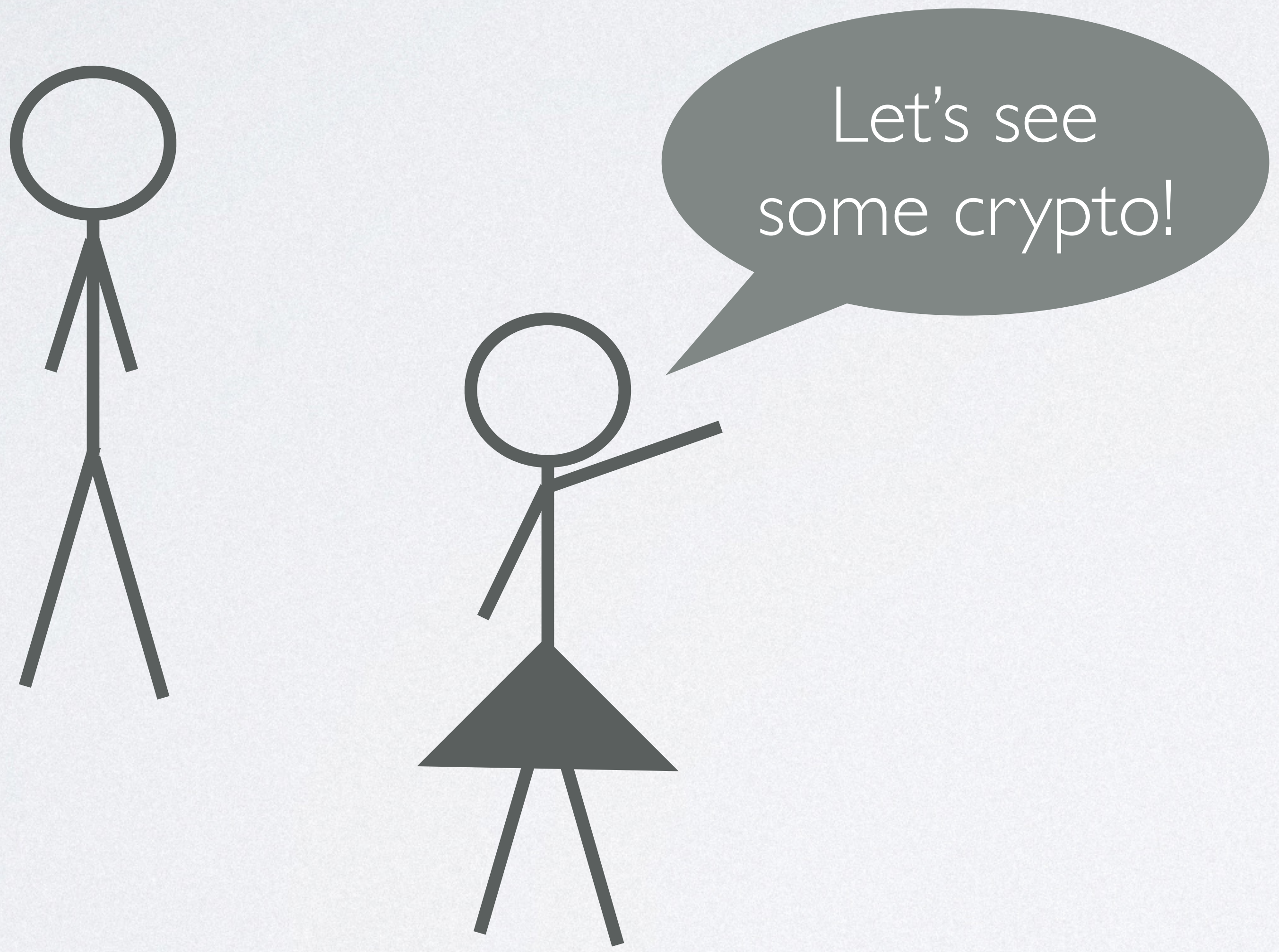
Non-Uniform Bounds in the Random-Permutation