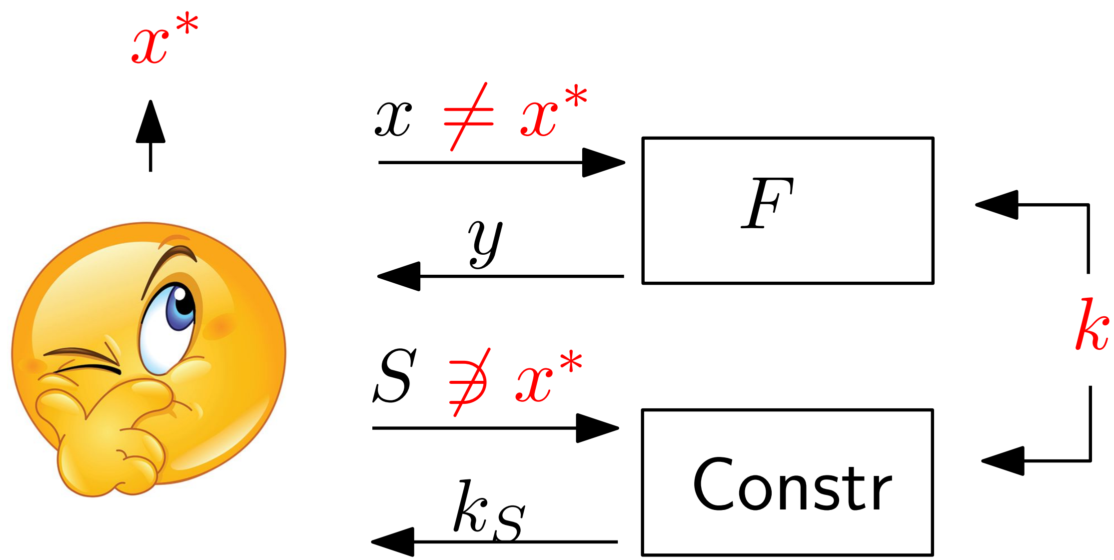




Constrained Pseudorandom Functions (CPRFs)

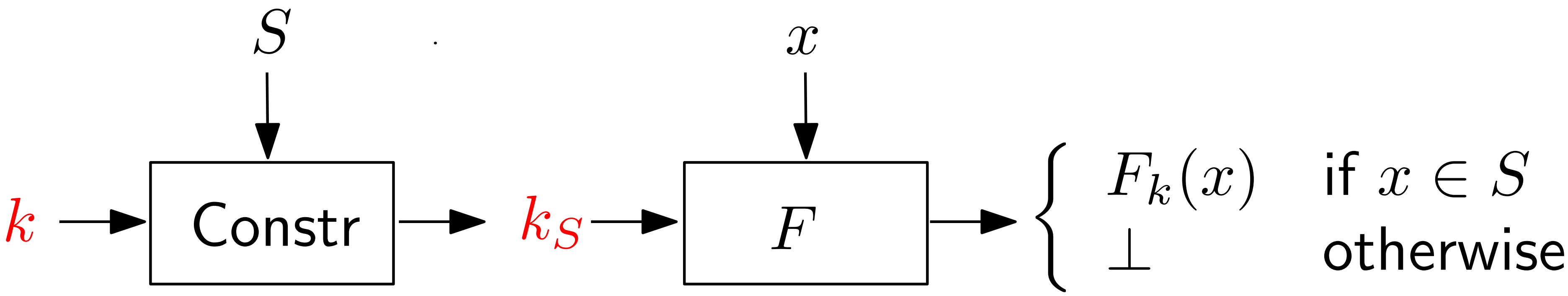
[BW13],[KPTZ13],[BGI14]

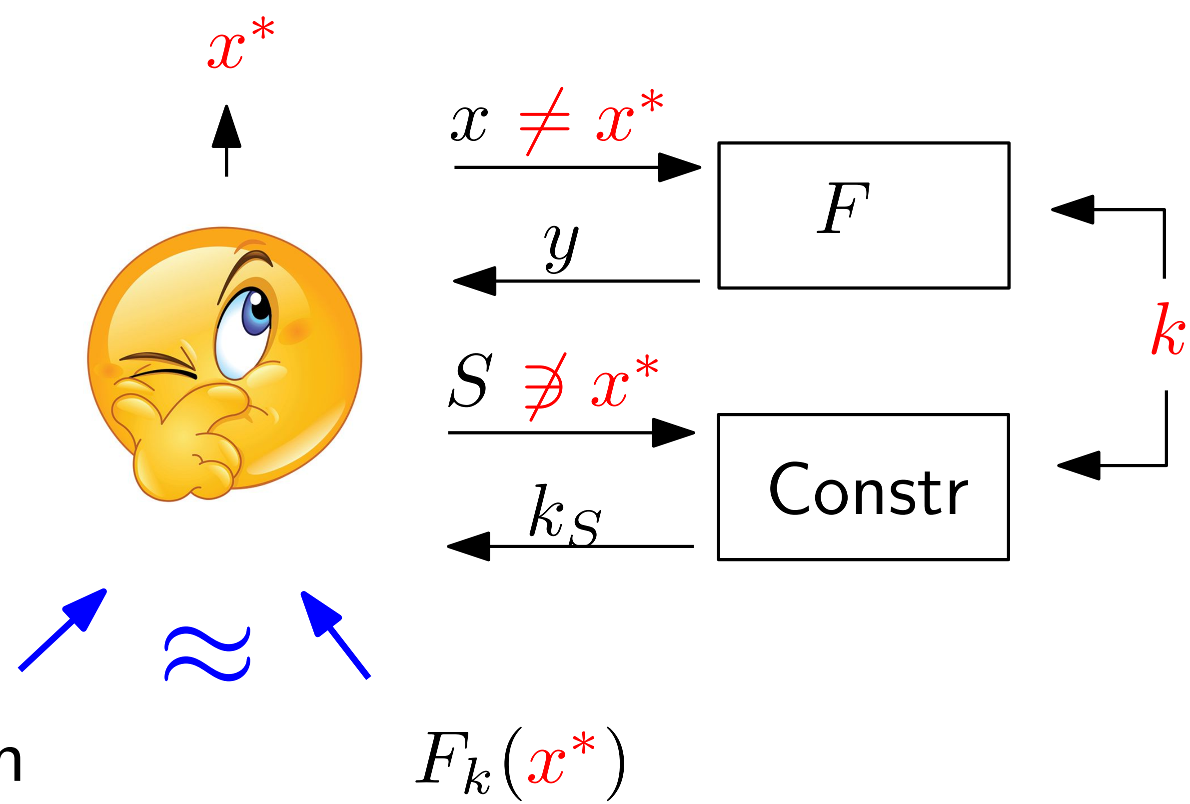




Constrained Pseudorandom Functions (CPRFs)

[BW13],[KPTZ13],[BGI14]





