MITSUBISHI ELECTRIC
Changes for the Better

家庭から宇宙まで、エコチェンジ。 eco Changes

Replacing SHA-2 with SHA-3 Enhances Generic Security of HMAC

Yusuke Naito

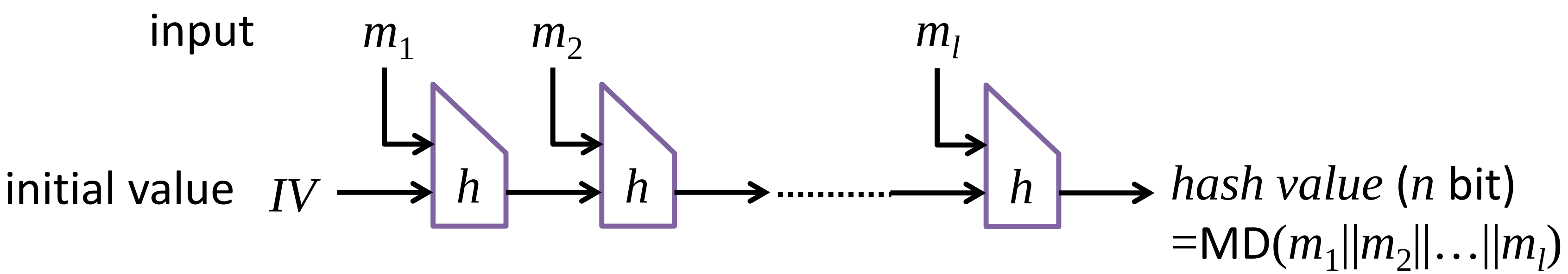
Mitsubishi Electric Corporation

Lei Wang

Shanghai Jiao Tong University

SHA-2

- FIPS 180-4
- Inputs: arbitrary length
- Outputs: 224 bit, 256 bit, 384 bit, 512 bit
- Use Merkle-Damgard (MD) construction
  - Iterates a compression function h



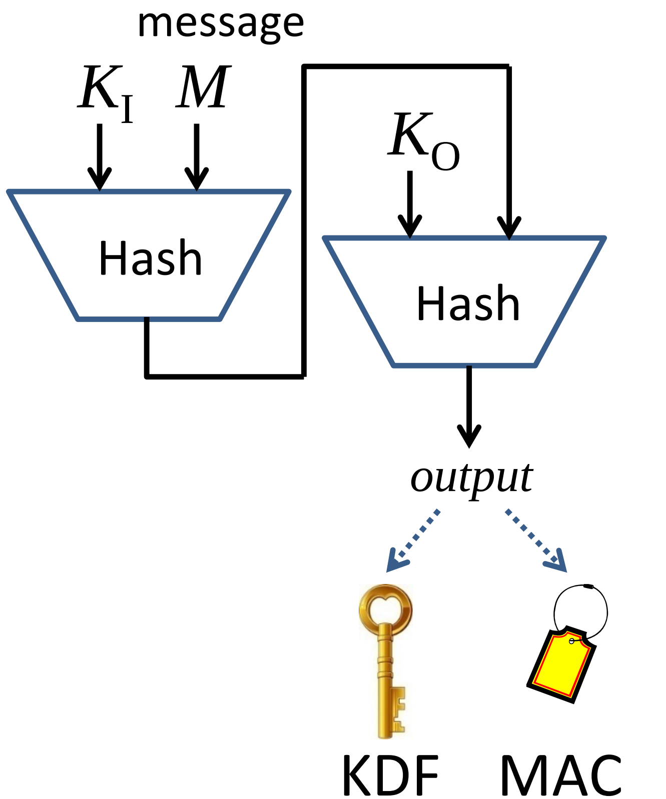
- Hash functions with MD are vulnerable to the length extension attack
- HMAC was designed to convert the hash function with MD into a secure keyed hash function

HMAC

- FIPS standard keyed hash (FIPS 198-1)
- Call a hash function two times
- Used as
  - Key derivation function (KDF)
  - Message authentication code (MAC)
- Widely used in e.g.,
  - SSL, SSH, IPSec, TLS, IKE, etc

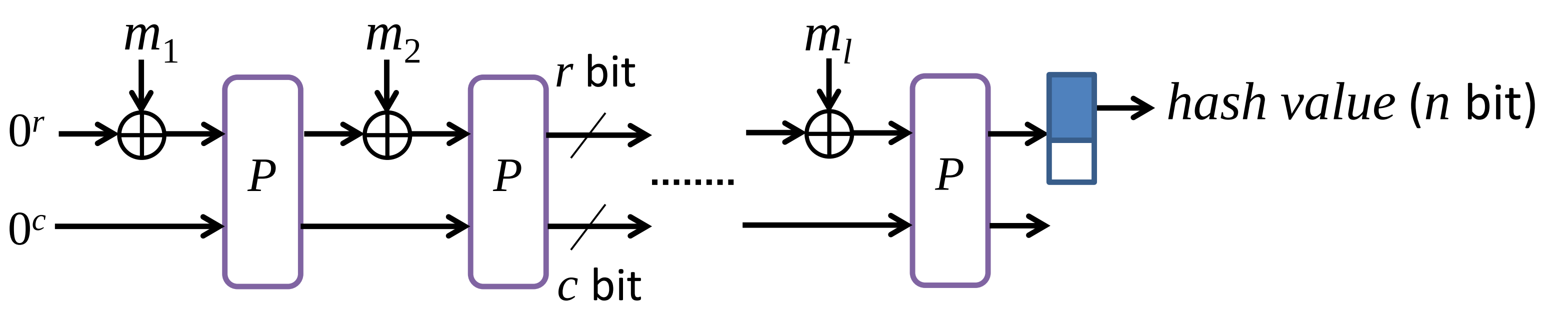
Secret key

$$K_{\mathrm{I}} = K \oplus \mathrm{ipad} \qquad K_{\mathrm{O}} = K \oplus \mathrm{opad}$$



SHA-3

- Standardized at FIPS 202 (Aug. 2015)
- Same interface as SHA-2
  - Inputs: arbitrary length
  - Outputs: 224-bit, 256-bit, 384-bit, 512-bit
- Use the sponge construction
  - Iterate a permutation $P:\{0,1\}^{r+c} \rightarrow \{0,1\}^{r+c}$



Motivation

- FIPS 202

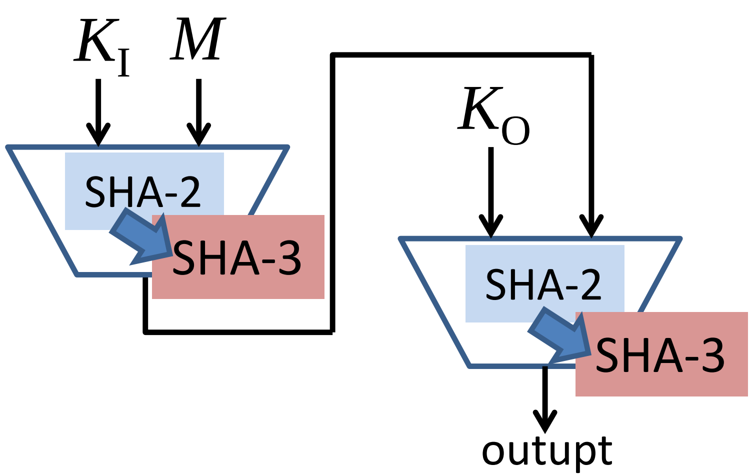
page 22

> SHA3-224, SHA3-256, SHA3-384, and SHA3-512 are approved cryptographic hash functions. One of the approved uses of cryptographic hash functions occurs within the Keyed-Hash Message Authentication Code (HMAC). The input block size in bytes, denoted by *B* in the HMAC specification [10], is given in Table 3 below for the SHA-3 hash functions[5]:

page 24

> The four SHA-3 hash functions are alternatives to the SHA-2 functions, and they are designed to

- SHA-2 may be replaced with SHA-3 in HMAC



Question

Is there an advantage of replacing SHA-2 with SHA-3 in HMAC?

Security

- PRF-security
- MAC-security (Unforgeability)

Generic Security

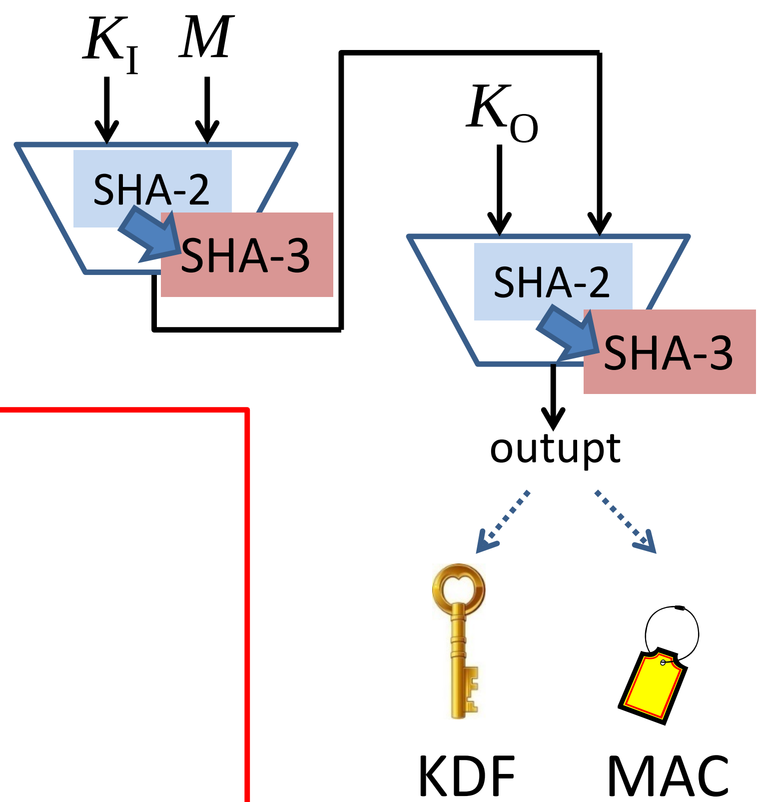
Assume that the underlying primitive has no structural fraw, i.e.,

- SHA-2 Case

➡ <u>HMAC-MD using random oracle h</u>

- SHA-3 Case

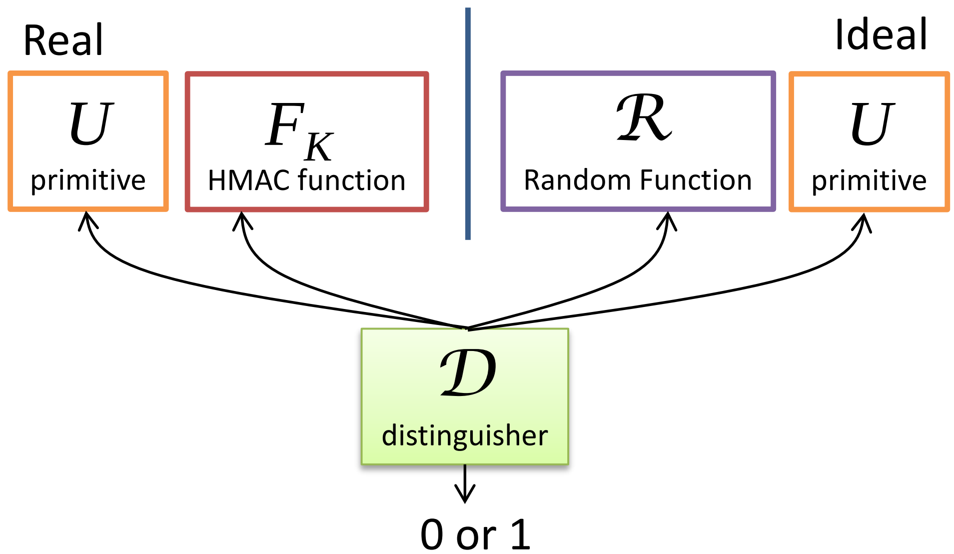
➡ <u>HMAC-Sponge using random permutation P</u>



Our Result

- Security of HMAC-MD (using random oracle h)
  - Proven in previous works
- Security of HMAC-Sponge (using random permutation P)
  - Not proven
- This paper
  - ✓ Prove the PRF- and MAC-security of HMAC-Sponge
  - ✓ Compare HMAC-Sponge with HMAC-MD in terms of the PRF- and MAC-security
  - ✓ Conclude that replacing SHA-2 with SHA-3 enhances the generic security of HMAC

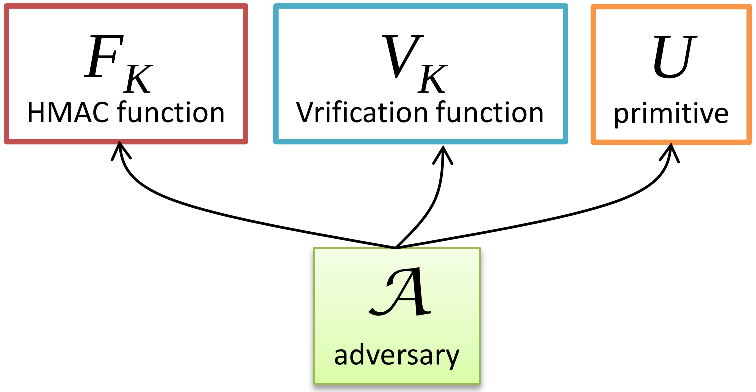
PRF-Security



- Indistinguishability between the real world and the ideal world
- A distinguisher $\mathcal{D}$ interacts with
  - (F_K, U) in the real world
  - $(\mathcal{R}, U)$ in the ideal world
- The advantage function is defined as

$$\mathrm{Adv}^{\mathrm{PRF}}_{\mathrm{HMAC}}(\mathcal{D}) := \Pr[\mathcal{D}=1 \text{ in the real world}] - \Pr[\mathcal{D}=1 \text{ in the ideal world}]$$