, Ideal-Cipher, and Generic-Group Models

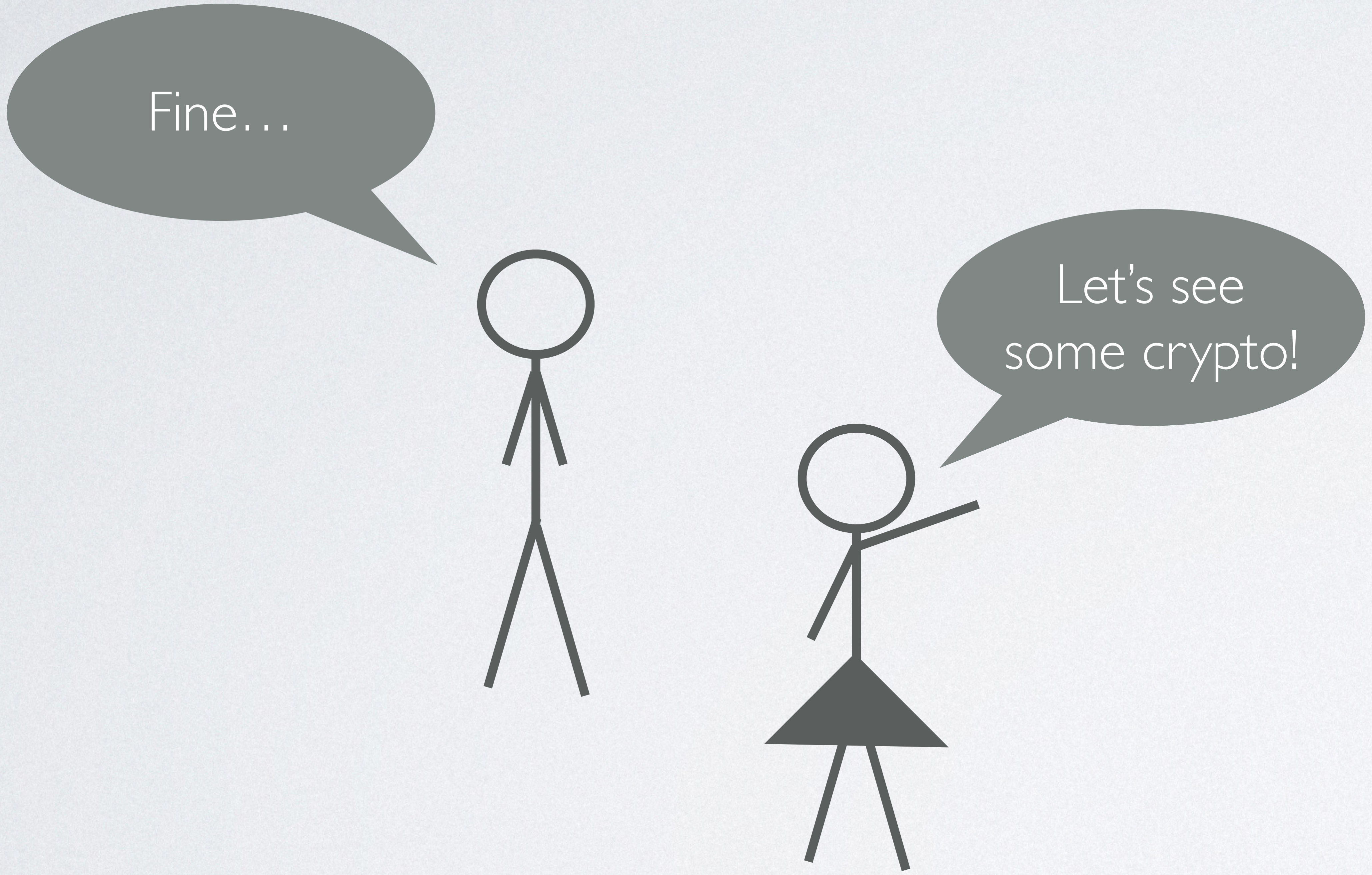