Types of CPRFs

- Polynomial-size S: Any PRF F is a CPRF

$$S = \{x_1, \ldots, x_p\}, \qquad k_S = \{F_k(x_1), \ldots, F_k(x_p)\}$$

Types of CPRFs

- Polynomial-size S: Any PRF F is a CPRF

$$S = \{x_1, \ldots, x_p\}, \qquad k_S = \{F_k(x_1), \ldots, F_k(x_p)\}$$

- Puncturable [SW14]: x^* input

$$k_{x^*} \Rightarrow F_k(x) \text{ if } x \neq x^*$$

(from PRGs)

Types of CPRFs

- Polynomial-size S: Any PRF F is a CPRF

$$S = \{x_1, \ldots, x_p\}, \qquad k_S = \{F_k(x_1), \ldots, F_k(x_p)\}$$

- Puncturable [SW14]: x^* input

$$k_{x^*} \Rightarrow F_k(x) \text{ if } x \neq x^*$$

(from PRGs)

- Circuit [BW13]: C circuit

$$k_C \Rightarrow F_k(x) \text{ if } C(x) = 1$$

(from multilin. maps)

ENS

CPRFs for Unbounded Inputs

- TMs [AFP16]: M Turing machine

$$k_M \Rightarrow F_k(x) \text{ if } M(x) = 1$$

(from public-coin diO)

Accepts unbounded inputs $x \in \{0,1\}^*$

CPRFs for Unbounded Inputs

- TMs [AFP16]: M Turing machine

$$k_M \Rightarrow F_k(x) \text{ if } M(x) = 1$$

(from public-coin diO)

Accepts unbounded inputs $x \in \{0,1\}^*$

- TMs [DKW16]: (from iO)

CPRFs for Unbounded Inputs

- TMs [AFP16]: M Turing machine

$$k_M \Rightarrow F_k(x) \text{ if } M(x) = 1$$

(from public-coin diO)

Accepts unbounded inputs $x \in \{0,1\}^*$

- TMs [DKW16]: (from iO)

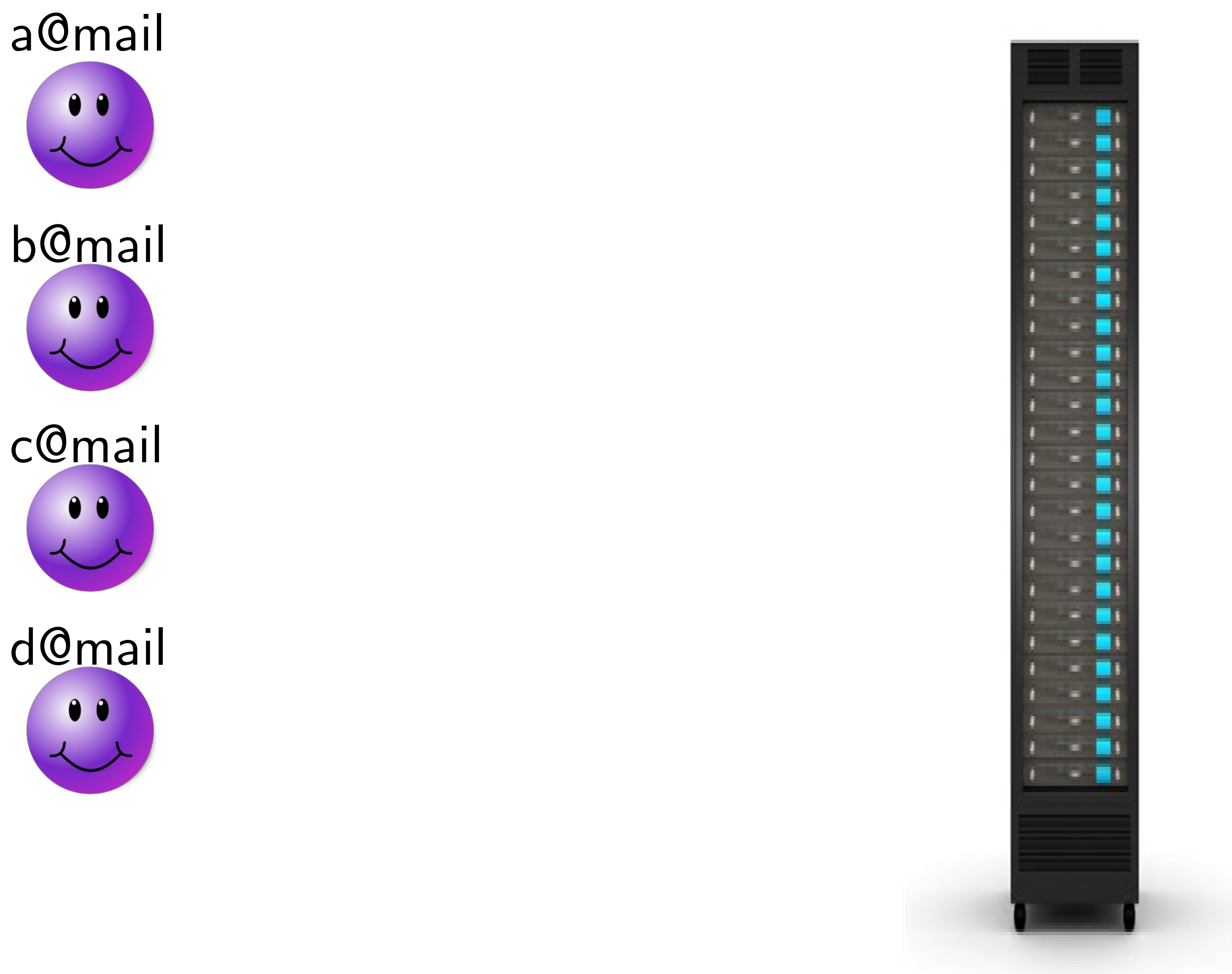
Drawback: constrained keys are obfuscated circuits