MAC-Security

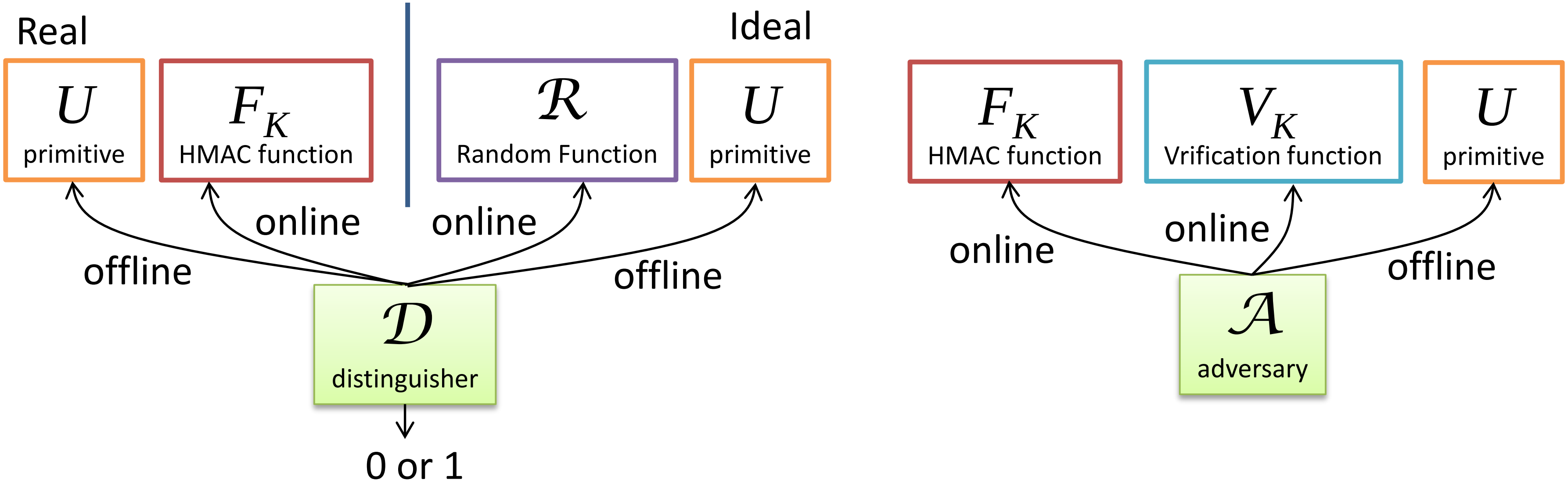


- An adversary $\mathcal{A}$ can interacts with (F_K, V_K ,U)
- Verification function V_K
  - accept a pair (M, tag)
  - check the equality $F_K(M) = tag$
  - return **accept** if the equality holds, and return **reject** otherwise
- $\mathcal{A}$ cannot make a trivial query (M, tag) to V_K,
  that is, M has not been queried to F_K
- The advantage function is defined as
  $\mathrm{Adv}^{\mathrm{mac}}_{\mathrm{HMAC}}(\mathcal{A})$:= Pr[$\mathcal{A}$ makes a query to V_K s.t. **accept** is returned]

Security Parameters

PRF-Security

MAC-Security

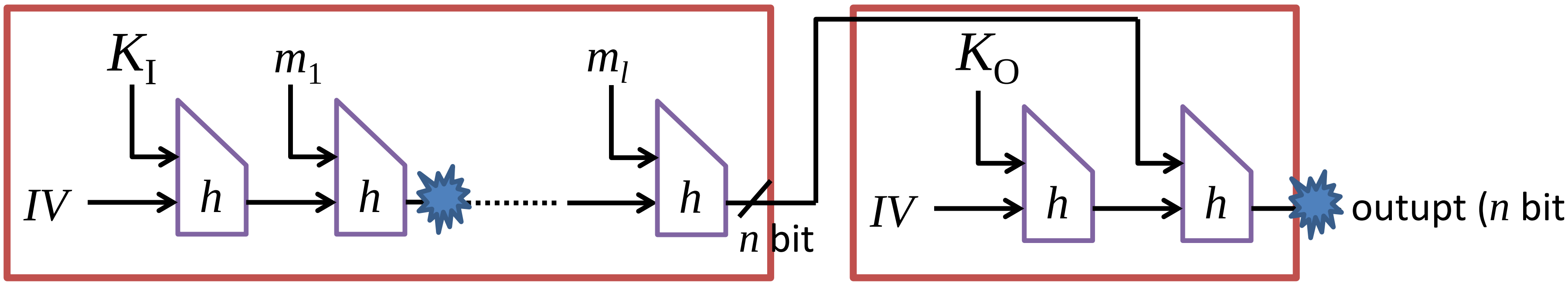


Security Parameters

- n: hash size
- Q: number of offline queries (primitive queries)
- q: number of online queries (construction queries)
- ℓ: maximum input length in blocks to HMAC

PRF- and MAC-Security of HMAC-MD

HMAC-MD



- The following bounds were proven.
  - $\forall \mathcal{D}$: $\mathrm{Adv}^{\mathrm{PRF}}_{\mathrm{HMAC\text{-}MD}}(\mathcal{D}) \leqq O(\ell q^2/2^n)$
  - $\forall \mathcal{A}$: $\mathrm{Adv}^{\mathrm{mac}}_{\mathrm{HMAC\text{-}MD}}(\mathcal{A}) \leqq O(\ell q^2/2^n)$

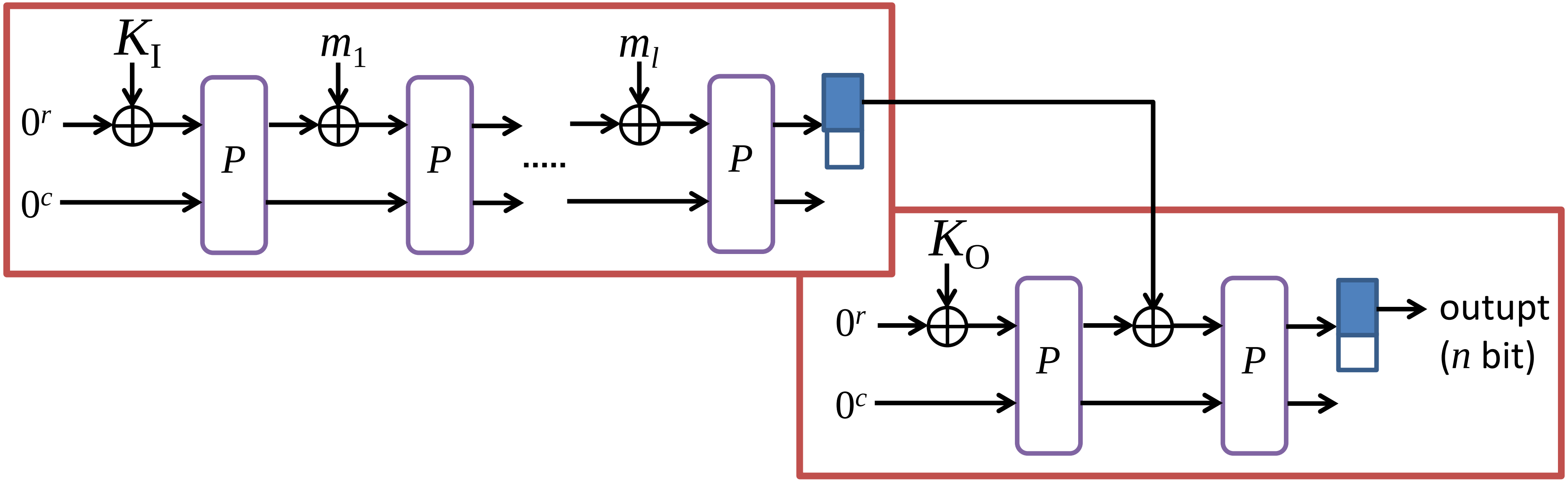
$\ell \times \underbrace{q^2/2^n}$

Collision in
n-bit internal states

- HMAC-MD is PRF- and MAC-secure up to $q = O(2^{n/2}/\ell^{1/2})$

PRF- and MAC-Security of HMAC-Sponge

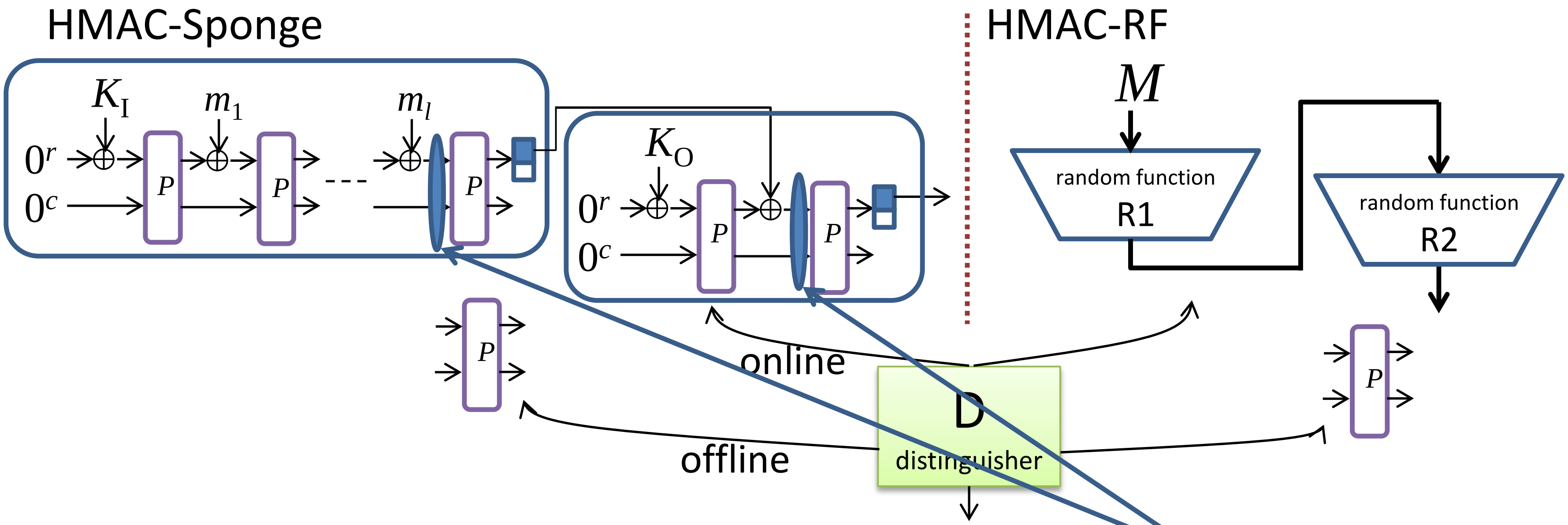
- HMAC-Sponge



- We prove that
  - $\forall \mathcal{D}$: $\mathrm{Adv}^{\mathrm{PRF}}_{\mathrm{HMAC\text{-}Sponge}}(\mathcal{D}) \leqq O(q^2/2^n + (\ell q)^2/2^{r+c} + \ell q Q/2^{r+c})$
  - $\forall \mathcal{A}$: $\mathrm{Adv}^{\mathrm{mac}}_{\mathrm{HMAC\text{-}Sponge}}(\mathcal{A}) \leqq O(nq/2^n + (\ell q)^2/2^{r+c} + \ell q Q/2^{r+c})$

Step 1: Ind. of HMAC-Sponge from HMAC-RF

Step 1



- The outputs of Sponge in HMAC are randomly drawn if the inputs are new
- If the inputs are not new then one of the following events occurs
  - Collision in inputs to P in HMAC-Sponge : $O((\ell q)^2/2^{r+c})$
  - Collision in inputs to P between online and offline queries: $O(\ell qQ/2^{r+c})$
- Indistinguisiable prob.: $O((\ell q)^2/2^{r+c} + \ell qQ/2^{r+c})$

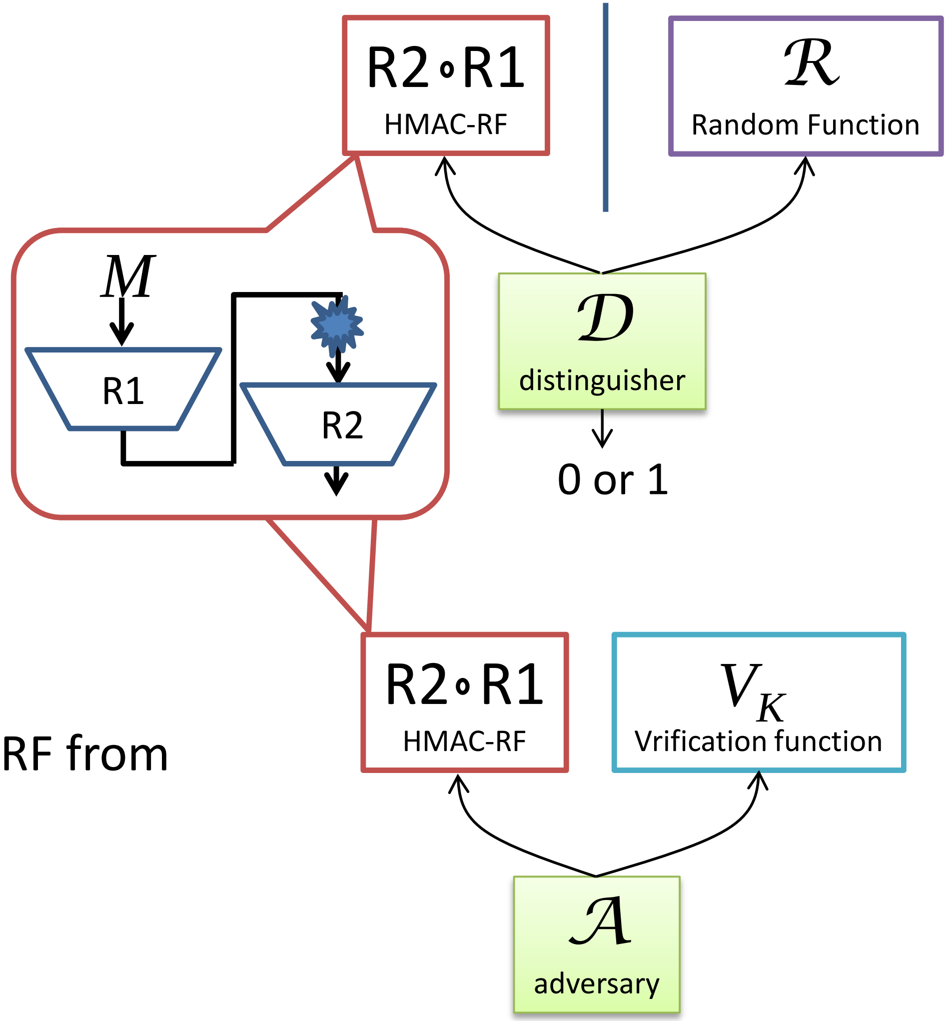
We can analyze the security of HMAC-Sponge by using HMAC-RF with the security loss $O((\ell q)^2/2^{r+c} + \ell qQ/2^{r+c})$

Step 2: The PRF- and MAC- Security of HMAC-RF

Step 2

PRF-Security of HMAC-RF

- If no collision occurs in then all outputs of HMAC-RF are randomly drawn
- PRF-adv≦collision prob. $O(q^2/2^n)$



MAC-Security of HMAC-RF

- By an n-multi-collision analysis in $\mathcal{A}$ needs to guess an output of HMAC-RF from at least $2^n/n$ output candidates
- MAC-adv≦ $O(q \times n/2^n)$

Step 3: Combining Step 1 and Step 2

Step 3

PRF-Security of HMAC-Sponge

In SHA-3, $2^n << 2^{r+c}$

- $\forall \mathcal{D}$: $\mathrm{Adv}^{\mathrm{PRF}}_{\text{HMAC-Sponge}}(\mathcal{D}) \leqq O(\underbrace{(\ell q)^2/2^{r+c} + \ell qQ/2^{r+c}}_{\text{from Step 1}} + \underbrace{q^2/2^n}_{\text{from Step 2}}) = O(q^2/2^n)$

HMAC-Sponge is PRF-secure up to $q = O(2^{n/2})$

MAC-Security of HMAC-Sponge

In SHA-3, $2^n << 2^{r+c}$

- $\forall \mathcal{D}$: $\mathrm{Adv}^{\mathrm{MAC}}_{\text{HMAC-Sponge}}(\mathcal{D}) \leqq O(\underbrace{(\ell q)^2/2^{r+c} + \ell qQ/2^{r+c}}_{\text{from Step 1}} + \underbrace{nq/2^n}_{\text{from Step 2}}) = O(nq/2^n)$

HMAC-Sponge is MAC-secure up to $q = O(2^n/n)$

Conclusion

- HMAC-MD is PRF- and MAC-secure up to $q = O(2^{n/2}/\ell^{1/2})$
- HMAC-Sponge is
  - PRF-secure up to $q = O(2^{n/2})$
  - MAC-secure up to $q = O(2^n/n)$
- HMAC-SHA-2 vs. HMAC-SHA-3

PRF-Security

Size n	HMAC-SHA-2 $O(\ell q^2/2^n)$	HMAC-SHA-3 $O(q^2/2^n)$
224	$\min\{2^{112}, 2^{128}/\ell^{1/2}\}$	2^{112}
256	$2^{128}/\ell^{1/2}$	2^{128}
384	$\min\{2^{192}, 2^{128}/\ell^{1/2}\}$	2^{192}
512	$2^{256}/\ell^{1/2}$	2^{256}

MAC-Security (Unforgeability)

Size n	HMAC-SHA-2 $O(\ell q^2/2^n)$	HMAC-SHA-3 $O(nq/2^n)$
224	$\min\{2^{112}, 2^{128}/\ell^{1/2}\}$	$2^{216.192...}$
256	$2^{128}/\ell^{1/2}$	2^{248}
384	$\min\{2^{192}, 2^{128}/\ell^{1/2}\}$	$2^{375.415...}$
512	$2^{256}/\ell^{1/2}$	2^{503}

Replacing SHA-2 with SHA-3 enhances generic security of HMAC!

Thank You!

Constrained PRFs for Unbounded Inputs

Hamza Abusalah, Georg Fuchsbauer, and Krzysztof Pietrzak

IST AUSTRIA
Institute of Science and Technology

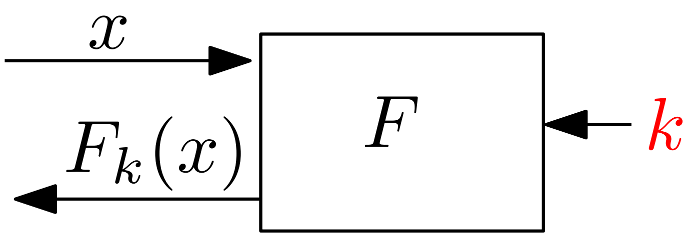
RSA Conference, 2016

Outline

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

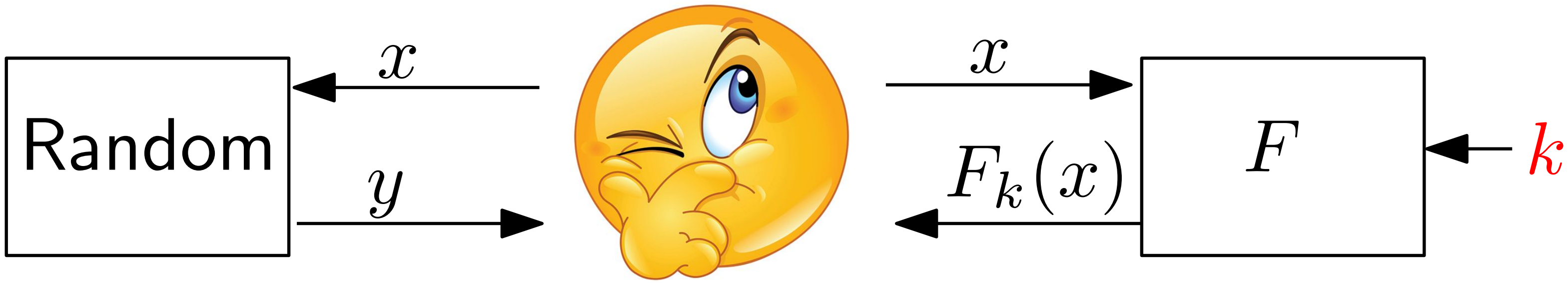
Pseudorandom Functions (PRFs)

[GGM86]



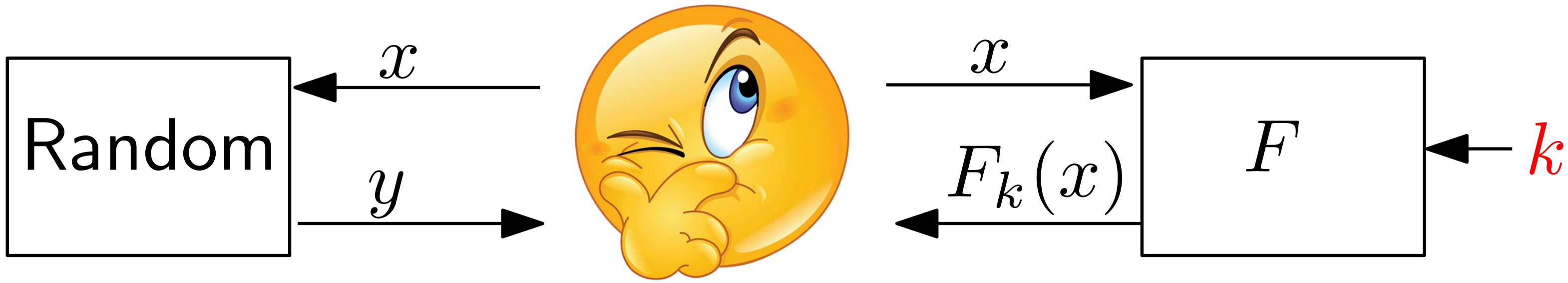
Pseudorandom Functions (PRFs)

[GGM86]



Pseudorandom Functions (PRFs)

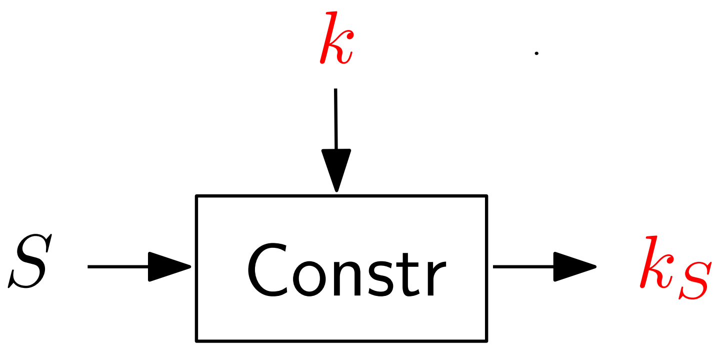
[GGM86]



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

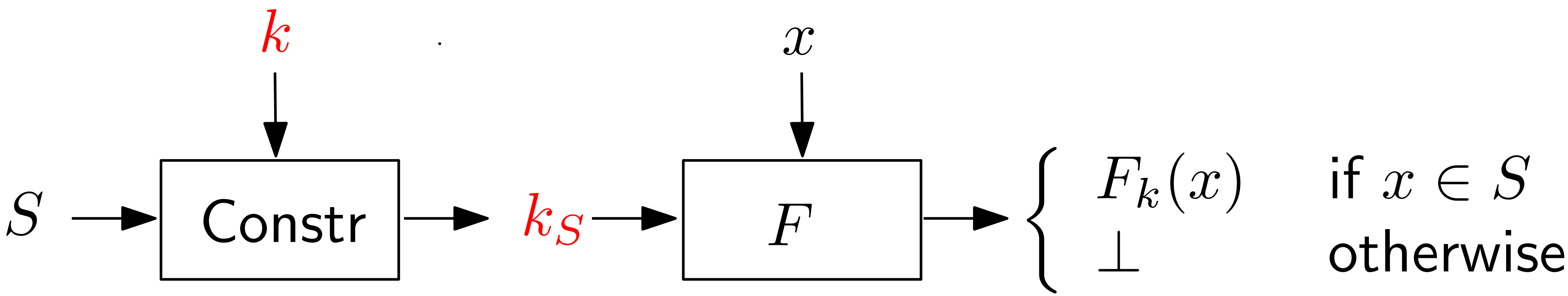
Constrained Pseudorandom Functions (CPRFs)

[BW13],[KPTZ13],[BGI14]



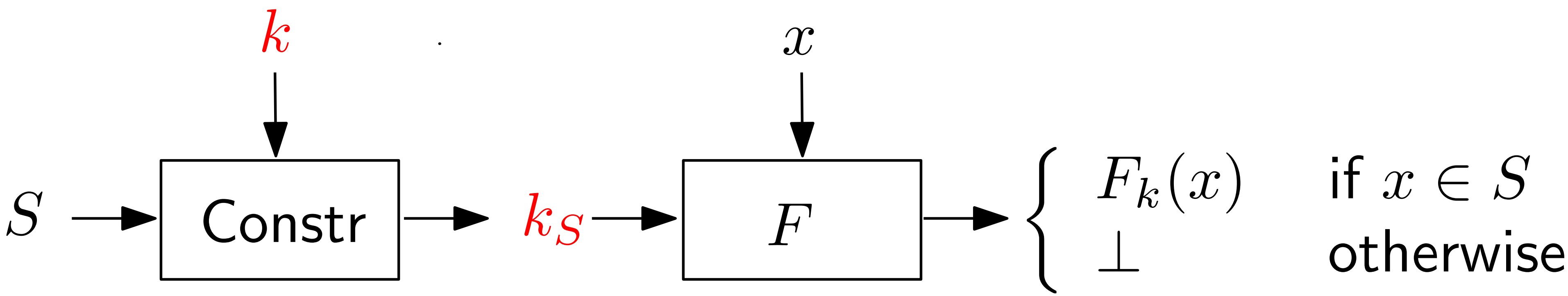
Constrained Pseudorandom Functions (CPRFs)

[BW13],[KPTZ13],[BGI14]



Constrained Pseudorandom Functions (CPRFs)

[BW13],[KPTZ13],[BGI14]

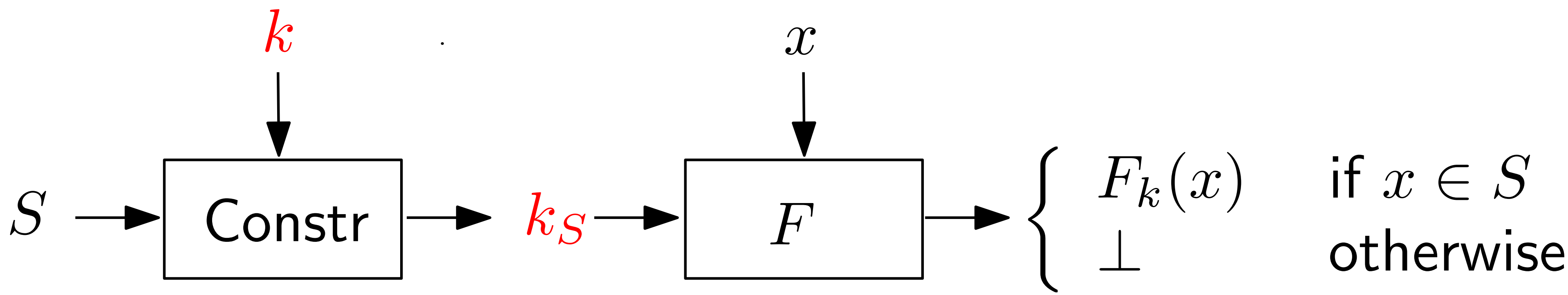


- Polynomial S: Any PRF F is a CPRF

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

Constrained Pseudorandom Functions (CPRFs)

[BW13],[KPTZ13],[BGI14]



- Polynomial S: Any PRF F is a CPRF

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

- Superpolynomial S with short description?