This work: constrained keys are short signatures

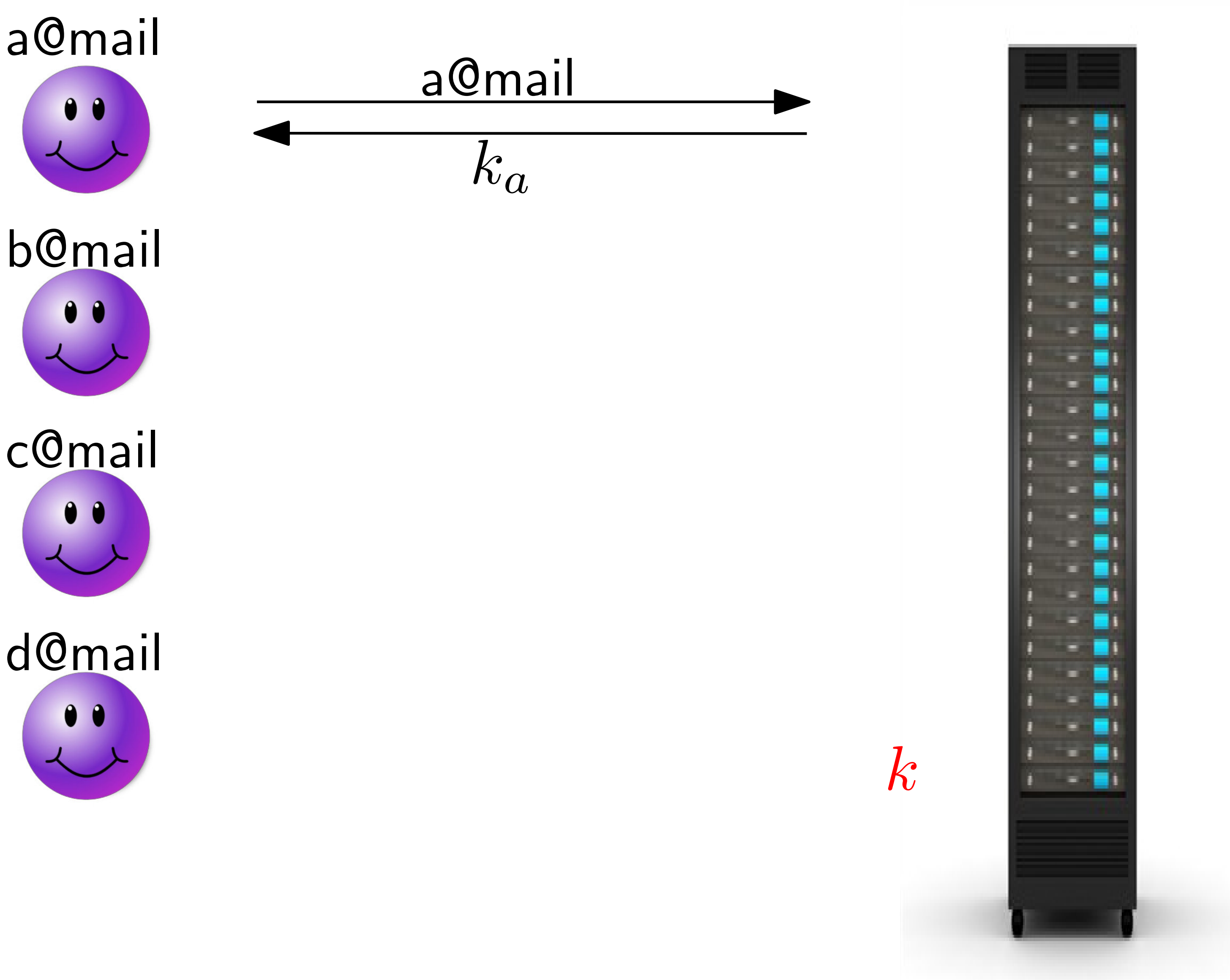
ENS

Application

Identity-Based Non-interactive Key Exchange

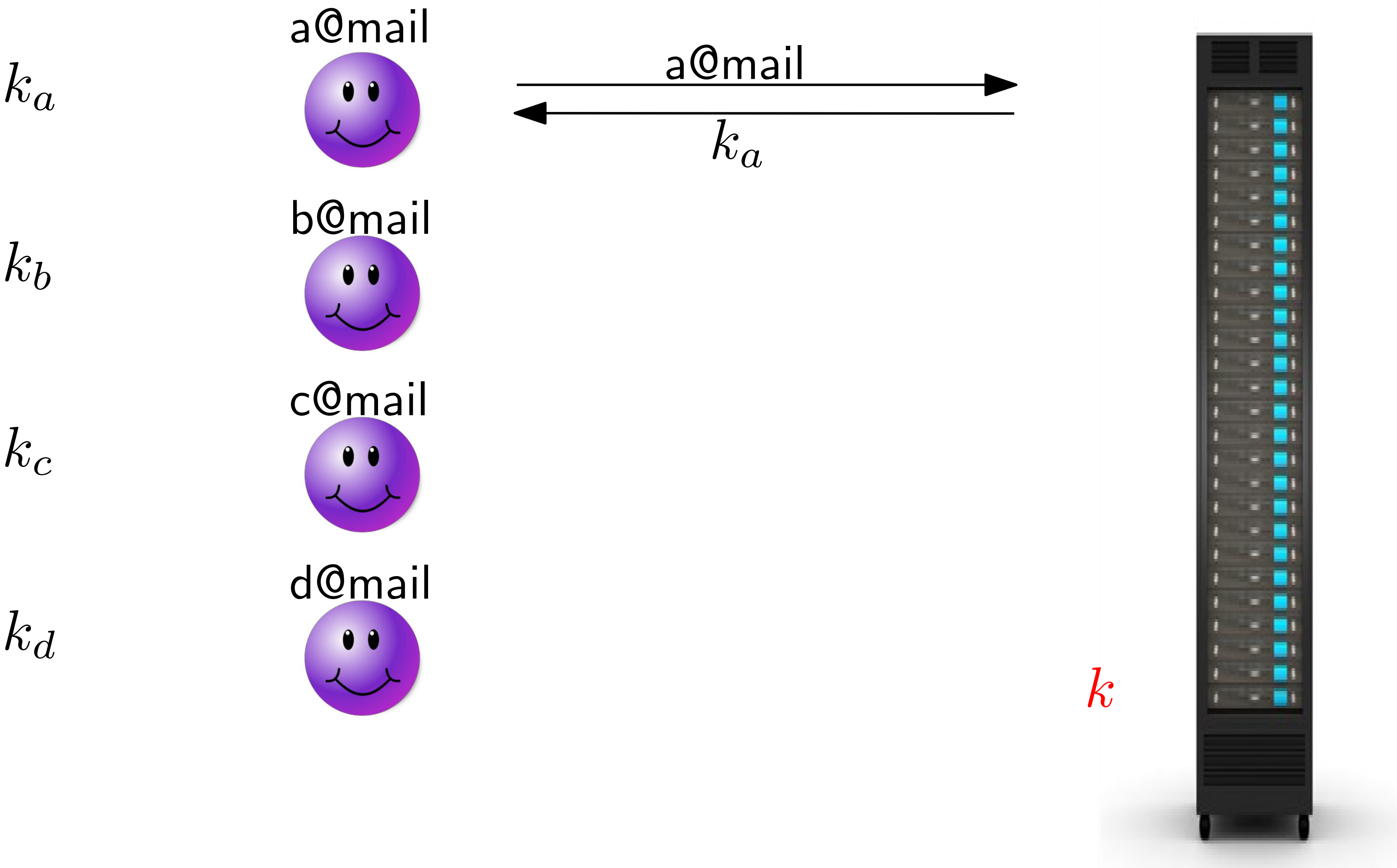


Identity-Based Non-interactive Key Exchange



ENS

Identity-Based Non-interactive Key Exchange