Different Flavors of CPRFs

1. Puncturable [SW14]. $S : x'$

$$k_S \Rightarrow F_k(x) \text{ if } x \neq x'$$

Different Flavors of CPRFs

1. Puncturable [SW14]. $S : x'$

$$k_S \Rightarrow F_k(x) \text{ if } x \neq x'$$

2. Prefix-fixing [BW13]. $S : v \in \{0,1\}^m \| ?^*$, e,g., $v = 101???$

$$k_S \Rightarrow F_k(x) \text{ if } x = 101 \| x'$$

Different Flavors of CPRFs

1. Puncturable [SW14]. $S : x'$

$$k_S \Rightarrow F_k(x) \text{ if } x \neq x'$$

2. Prefix-fixing [BW13]. $S : v \in \{0,1\}^m \| ?^*$, e,g., $v = 101???$

$$k_S \Rightarrow F_k(x) \text{ if } x = 101\|x'$$

3. Bit-fixing [BW13]. $S : v \in \{0,1,?\}^n$, e.g., $v = 1?010?$

$$k_S \Rightarrow F_k(x) \text{ if } x \text{ agrees with } v \text{ on } 0/1$$

Different Flavors of CPRFs

1. Puncturable [SW14]. $S : x'$

$$k_S \Rightarrow F_k(x) \text{ if } x \neq x'$$

2. Prefix-fixing [BW13]. $S : v \in \{0,1\}^m \| ?^*$, e,g., $v = 101???$

$$k_S \Rightarrow F_k(x) \text{ if } x = 101 \| x'$$

3. Bit-fixing [BW13]. $S : v \in \{0, 1, ?\}^n$, e.g., $v = 1?010?$

$$k_S \Rightarrow F_k(x) \text{ if } x \text{ agrees with } v \text{ on } 0/1$$

4. Circuit [BW13]. $S :$ a circuit C

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

Different Flavors of CPRFs

1. Puncturable [SW14]. $S : x'$

$$k_S \Rightarrow F_k(x) \text{ if } x \neq x'$$

2. Prefix-fixing [BW13]. $S : v \in \{0,1\}^m \| ?^*$, e,g., $v = 101???$

$$k_S \Rightarrow F_k(x) \text{ if } x = 101\|x'$$

3. Bit-fixing [BW13]. $S : v \in \{0,1,?\}^n$, e.g., $v = 1?010?$

$$k_S \Rightarrow F_k(x) \text{ if } x \text{ agrees with } v \text{ on } 0/1$$

4. Circuit [BW13]. $S :$ a circuit C

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

5. This work:Turing Machine (TM). $S :$ a TM M

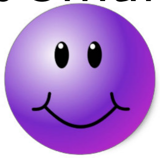
$$k_S \Rightarrow F_k(x) \text{ if } M(x) = 1$$

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

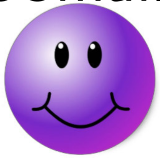
Identity-Based Non-interactive Key Exchange