Identity-Based Non-interactive Key Exchange

k_a , k_{ac}

a@mail

k_b

b@mail

k_c , k_{ac}

c@mail

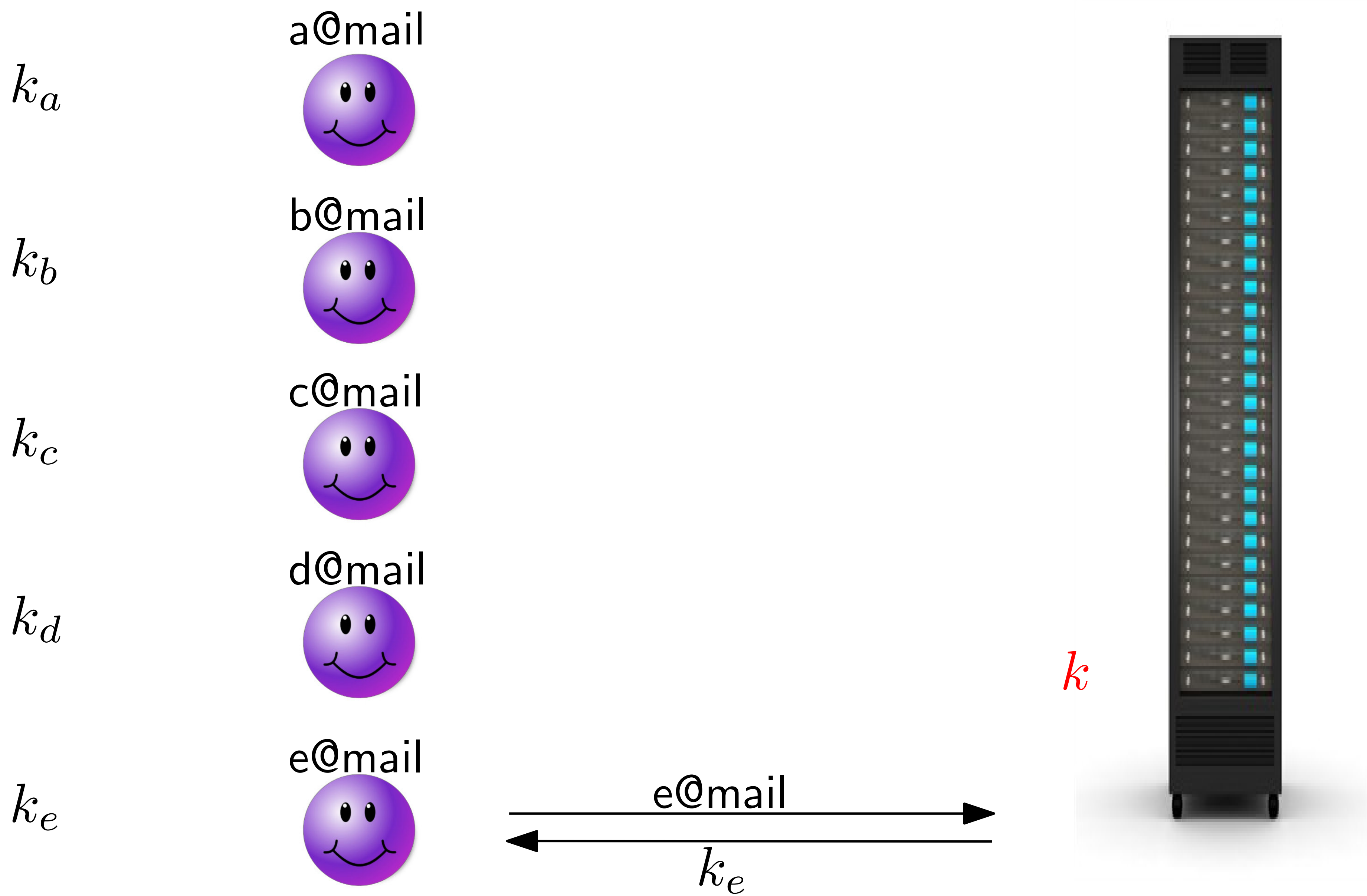
k_d

d@mail

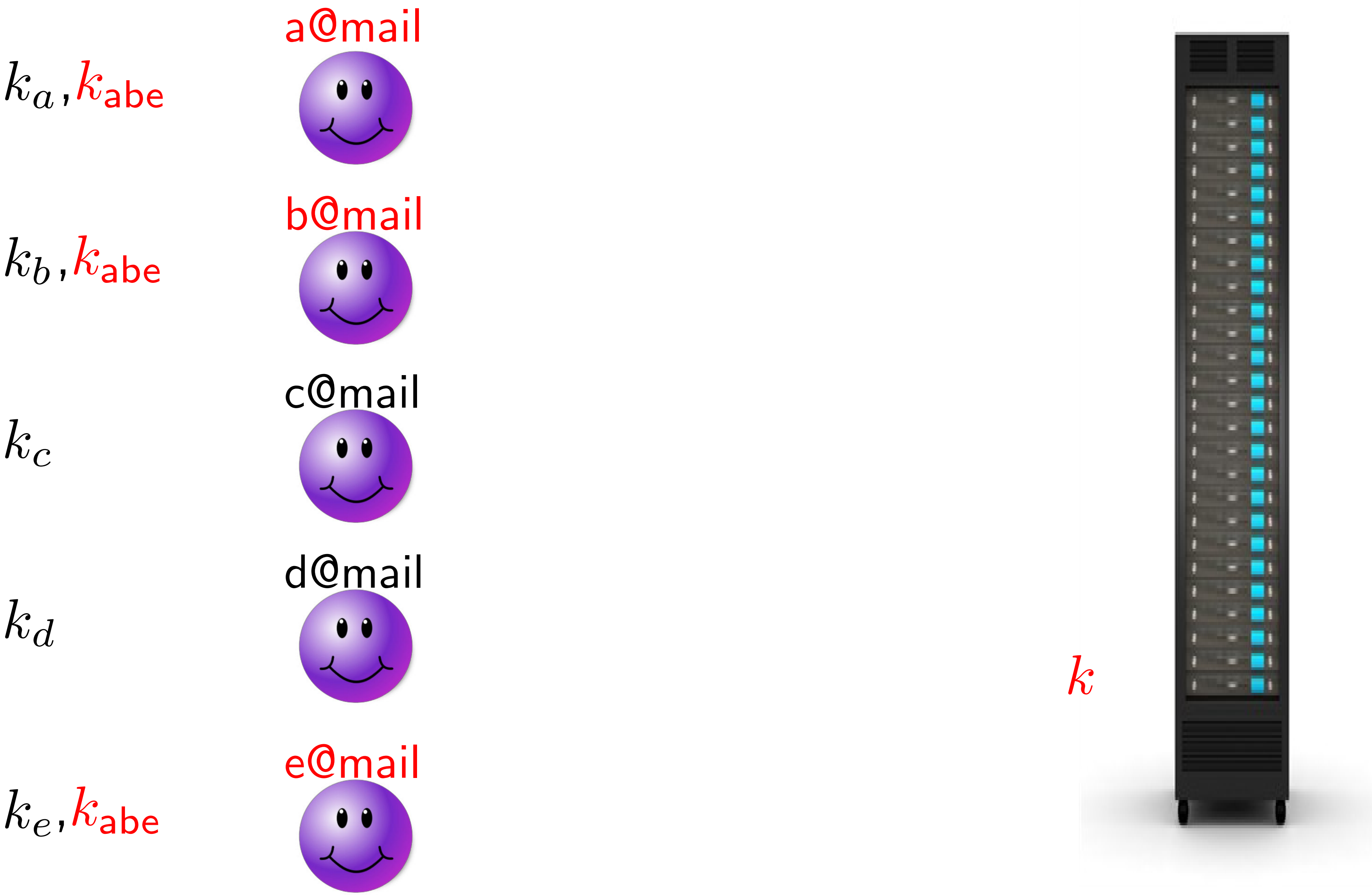
k

ENS

Identity-Based Non-interactive Key Exchange



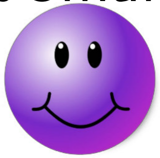
Identity-Based Non-interactive Key Exchange



Identity-Based Non-interactive Key Exchange

a@mail

b@mail



$\mathsf{F}_{k} : \{0,1\}^* \rightarrow \{0,1\}^m$

k

Identity-Based Non-interactive Key Exchange

a@mail

b@mail

$\mathsf{F}_{k} : \{0,1\}^* \to \{0,1\}^m$

$k_{\mathsf{abe}} :=$
$\mathsf{F}_{k}(\mathsf{a@mail} \| \mathsf{b@mail} \| \mathsf{e@mail})$

k

Identity-Based Non-interactive Key Exchange

$$\mathsf{F}_{k} : \{0,1\}^* \to \{0,1\}^m$$

b@mail

b@mail →

k

Identity-Based Non-interactive Key Exchange

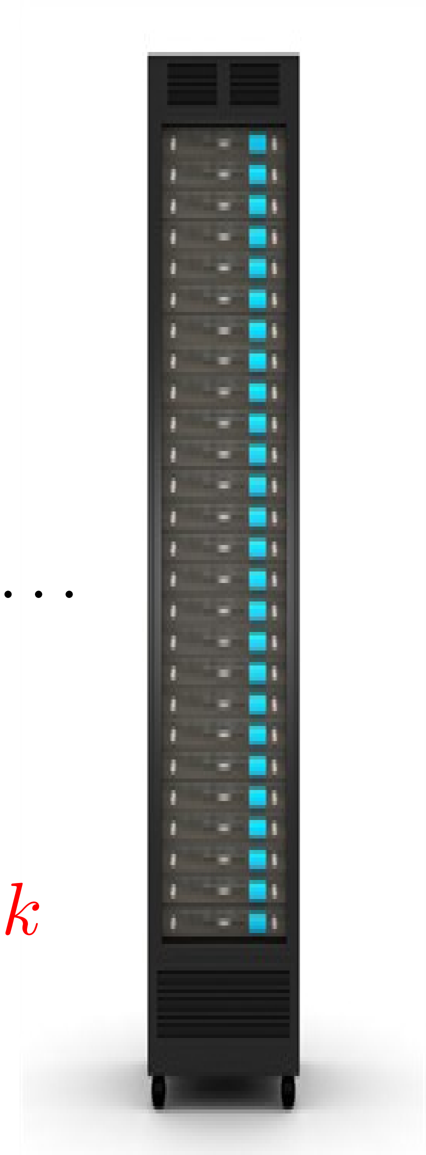
$$\mathsf{F}_{\color{red}k} : \{0,1\}^* \rightarrow \{0,1\}^m$$

b@mail

b@mail →

$$M_{\mathsf{b}}(x) = \begin{cases} 1 & \text{if } x = \dots \text{ b@mail } \dots \\ 0 & \text{otherwise} \end{cases}$$