a@mail

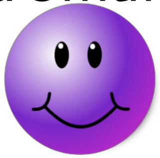
b@mail



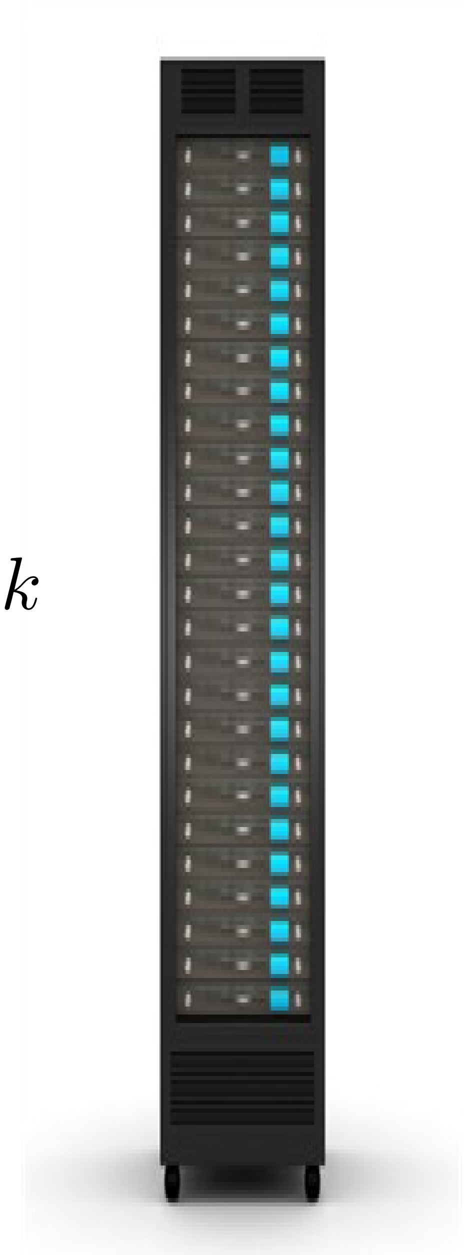
c@mail



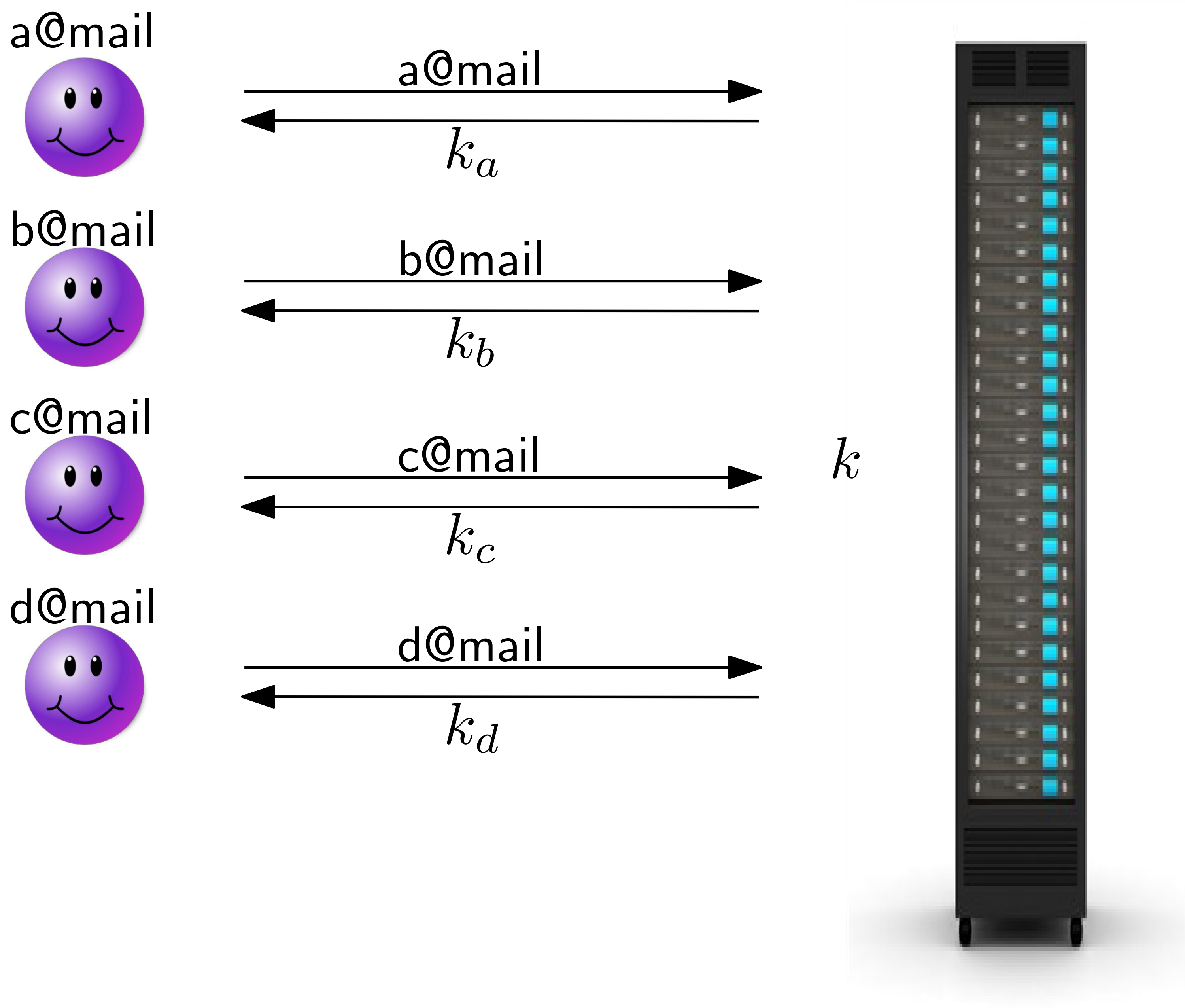
d@mail



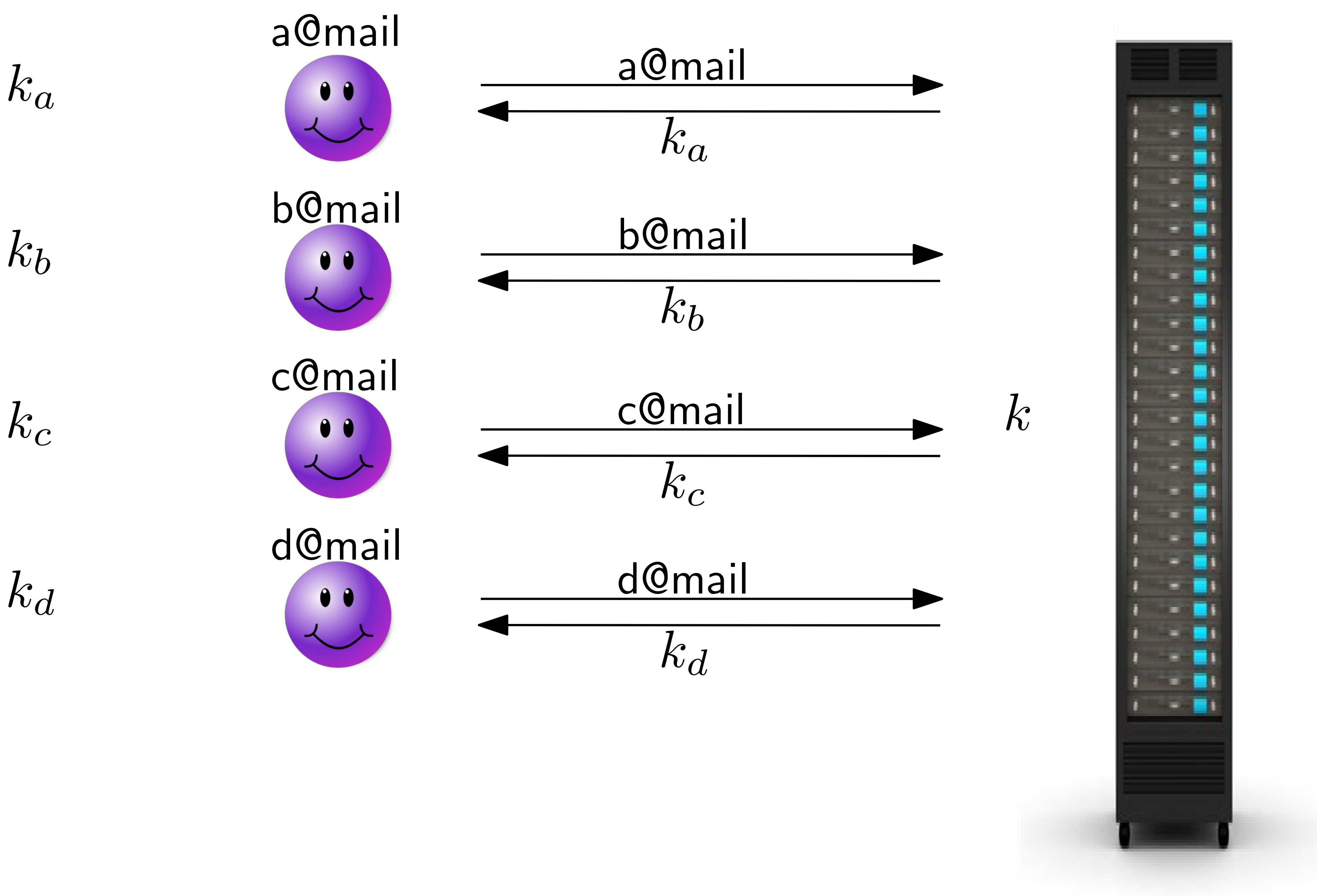
k



Identity-Based Non-interactive Key Exchange



Identity-Based Non-interactive Key Exchange



Identity-Based Non-interactive Key Exchange

k_a

a@mail

k_b

b@mail

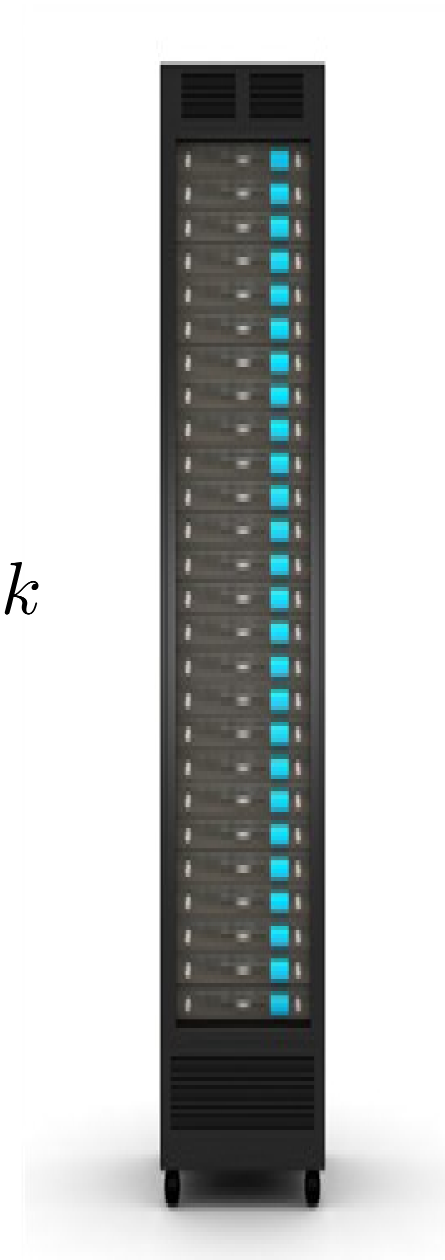
k_c

c@mail

k_d

d@mail

k



Identity-Based Non-interactive Key Exchange

k_a ,k_{ac}

a@mail

k_b

b@mail

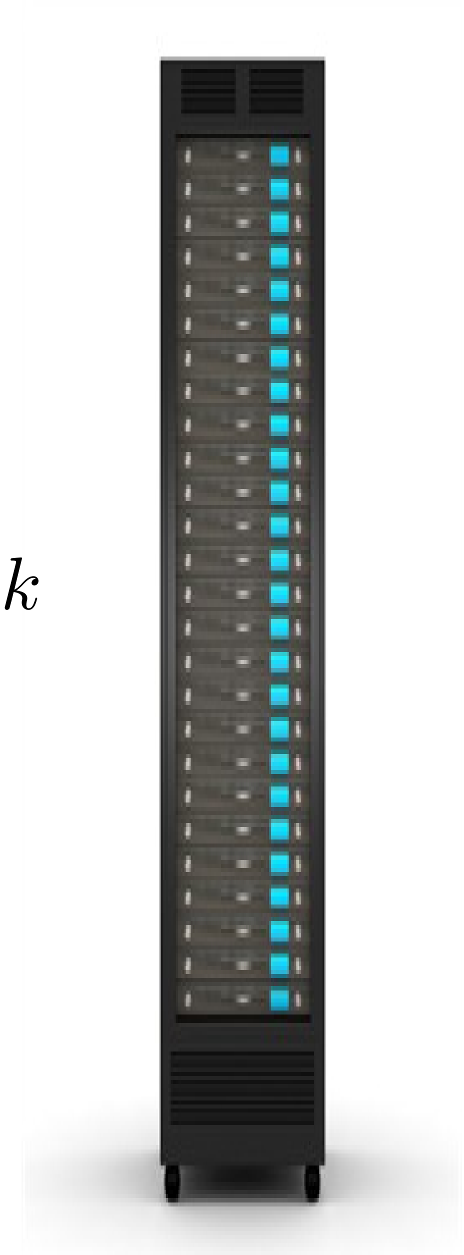
k_c ,k_{ac}

c@mail

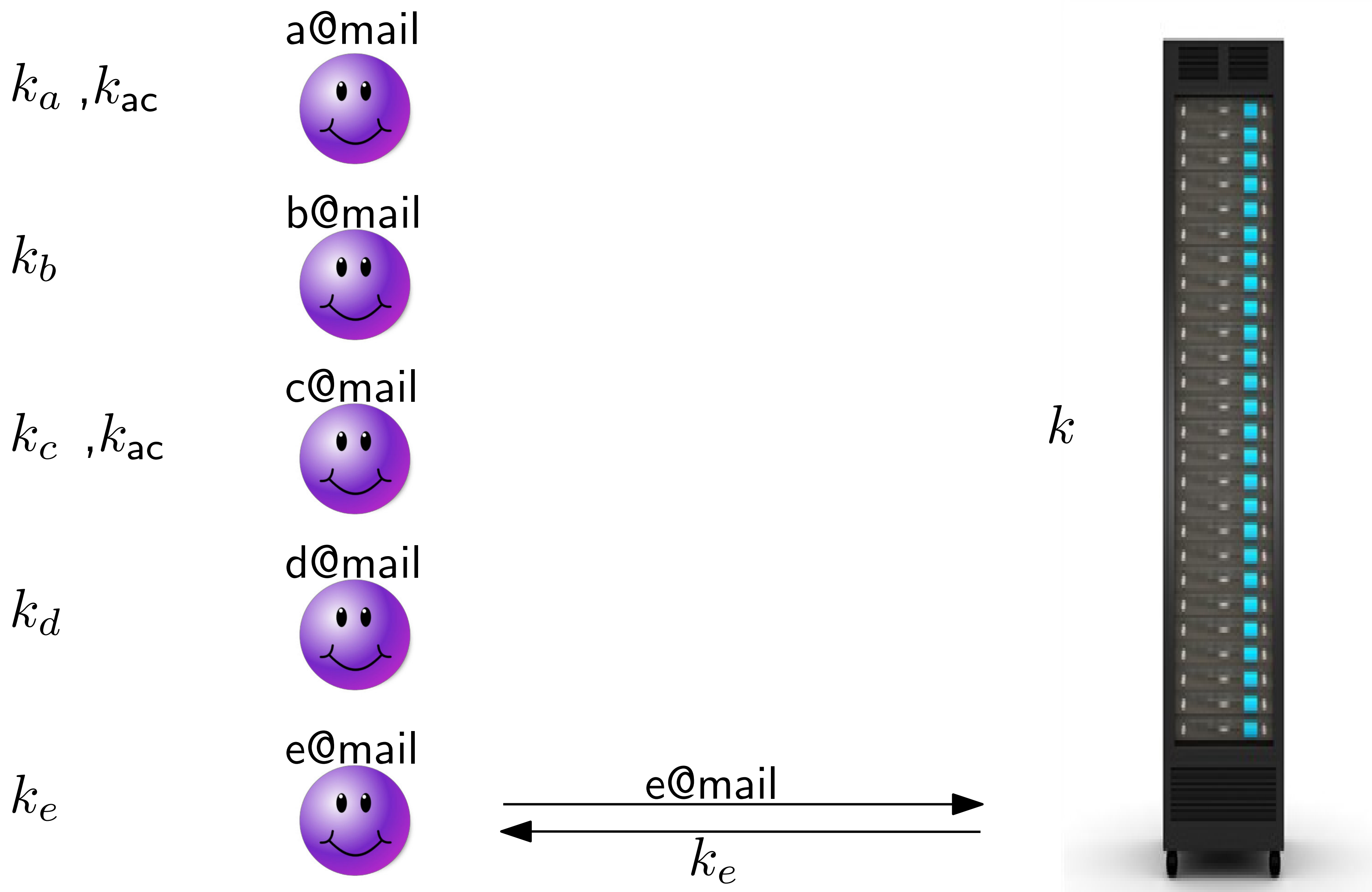
k_d

d@mail

k



Identity-Based Non-interactive Key Exchange

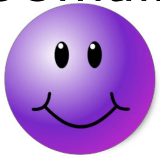


Identity-Based Non-interactive Key Exchange

k_a ,k_{ac} a@mail

k_b b@mail

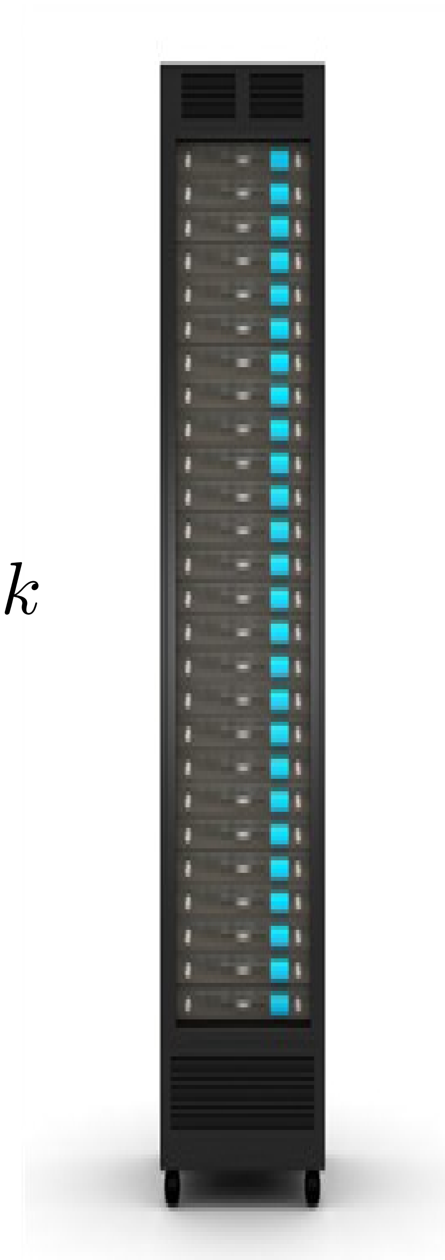
k_c ,k_{ac} c@mail



k_d d@mail

k_e e@mail

k



Identity-Based Non-interactive Key Exchange

k_a ,k_{ac} ,k_{abe}

a@mail

k_b ,k_{abe}

b@mail

k_c ,k_{ac}

c@mail

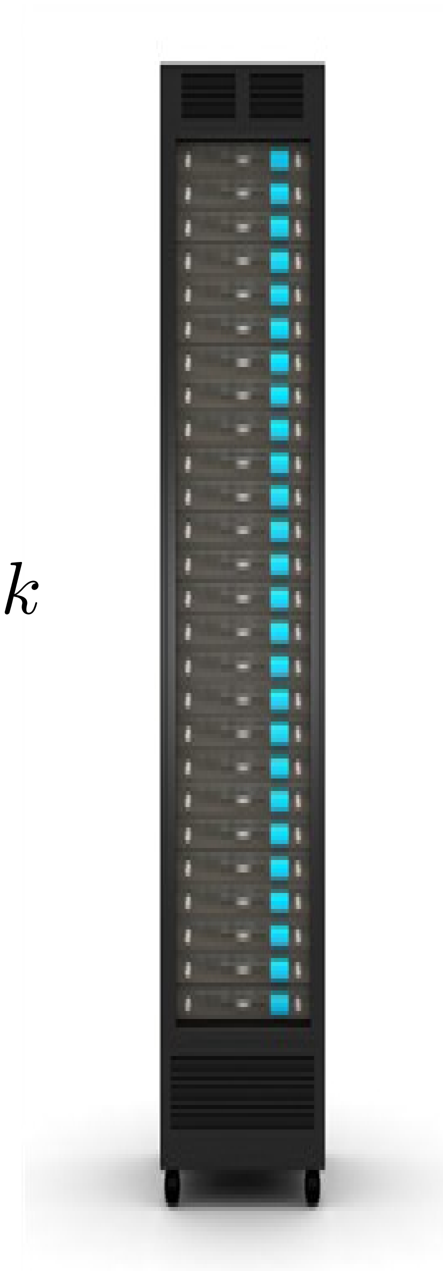
k_d

d@mail

k_e ,k_{abe}

e@mail

k



Identity-Based Non-interactive Key Exchange

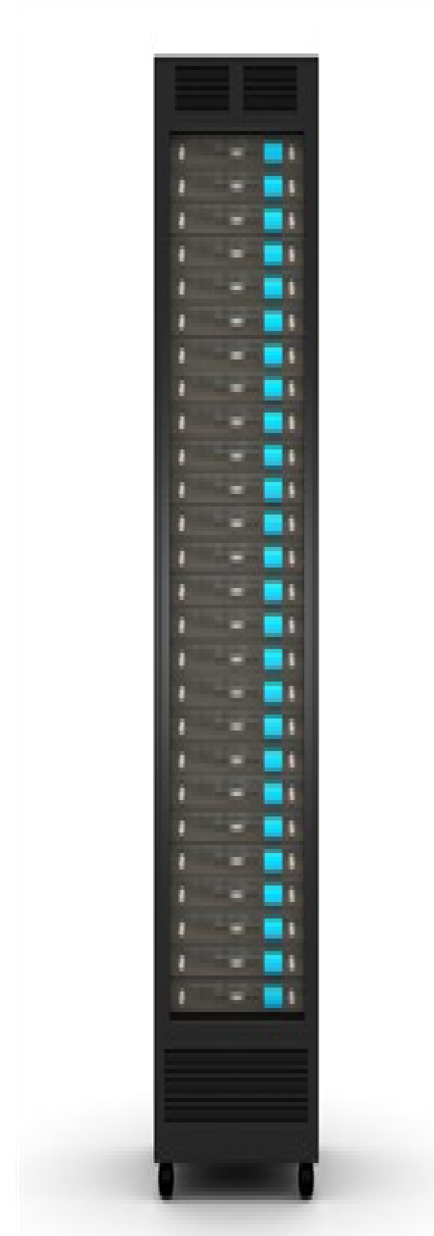
a@mail

b@mail

c@mail

d@mail

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



Identity-Based Non-interactive Key Exchange

a@mail

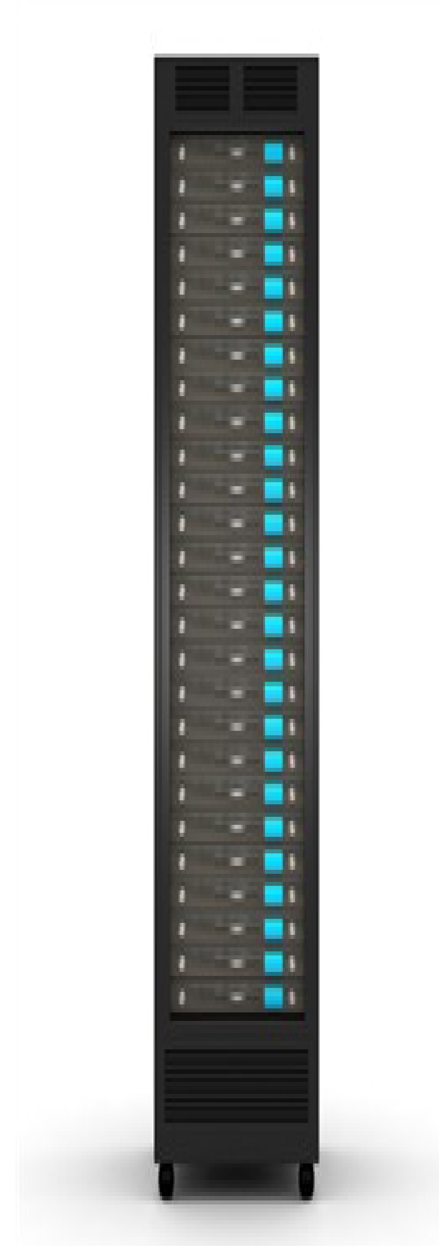
$\mathsf{F}_{k} : \{0,1\}^* \rightarrow \{0,1\}^m$