$\color{red}k$

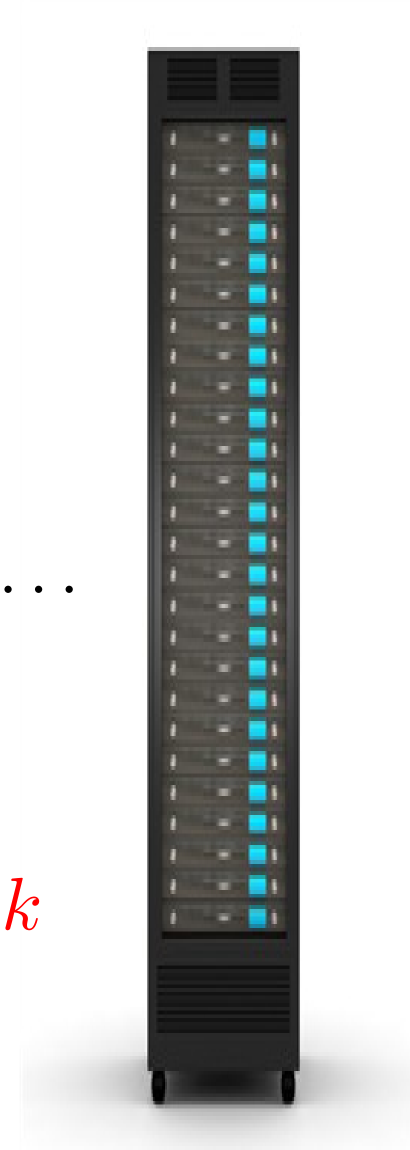


Identity-Based Non-interactive Key Exchange

$\mathsf{F}_{\color{red}k} : \{0,1\}^* \rightarrow \{0,1\}^m$

b@mail

b@mail →



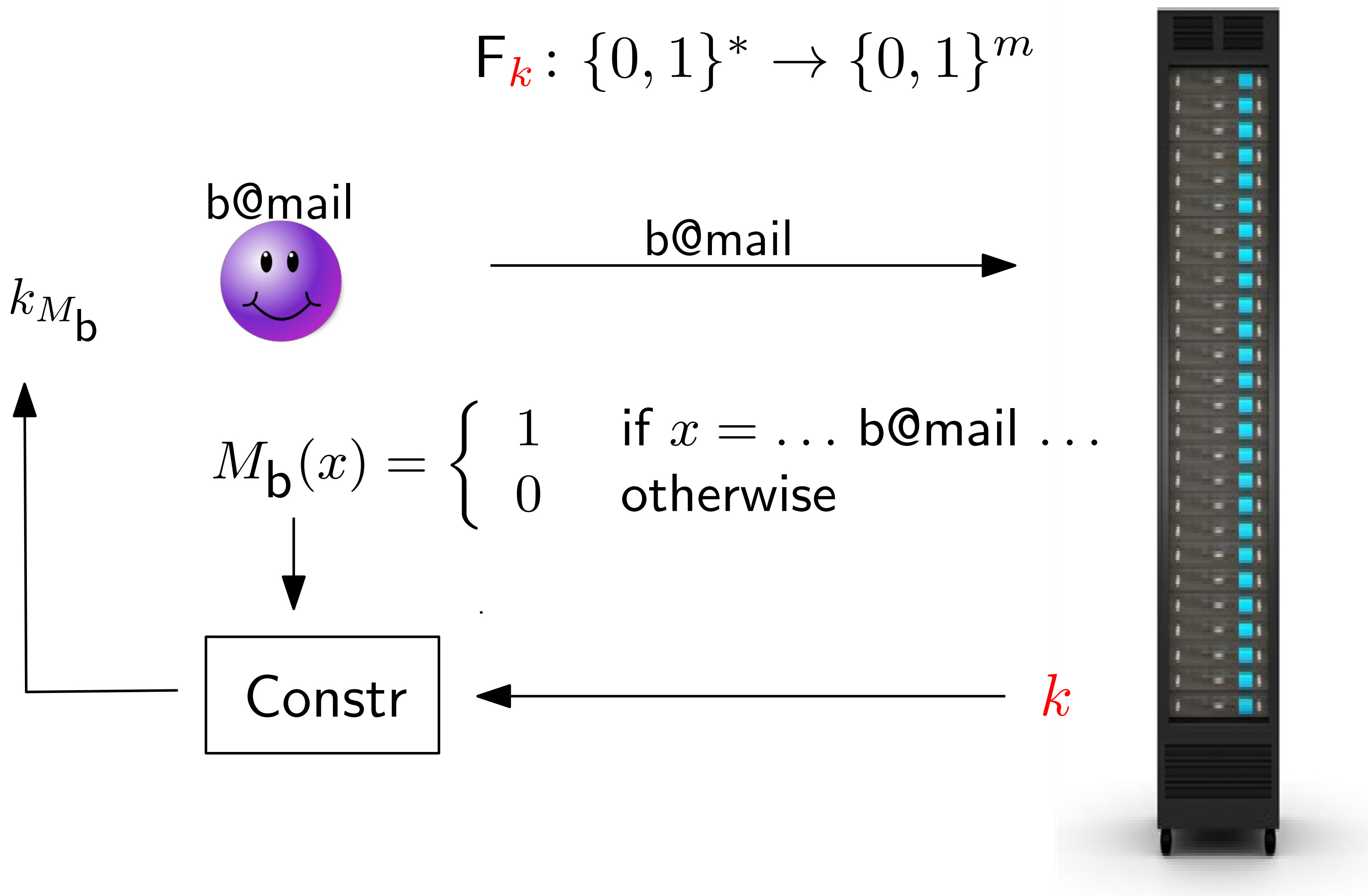
$$M_{\mathsf{b}}(x) = \begin{cases} 1 & \text{if } x = \ldots \text{ b@mail } \ldots \\ 0 & \text{otherwise} \end{cases}$$

↓

Constr ← $\color{red}k$

ENS

Identity-Based Non-interactive Key Exchange



Obfuscation

Application

Obfuscation

Program Obfuscation

Virtual black-box
[BGI+01]

Differing-input
[BGI+01],[BCP14]

Public-coin differing-input
[ISP15]

Indistinguishability
[BGI+01], [GGH+13]

Program Obfuscation

Virtual black-box
[BGI+01]

Impossible
[BGI+01]

Differing-input
[BGI+01],[BCP14]

Implausible
[GGH+14]

TM-impossible
[BSW16]

Public-coin differing-input
[ISP15]

Indistinguishability
[BGI+01], [GGH+13]

ENS

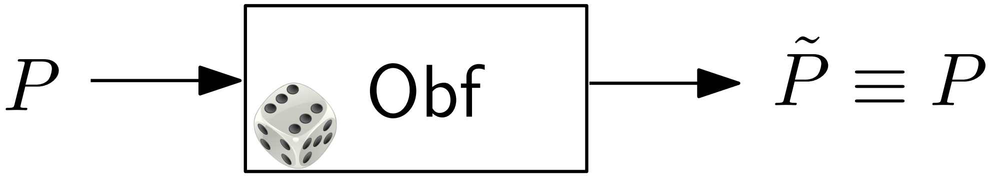
Program Obfuscation

Functionality:

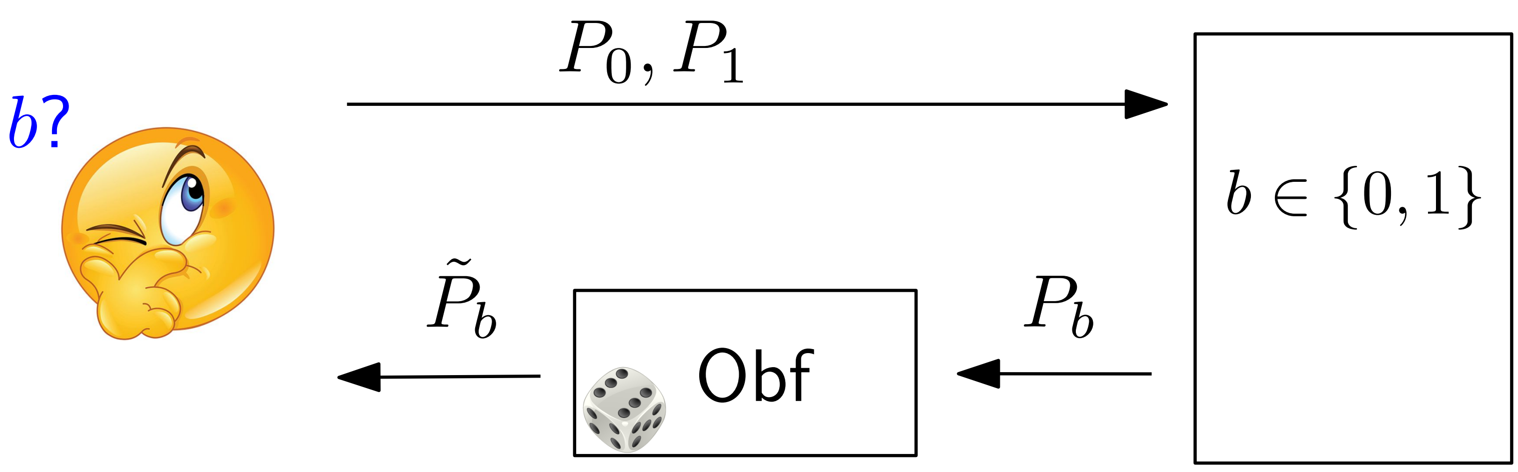
$$(\forall x : \tilde{P}(x) = P(x))$$

Program Obfuscation

Functionality:

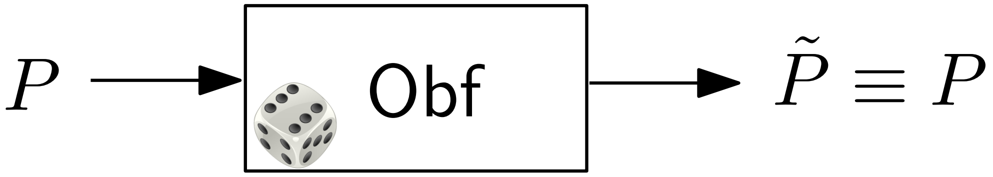


Security:



Program Obfuscation

Functionality:

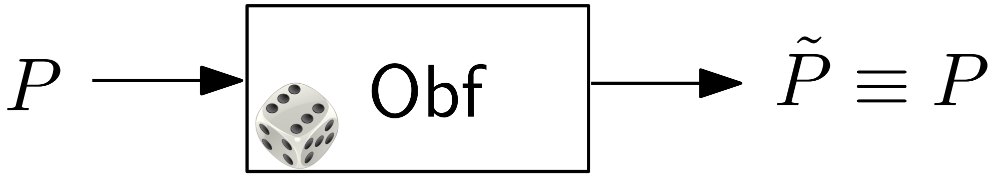


Security:

- diO: must be hard to find x: $P_0(x) \neq P_1(x)$

Program Obfuscation

Functionality:



Security:

- diO: must be hard to find x: $P_0(x) \neq P_1(x)$
- public-coin diO:

  hard to find x: $P_0(x) \neq P_1(x)$

  even when given coins for computing P_0, P_1

Program Obfuscation

Functionality:

Security:

- diO: must be hard to find x: $P_0(x) \neq P_1(x)$
- public-coin diO:
  hard to find x: $P_0(x) \neq P_1(x)$
  even when given coins for computing P_0, P_1
- iO: $P_0 \equiv P_1$

Obfuscation

Constructions

ENS

Constructions

TM CPRFs

1) Warm-up: A circuit CPRF assuming

- Puncturable PRFs
- iO

TM CPRFs

1) Warm-up: A circuit CPRF assuming

- Puncturable PRFs
- iO

2) A Turing-machine CPRF assuming

- Puncturable PRFs
- Public-coin diO
- SNARKs (Succinct non-interactive arguments of knowledge)

TM CPRFs

1) Warm-up: A circuit CPRF assuming

- Puncturable PRFs
- iO

2) A Turing-machine CPRF assuming

- Puncturable PRFs
- Public-coin diO
- SNARKs (Succinct non-interactive arguments of knowledge)

3) A TM CPRF with short keys

- assuming the same

ENS

A Circuit CPRF

- puncturable PRF PF_k
- iO

Circuit-constrained PRF:

- $\mathsf{F}_k(x) := \mathsf{PF}_k(x)$

A Circuit CPRF

- puncturable PRF PF_k
- iO

Circuit-constrained PRF:

- $\mathsf{F}_k(x) := \mathsf{PF}_k(x)$
- $\mathsf{Constr}(k, C) \to k_C$:

$$P_{k,C}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if } C(x) = 1 \\ \bot & \text{otherwise} \end{cases}$$

A Circuit CPRF

- puncturable PRF PF_k
- iO

Circuit-constrained PRF:

- $\mathsf{F}_k(x) := \mathsf{PF}_k(x)$
- $\mathsf{Constr}(k, C) \to k_C$:

$$k_C \leftarrow \mathsf{iO}\left(P_{k,C}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if } C(x) = 1 \\ \perp & \text{otherwise} \end{cases} \right)$$

ENS

A Circuit CPRF

- puncturable PRF PF_k
- iO

Circuit-constrained PRF:

- $\mathsf{F}_k(x) := \mathsf{PF}_k(x)$
- $\mathsf{Constr}(k, C) \to k_C$:

$$k_C \leftarrow \mathsf{iO}\left(P_{k,C}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if } C(x) = 1 \\ \perp & \text{otherwise} \end{cases} \right)$$

Theorem. F is a secure circuit CPRF.

A Circuit CPRF

- puncturable PRF PF_k
- iO

Circuit-constrained PRF:

- $\mathsf{F}_k(x) := \mathsf{PF}_{\mathbf{k_{x^*}}}(x)$
- $\mathsf{Constr}(k, C) \to k_C$:

$$k_C \leftarrow \mathsf{iO}\left(P_{k,C}(x) := \begin{cases} \mathsf{PF}_{\mathbf{k_{x^*}}}(x) & \text{if } C(x) = 1 \\ \perp & \text{otherwise} \end{cases} \right)$$

Theorem. F is a secure circuit CPRF.

ENS

A CPRF for **Unbounded** Inputs

- $\mathsf{F}_k(x) := \mathsf{PF}_k(x)$
- $\mathsf{Constr}(k, C)$:

$$k_C \leftarrow \mathsf{iO}\left(P_{k,C}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if } C(x) = 1 \\ \perp & \text{otherwise} \end{cases} \right)$$

ENS

A CPRF for **Unbounded** Inputs

- $\mathsf{F}_k(x) := \mathsf{PF}_k(x)$
- $\mathsf{Constr}(k, C)$:

$$k_C \leftarrow \mathsf{iO}\left(P_{k,C}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if } C(x) = 1 \\ \perp & \text{otherwise} \end{cases} \right)$$

M?

ENS

A CPRF for **Unbounded** Inputs

- $\mathsf{F}_k(x) := \mathsf{PF}_k(x)$
- $\mathsf{Constr}(k, C)$:

$$k_C \leftarrow \textcolor{blue}{\mathsf{iO}}\left(P_{k,C}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if } C(x) = 1 \\ \perp & \text{otherwise} \end{cases} \right)$$

M?

Circuit!

A CPRF for **Unbounded** Inputs

- $\mathsf{F}_k(x) := \mathsf{PF}_k(x)$
- $\mathsf{Constr}(k, C)$:

$$k_C \leftarrow \mathsf{iO}\left(P_{k,C}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if } C(x) = 1 \\ \perp & \text{otherwise} \end{cases} \right)$$

M?

Circuit!

- cannot run TM

  $\Rightarrow$ use SNARK proving $M(x) = 1$

A CPRF for **Unbounded** Inputs

- $\mathsf{F}_k(x) := \mathsf{PF}_k(x)$
- $\mathsf{Constr}(k, C)$:

$$k_C \leftarrow \mathsf{iO}\left(P_{k,C}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if } C(x) = 1 \\ \perp & \text{otherwise} \end{cases} \right)$$

unbounded!

Circuit!

- cannot run TM
  $\Rightarrow$ use SNARK proving $M(x) = 1$
- does not accept $x \in \{0,1\}^*$
  $\Rightarrow$ hash x

ENS

A CPRF for **Unbounded** Inputs

- $\mathsf{F}_k(x) := \mathsf{PF}_k(H(x))$ with $H\colon \{0,1\}^* \to \{0,1\}^n$

A CPRF for **Unbounded** Inputs

- $\mathsf{F}_k(x) := \mathsf{PF}_k(H(x))$
- $\mathsf{Constr}(k, M)$:

$$P_{k,M}(h, \pi) = \begin{cases} \mathsf{PF}_k(h) & \text{if } \pi \text{ proves } \exists x : H(x) = h \\ & \qquad \wedge\ M(x) = 1 \\ \bot & \text{otherwise} \end{cases}$$

A CPRF for **Unbounded** Inputs

- $\mathsf{F}_k(x) := \mathsf{PF}_k(H(x))$
- $\mathsf{Constr}(k, M)$:

$$k_M \leftarrow \mathsf{diO}\left(P_{k,M}(h, \pi) = \begin{cases} \mathsf{PF}_k(h) & \text{if } \pi \text{ proves } \exists x : H(x) = h \\ & \qquad\qquad \wedge\, M(x) = 1 \\ \bot & \text{otherwise} \end{cases}\right)$$

A CPRF for **Unbounded** Inputs

- $\mathsf{F}_k(x) := \mathsf{PF}_k(H(x))$
- $\mathsf{Constr}(k, M)$:

$$k_M \leftarrow \mathsf{diO}\left(P_{k,M}(h,\pi) = \begin{cases} \mathsf{PF}_k(h) & \text{if } \pi \text{ proves } \exists x : H(x) = h \\ & \qquad\qquad \wedge\ M(x) = 1 \\ \bot & \text{otherwise} \end{cases}\right)$$

Why diO?