b@mail

$\xrightarrow{\text{b@mail}}$

c@mail

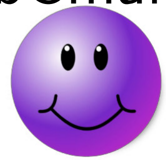
d@mail



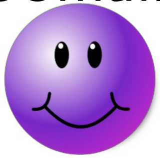
Identity-Based Non-interactive Key Exchange

a@mail

b@mail



c@mail

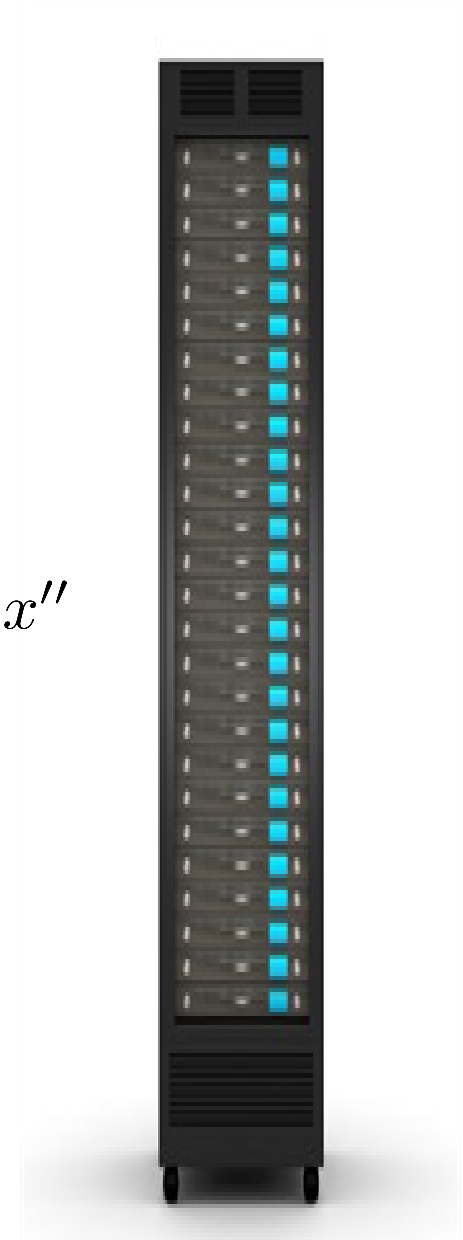


d@mail

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

$$\xrightarrow{\text{b@mail}}$$

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



Identity-Based Non-interactive Key Exchange

a@mail

b@mail

c@mail

d@mail

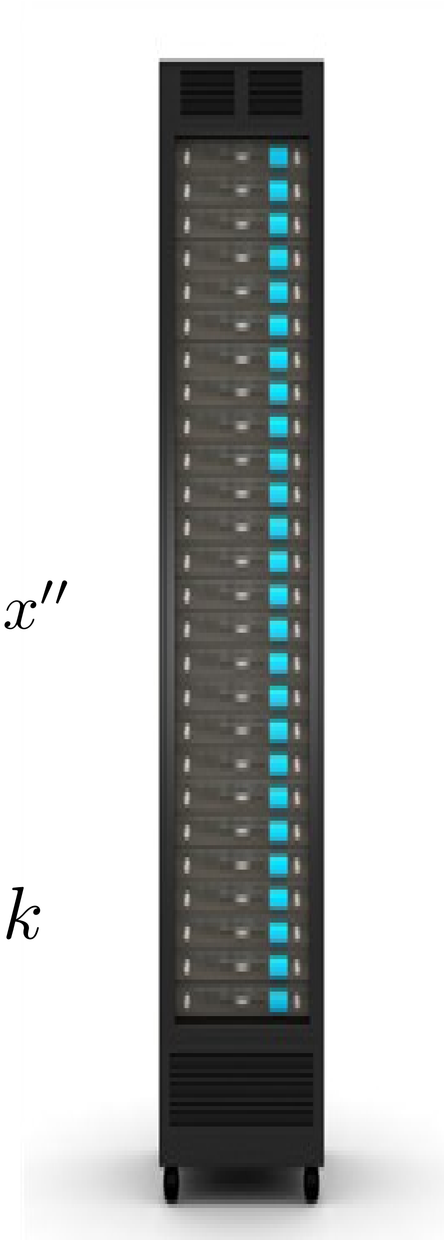
$\mathsf{F}_{k} : \{0,1\}^* \to \{0,1\}^m$

$\xrightarrow{\text{b@mail}}$

$M_{\text{b@mail}}(x) =$

$\begin{cases} 1 & \text{if } x = x' \| \text{ b@mail } \| x'' \\ 0 & \text{otherwise} \end{cases}$