- consider M with $\quad M(x^*) = 0 \quad M(x') = 1 \qquad H(x^*) = H(x')$

$$\Rightarrow P_{k,M} \not\equiv P_{k_{x^*},M}$$

ENS

A CPRF for Unbounded Inputs **with Short Keys**

- $\mathsf{F}_k(x) := \mathsf{PF}_k(H(x))$
- $\mathsf{Constr}(k, M)$: signature σ on M

A CPRF for Unbounded Inputs **with Short Keys**

- $\mathsf{F}_k(x) := \mathsf{PF}_k(H(x))$
- $\mathsf{Constr}(k, M)$: signature σ on M
- public:

$$\mathsf{diO}\left(P_{k,vk}(M, h, \pi, \sigma) = \begin{cases} \mathsf{PF}_k(h) & \text{if } \pi \text{ proves } \exists x : H(x) = h \\ & \qquad \wedge\ M(x) = 1 \\ & \text{and } \mathsf{Verify}(vk, M, \sigma) \\ \perp & \text{otherwise} \end{cases}\right)$$

A CPRF for Unbounded Inputs **with Short Keys**

- $\mathsf{F}_k(x) := \mathsf{PF}_k(H(x))$
- $\mathsf{Constr}(k, M)$: signature σ on M
- public:

$$\mathsf{diO}\left(P_{k,vk}(M, h, \pi, \sigma) = \begin{cases} \mathsf{PF}_k(h) & \text{if } \pi \text{ proves } \exists x : H(x) = h \\ & \qquad \wedge\ M(x) = 1 \\ & \text{and } \mathsf{Verify}(vk, M, \sigma) \\ \perp & \text{otherwise} \end{cases}\right)$$

Security:

- $\mathsf{diO}(P_{k,vk}) \approx \mathsf{diO}(P_{k_{x^*},vk})$

ENS

A CPRF for Unbounded Inputs **with Short Keys**

- $\mathsf{F}_k(x) := \mathsf{PF}_k(H(x))$
- $\mathsf{Constr}(k, M)$: signature σ on M
- public:

$$\mathsf{diO}\left(P_{k,vk}(M, h, \pi, \sigma) = \begin{cases} \mathsf{PF}_k(h) & \text{if } \pi \text{ proves } \exists x : H(x) = h \\ & \qquad \wedge\ M(x) = 1 \\ & \text{and } \mathsf{Verify}(vk, M, \sigma) \\ \bot & \text{otherwise} \end{cases}\right)$$

Security:

- $\mathsf{diO}(P_{k,vk}) \approx \mathsf{diO}(P_{k_{x^*},vk})$
- Differing inputs? Yes: σ on M with $M(x^*) = 1$

A CPRF for Unbounded Inputs **with Short Keys**

- $\mathsf{F}_k(x) := \mathsf{PF}_k(H(x))$
- $\mathsf{Constr}(k, M)$: signature σ on M
- public:

$$\mathsf{diO}\left(P_{k,vk}(M, h, \pi, \sigma) = \begin{cases} \mathsf{PF}_k(h) & \text{if } \pi \text{ proves } \exists x : H(x) = h \\ & \qquad \wedge\ M(x) = 1 \\ & \text{and } \mathsf{Verify}(vk, M, \sigma) \\ \bot & \text{otherwise} \end{cases}\right)$$

Security:

- $\mathsf{diO}(P_{k,vk}) \approx \mathsf{diO}(P_{k_{x^*},vk})$
- Differing inputs? Yes: σ on M with $M(x^*) = 1$
- Hard to find when given coins? No! can reconstruct signing key

ENS

Functional Signatures w/ Obliv. Samplable Keys

Signature scheme with sampling algorithm

$$(vk^*, sk_{x^*}) \leftarrow \mathsf{OSmp}(x^*; r)$$

Functional Signatures w/ Obliv. Samplable Keys

Signature scheme with sampling algorithm

$$(vk^*, sk_{x^*}) \leftarrow \mathsf{OSmp}(x^*; r)$$

such that:

- $vk^* \approx vk$

Functional Signatures w/ Obliv. Samplable Keys

Signature scheme with sampling algorithm

$$(vk^*, sk_{x^*}) \leftarrow \mathsf{OSmp}(x^*; r)$$

such that:

- $vk^* \approx vk$
- sk_{x^*} allows signing M's with $M(x^*) = 0$

Functional Signatures w/ Obliv. Samplable Keys

Signature scheme with sampling algorithm

$$(vk^*, sk_{x^*}) \leftarrow \mathsf{OSmp}(x^*; r)$$

such that:

- $vk^* \approx vk$
- sk_{x^*} allows signing M's with $M(x^*) = 0$
- given r, hard to forge σ on M with $M(x^*) = 1$

ENS

Functional Signatures w/ Obliv. Samplable Keys

Signature scheme with sampling algorithm

$$(vk^*, sk_{x^*}) \leftarrow \mathsf{OSmp}(x^*; r)$$

such that:

- $vk^* \approx vk$
- sk_{x^*} allows signing M's with $M(x^*) = 0$ → Reduction can answer Constr queries
- given r, hard to forge σ on M with $M(x^*) = 1$

Functional Signatures w/ Obliv. Samplable Keys

Signature scheme with sampling algorithm

$$(vk^*, sk_{x^*}) \leftarrow \mathsf{OSmp}(x^*; r)$$

such that:

- $vk^* \approx vk$
- sk_{x^*} allows signing M's with $M(x^*) = 0$
- given r, hard to forge σ on M with $M(x^*) = 1$

  → – hard to find differing input
  – apply diO

Functional Signatures w/ Obliv. Samplable Keys

Signature scheme with sampling algorithm

$$(vk^*, sk_{x^*}) \leftarrow \mathsf{OSmp}(x^*; r)$$

such that:

- $vk^* \approx vk$
- sk_{x^*} allows signing M's with $M(x^*) = 0$
- given r, hard to forge σ on M with $M(x^*) = 1$

Construction from: commitment, PRF, SNARK

ENS

Functional Signatures w/ Obliv. Samplable Keys

Signature scheme with sampling algorithm

$$(vk^*, sk_{x^*}) \leftarrow \mathsf{OSmp}(x^*; r)$$

such that:

Thank you!

- $vk^* \approx vk$
- sk_{x^*} allows signing M's with $M(x^*) = 0$
- given r, hard to forge σ on M with $M(x^*) = 1$

Construction from: commitment, PRF, SNARK

(and thanks to Hamza Abusalah for letting me reuse his slides)

ENS