$k_{M_{\text{b@mail}}} \leftarrow \boxed{\text{Constr}} \leftarrow k$



Identity-Based Non-interactive Key Exchange

a@mail

$k_{M_{\text{a@mail}}}$ 

b@mail

$k_{M_{\text{b@mail}}}$ 

c@mail

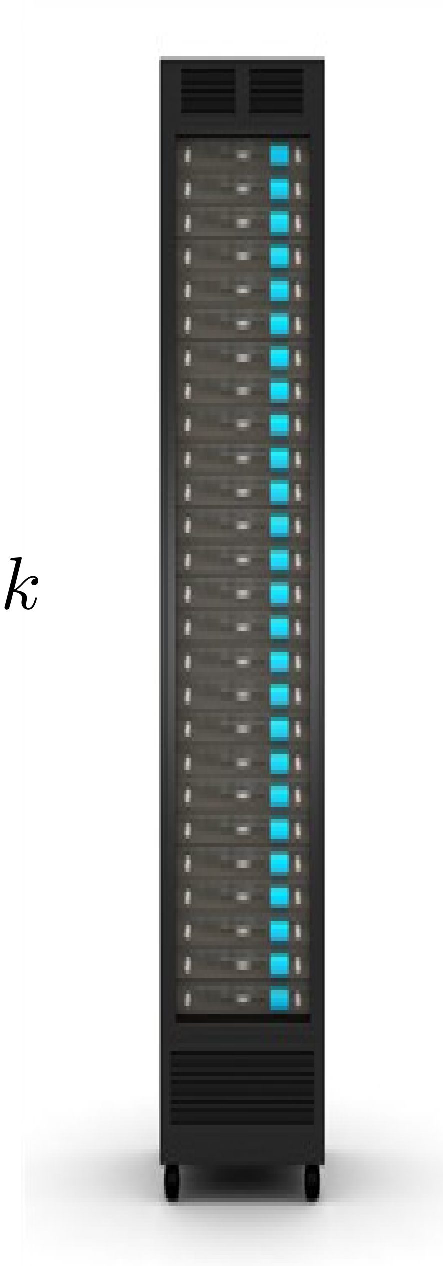
$k_{M_{\text{c@mail}}}$ 

d@mail

$k_{M_{\text{d@mail}}}$ 

e@mail

$k_{M_{\text{e@mail}}}$ 

k 

Identity-Based Non-interactive Key Exchange

$k_{M_{\text{a@mail}}}$ a@mail

$k_{M_{\text{b@mail}}}$ b@mail

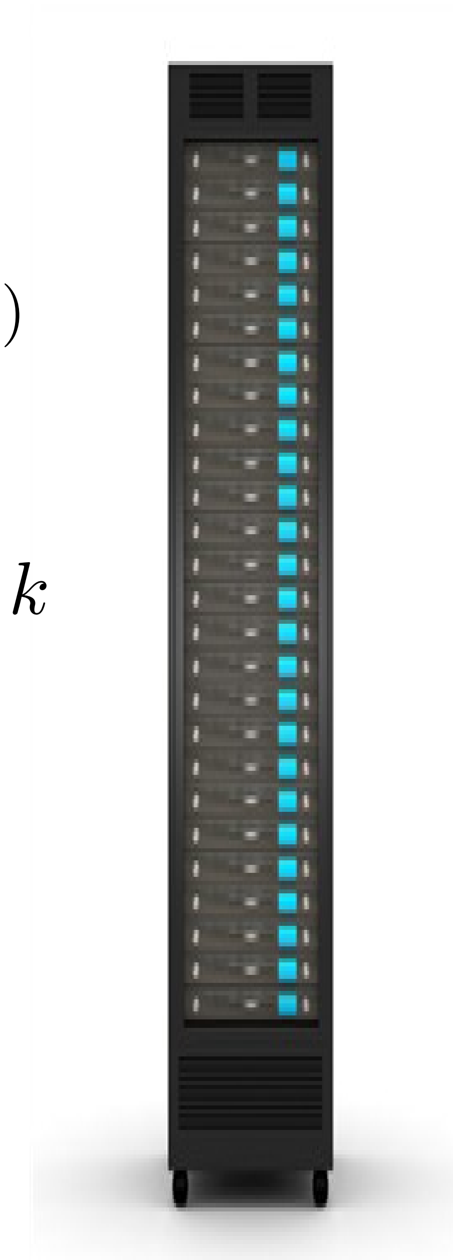
$k_{M_{\text{c@mail}}}$ c@mail

$k_{M_{\text{d@mail}}}$ d@mail

$k_{M_{\text{e@mail}}}$ e@mail

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

k



Identity-Based Non-interactive Key Exchange

a@mail

$k_{M_{\mathsf{a@mail}}}$ 

b@mail

$k_{M_{\mathsf{b@mail}}}$ 

c@mail

$k_{M_{\mathsf{c@mail}}}$ 

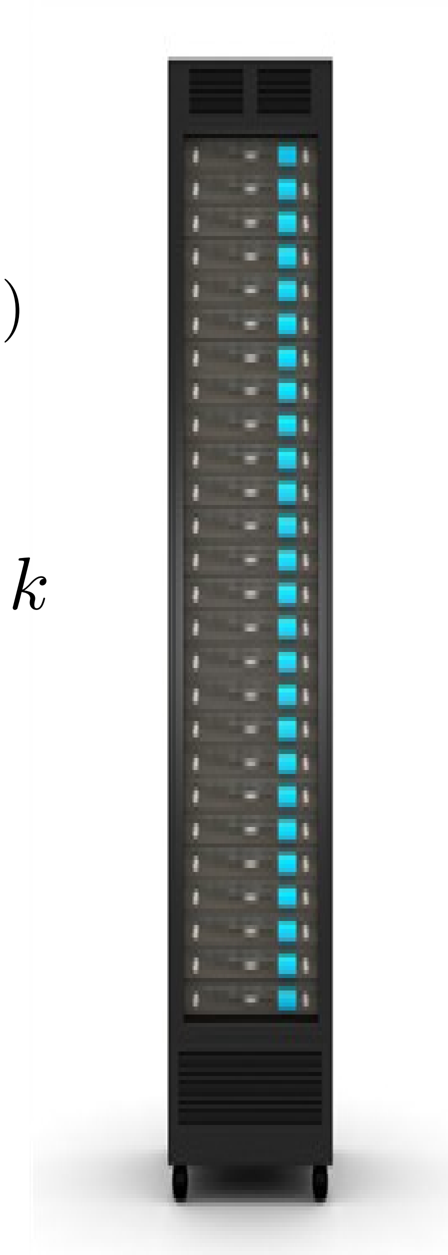
d@mail

$k_{M_{\mathsf{d@mail}}}$ 

e@mail

$k_{M_{\mathsf{e@mail}}}$ 

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

k 

Identity-Based Non-interactive Key Exchange

a@mail

$k_{M_{\text{a@mail}}}$ 

b@mail

$k_{M_{\text{b@mail}}}$ 

c@mail

$k_{M_{\text{c@mail}}}$ 

d@mail

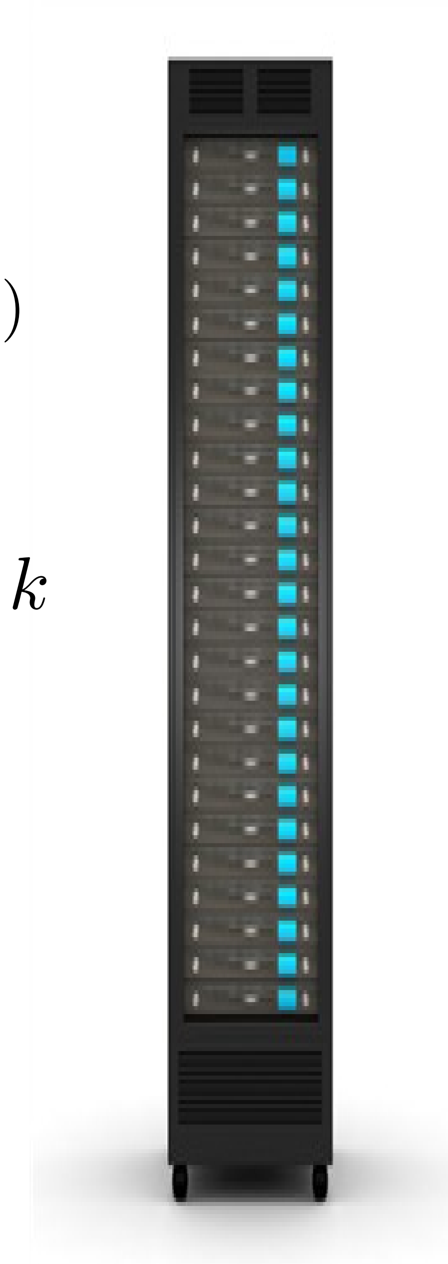
$k_{M_{\text{d@mail}}}$ 

e@mail

$k_{M_{\text{e@mail}}}$ 

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

k



Identity-Based Non-interactive Key Exchange

a@mail

$k_{M_{\mathsf{a@mail}}}$

b@mail

$k_{M_{\mathsf{b@mail}}}$

c@mail

$k_{M_{\mathsf{c@mail}}}$

d@mail

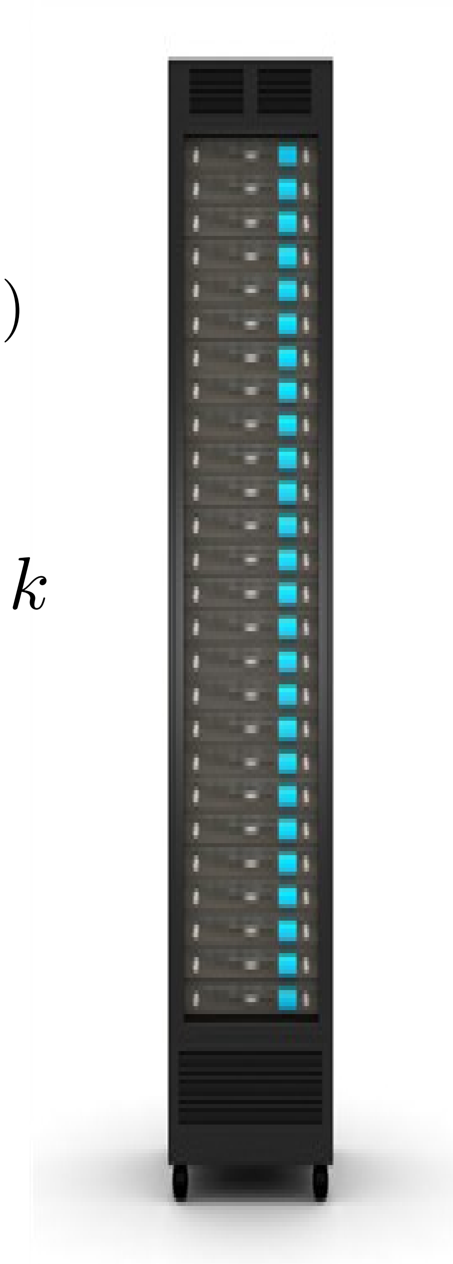
$k_{M_{\mathsf{d@mail}}}$

e@mail

$k_{M_{\mathsf{e@mail}}}$

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

k



TM CPRFs

1) A warm-up: a simple circuit CPRF assuming

- Puncturable PRFs
- Indistinguishability obfuscation

TM CPRFs

1) A warm-up: a simple circuit CPRF assuming

- Puncturable PRFs
- Indistinguishability obfuscation

2) A TM CPRF assuming

- Punctured PRFs
- Public coin differing input obfuscation
- Succinct non-interactive arguments of knowledge (SNARKs)
- Collision resistant hashing

Program Obfuscation

[BGI+01]

Virtual Black Box
[BGI+01]

Differing Input
[BGI+01],[BCP14]

Public Coin Differing Input
[ISP15]

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

Program Obfuscation

[BGI$^+$01]

Virtual Black Box
[BGI$^+$01]

Impossible
[BGI$^+$01]

Differing Input
[BGI$^+$01],[BCP14]

Implausible
[GGH$^+$14]

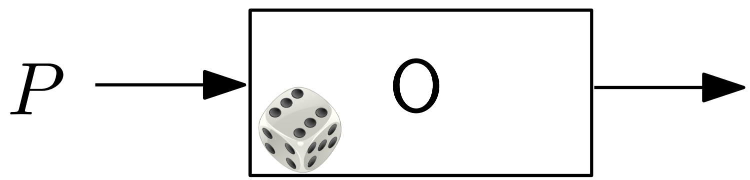
TM-impossible
[BSW16]

Public Coin Differing Input
[ISP15]

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

Program Obfuscation (1)

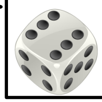
1) Functionality:



$$\tilde{P} \equiv P$$
$$\forall x, \tilde{P}(x) = P(x)$$

Program Obfuscation (1)

1) Functionality:



$$P \longrightarrow \boxed{\;\mathsf{O}\;} \longrightarrow \tilde{P} \equiv P$$

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

2) Indistinguishability obfuscation:

$b?$

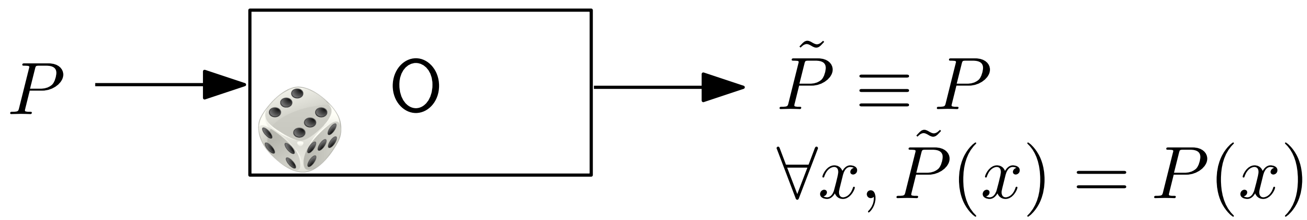
$$\xrightarrow{P_0, P_1}$$

$$P_0 \equiv P_1$$

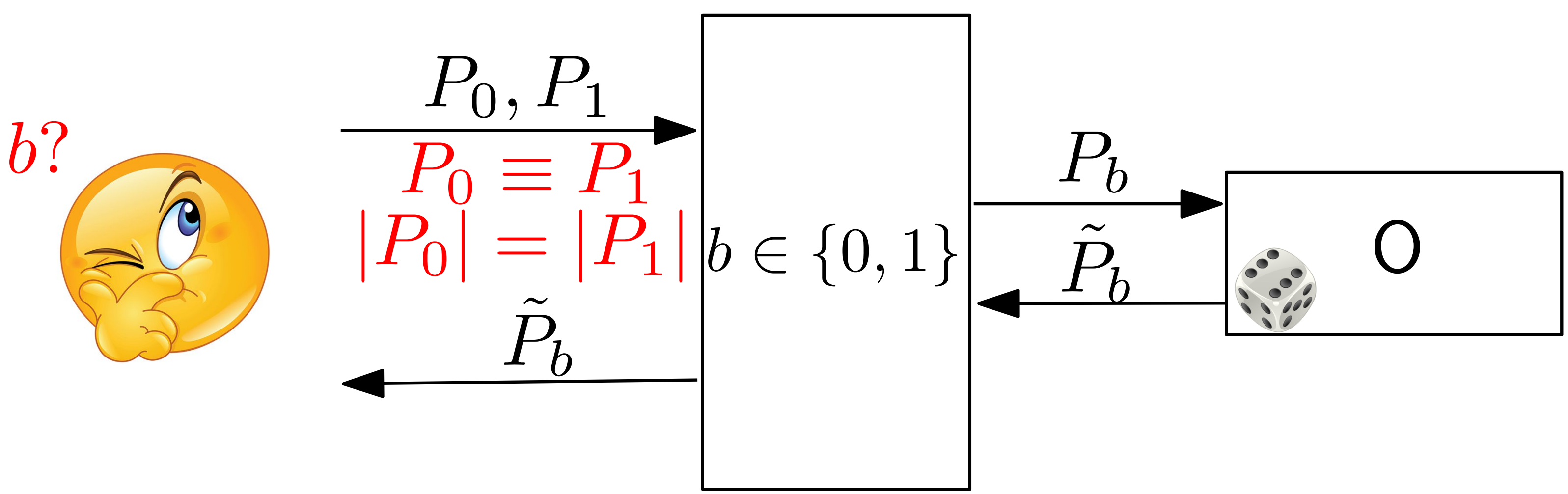
$$|P_0| = |P_1|$$

Program Obfuscation (1)

1) Functionality:

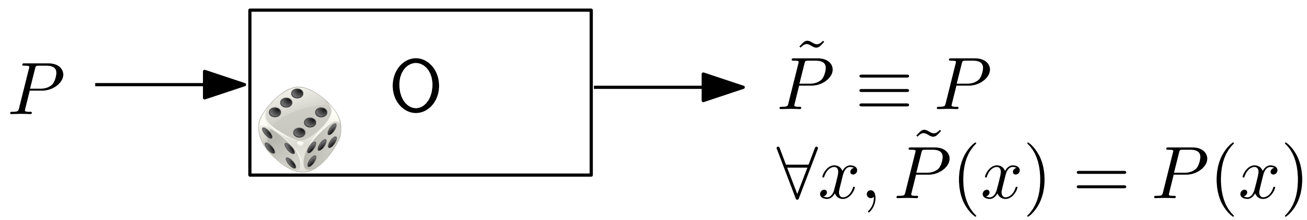


2) Indistinguishability obfuscation: hard to guess b

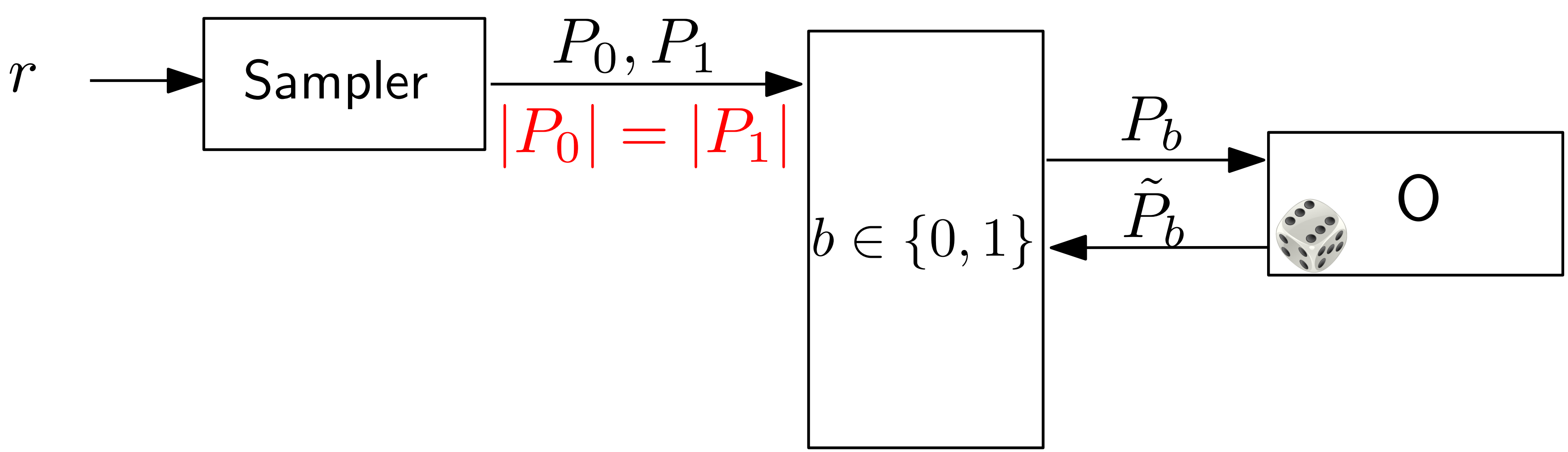


Program Obfuscation (2)

1) Functionality:

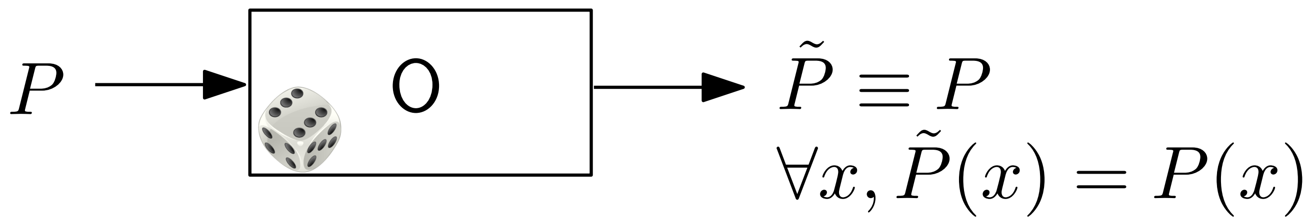


2) Differing input obfuscation:

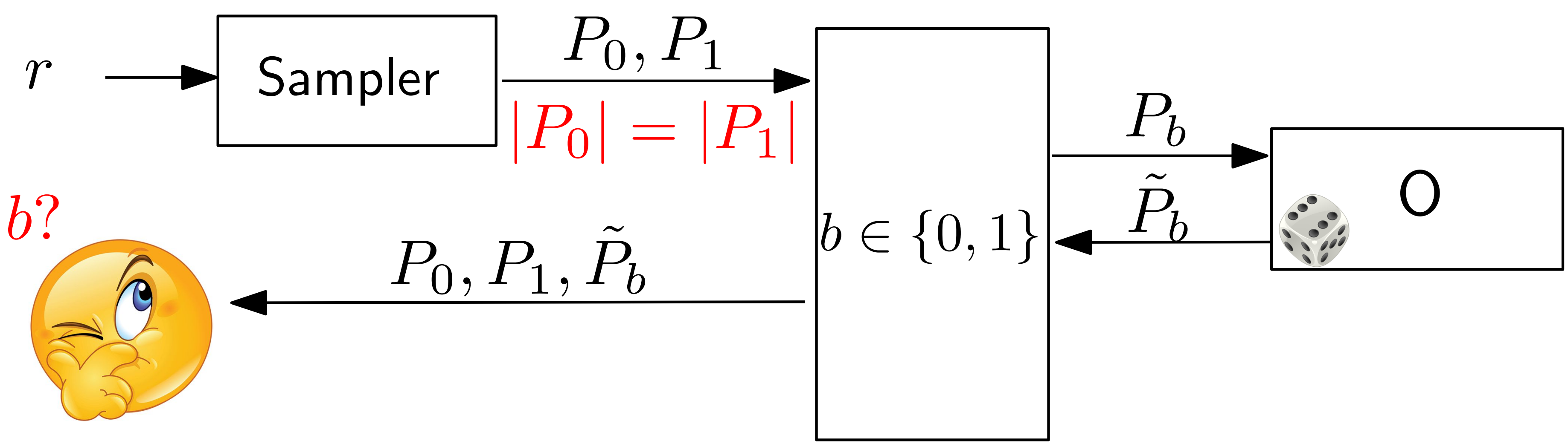


Program Obfuscation (2)

1) Functionality:

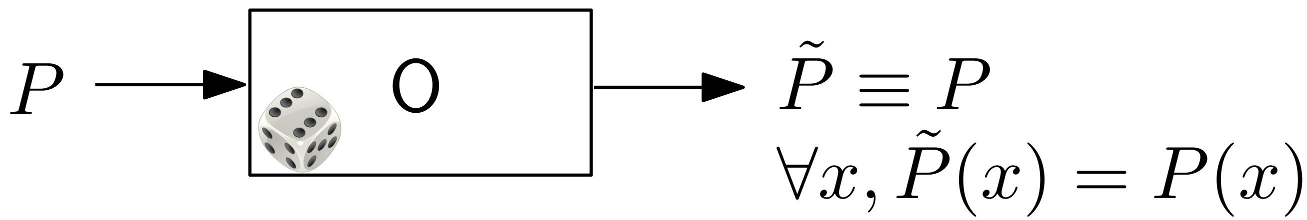


2) Differing input obfuscation: hard to guess b if it's hard to find x, s.t. $P_0(x) \neq P_1(x)$

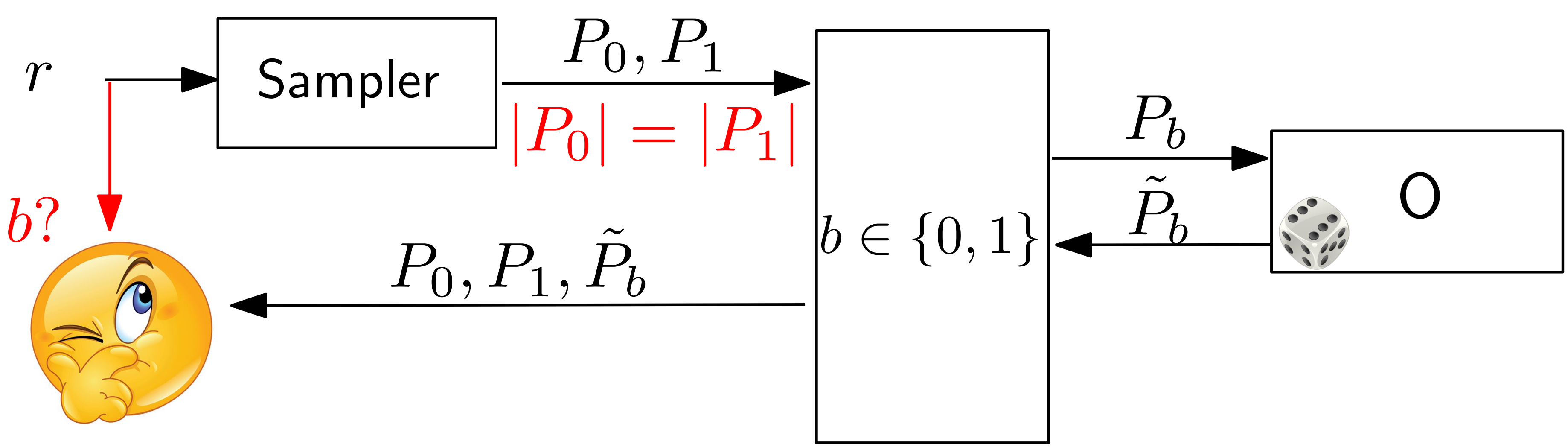


Program Obfuscation (3)

1) Functionality:



2) Public coin differing input obfuscation: hard to guess b if it's hard to find x, s.t. $P_0(x) \neq P_1(x)$



A Circuit CPRF

- $\mathsf{PF}_k : \{0,1\}^n \to \{0,1\}^m$ a puncturable PRF
- iO an indistinguishability obfuscator

A Circuit CPRF

- $\mathsf{PF}_k : \{0,1\}^n \to \{0,1\}^m$ a puncturable PRF
- iO an indistinguishibility obfuscator

Define a circuit CPRF F as:

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

A Circuit CPRF

- $\mathsf{PF}_k : \{0,1\}^n \to \{0,1\}^m$ a puncturable PRF
- iO an indistinguishibility obfuscator

Define a circuit CPRF F as:

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

$\mathsf{Constr}(k, C) \to k_C$:

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

A Circuit CPRF

- $\mathsf{PF}_k : \{0,1\}^n \to \{0,1\}^m$ a puncturable PRF
- iO an indistinguishability obfuscator

Define a circuit CPRF F as:

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

$\mathsf{Constr}(k, C) \to k_C$:

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

A Circuit CPRF

- $\mathsf{PF}_k : \{0,1\}^n \to \{0,1\}^m$ a puncturable PRF
- iO an indistinguishability obfuscator

Define a circuit CPRF F as:

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

$\mathsf{Constr}(k, C) \to k_C$:

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

Thm 1. F *is a secure circuit CPRF.*

Towards a TM CPRF (1)

Constrained keys $k_{\color{red}C}$:

$$\mathsf{iO}\left(P_{k,{\color{red}C}}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if} \quad \underbrace{{\color{red}C(x) = 1}}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases}\right)$$

Towards a TM CPRF (1)

Constrained keys k_{M}:

$$\mathrm{iO}\left(P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases} \right)$$

Towards a TM CPRF (1)

Constrained keys k_{M}:

$$\mathsf{iO}\left(P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases}\right)$$

iO for Turing machines [KLW15]

Towards a TM CPRF (1)

Constrained keys k_{M}:

$$\mathsf{iO}\left(P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases}\right)$$

iO for Turing machines [KLW15]

Security Proof: $\mathsf{iO}(P_{k,M}) \approx_c \mathsf{iO}(P_{k_{x'},M})$

where $k_{x'}$ evalaues PF on all $x \neq x'$

Towards a TM CPRF (1)

Constrained keys k_M:

$$\mathsf{iO}\left(P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases} \right)$$

iO for Turing machines [KLW15]

Security Proof: $\mathsf{iO}(P_{k,M}) \approx_c \mathsf{iO}(P_{k_{x'},M})$

where $k_{x'}$ evalaues PF on all $x \neq x'$

1) $P_{k,M} \equiv P_{k_{x'},M}$

Towards a TM CPRF (1)

Constrained keys k_M:

$$\mathsf{iO}\left(P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \bot & \text{otherwise} \end{cases} \right)$$

iO for Turing machines [KLW15]

Security Proof: $\mathsf{iO}(P_{k,M}) \approx_c \mathsf{iO}(P_{k_{x'},M})$

where $k_{x'}$ evalaues PF on all $x \neq x'$

1) $P_{k,M} \equiv P_{k_{x'},M}$

2) $|P_{k,M}| \stackrel{?}{=} |P_{k_{x'},M}|$: For $x' \in \{0,1\}^*$, $k_{x'}$ unbounded

Towards a TM CPRF (2)

Let H: $\{0,1\}^* \to \{0,1\}^n$ be a hash function.

$$P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(x) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases}$$

Towards a TM CPRF (2)

Let H: $\{0,1\}^* \to \{0,1\}^n$ be a hash function.

$$P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(H(x)) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases}$$

Towards a TM CPRF (2)

Let H: $\{0,1\}^* \to \{0,1\}^n$ be a hash function.

$$P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(H(x)) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases}$$

1)$|P_{k_{H(x')}}|$ bounded: $k_{H(x')}$ is bounded even for $x' \in \{0,1\}^*$

Towards a TM CPRF (2)

Let H: $\{0,1\}^* \to \{0,1\}^n$ be a hash function.

$$P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(H(x)) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases}$$

1) $|P_{k_{H(x')}}|$ bounded: $k_{H(x')}$ is bounded even for $x' \in \{0,1\}^*$

2) $P_{k,M} \not\equiv P_{k_{H(x')},M}$

Towards a TM CPRF (2)

Let H: $\{0,1\}^* \to \{0,1\}^n$ be a hash function.

$$P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(H(x)) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases}$$

1) $|P_{k_{H(x')}}|$ bounded: $k_{H(x')}$ is bounded even for $x' \in \{0,1\}^*$

2) $P_{k,M} \not\equiv P_{k_{H(x')},M}$

Differing inputs: $x \neq x'$ s.t. $H(x) = H(x') := h'$

$$P_{k,M}(x,\pi) = \mathsf{PF}_k(h') \quad \neq \quad P_{k_{h'},M}(x) = \perp$$

Towards a TM CPRF (2)

Let $H\colon \{0,1\}^* \to \{0,1\}^n$ be a hash function.

$$P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(H(x)) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases}$$

1) $|P_{k_{H(x')}}|$ bounded: $k_{H(x')}$ is bounded even for $x' \in \{0,1\}^*$

2) $P_{k,M} \not\equiv P_{k_{H(x')},M}$

Differing inputs: $x \neq x'$ s.t. $H(x) = H(x') := h'$

$$P_{k,M}(x,\pi) = \mathsf{PF}_k(h') \quad \neq \quad P_{k_{h'},M}(x) = \perp$$

Security proof: Public-coin diO for Turing machines [ISP15]

Towards a TM CPRF (3)

Instead of:

$$k_M \leftarrow \mathsf{diO}\left(P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(H(x)) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases} \right)$$

Towards a TM CPRF (3)

Instead of:

$$k_M \leftarrow \mathsf{diO}\left(P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(H(x)) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases}\right)$$

$$P_k(h,\pi) := \begin{cases} \mathsf{PF}_k(h) & \text{if } \pi \text{ proves } h = H(x) \wedge M(x) = 1 \\ \perp & \text{otherwise} \end{cases}$$

Towards a TM CPRF (3)

Instead of:

$$k_M \leftarrow \mathsf{diO}\left(P_{k,M}(x) := \begin{cases} \mathsf{PF}_k(H(x)) & \text{if} \quad \underbrace{M(x) = 1}_{\text{Input Consistency}} \\ \perp & \text{otherwise} \end{cases}\right)$$

$$P_k(h, \pi) := \begin{cases} \mathsf{PF}_k(h) & \text{if } \pi \text{ proves } h = H(x) \wedge M(x) = 1 \\ \perp & \text{otherwise} \end{cases}$$

If π is a Succinct Non-interactive Argument of Knowledge (SNARK):

Security proof: Public-coin diO for circuits [ISP15]

A TM CPRF

- $\mathsf{PF}_k : \{0,1\}^n \to \{0,1\}^m$ a puncturable PRF
- $H : \{0,1\}^* \to \{0,1\}^n$ a collision resistant hash
- SNARKs
- A public-coin diO for circuits

A TM CPRF

- $\mathsf{PF}_k : \{0,1\}^n \to \{0,1\}^m$ a puncturable PRF
- $H : \{0,1\}^* \to \{0,1\}^n$ a collision resistant hash
- SNARKs
- A public-coin diO for circuits

Define a TM CPRF F as:

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

A TM CPRF

- $\mathsf{PF}_k : \{0,1\}^n \to \{0,1\}^m$ a puncturable PRF
- $H : \{0,1\}^* \to \{0,1\}^n$ a collision resistant hash
- SNARKs
- A public-coin diO for circuits

Define a TM CPRF F as:

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

$\mathsf{Constr}(k, M) \to k_M$:

$$P(h,\pi) := \begin{cases} \mathsf{PF}_k(h) & \text{if } \pi \text{ SNARK: } H(x) = h \wedge M(x) = 1 \\ \perp & \text{otherwise} \end{cases}$$

A TM CPRF

- $\mathsf{PF}_k : \{0,1\}^n \to \{0,1\}^m$ a puncturable PRF
- $H : \{0,1\}^* \to \{0,1\}^n$ a collision resistant hash
- SNARKs
- A public-coin diO for circuits

Define a TM CPRF F as:

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

$\mathsf{Constr}(k, M) \to k_M$:

$$P(h,\pi) := \begin{cases} \mathsf{PF}_k(h) & \text{if } \pi \text{ SNARK: } H(x) = h \wedge M(x) = 1 \\ \perp & \text{otherwise} \end{cases}$$

$$k_M = \tilde{P} \leftarrow \mathsf{diO}(P)$$