Sandro Coretti
New York University

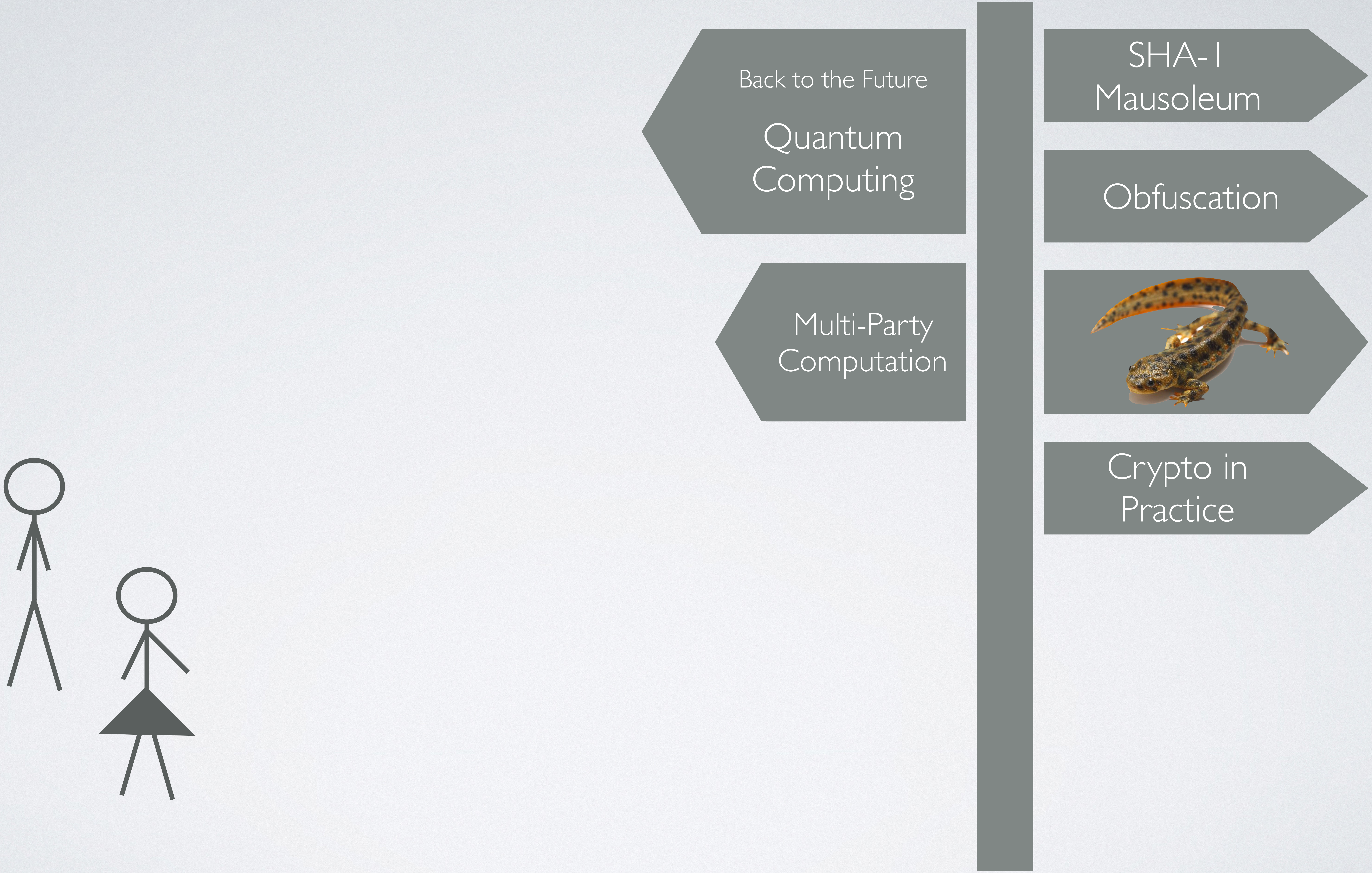
Joint work with:

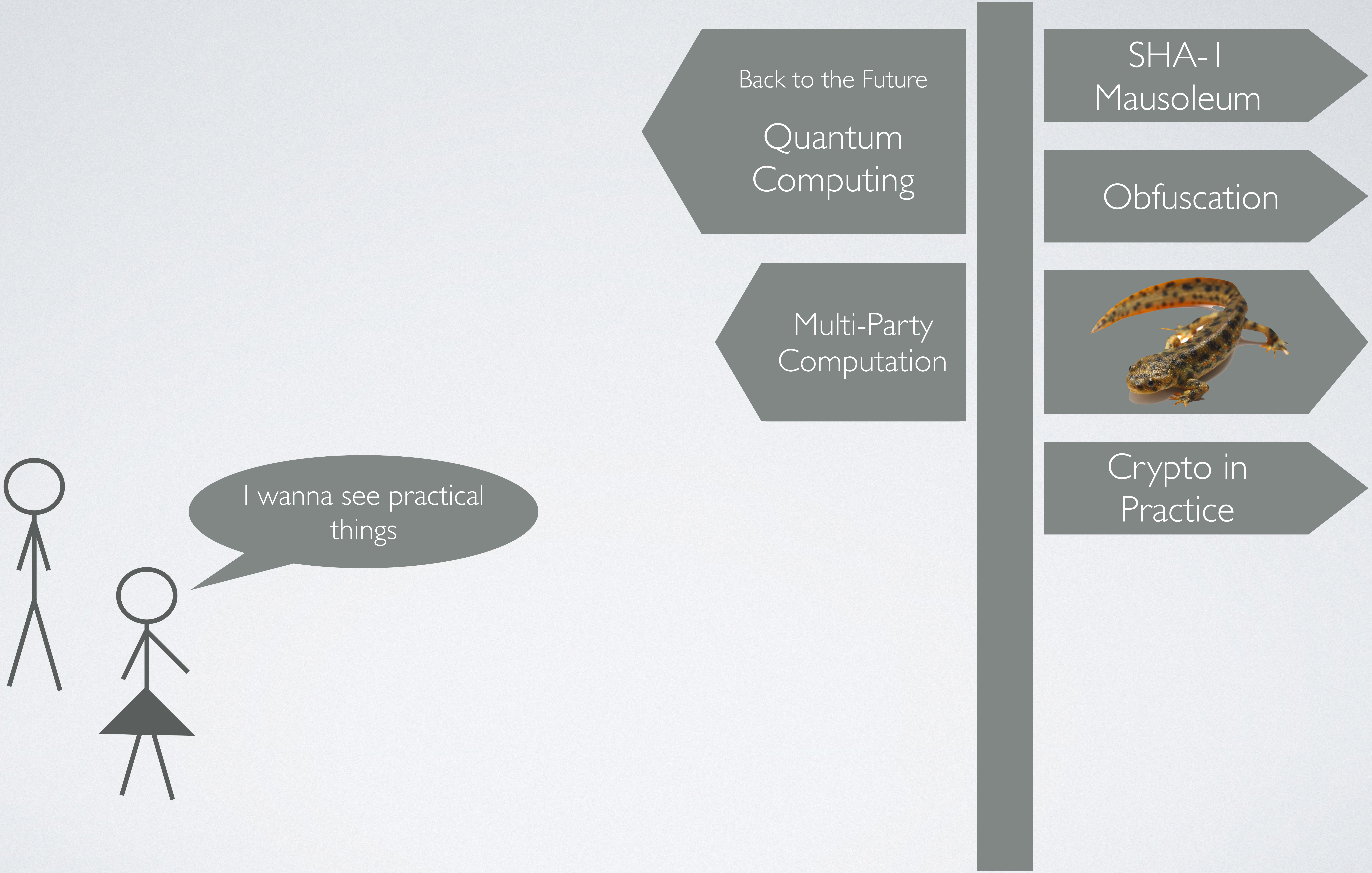
Yevgeniy Dodis
New York University

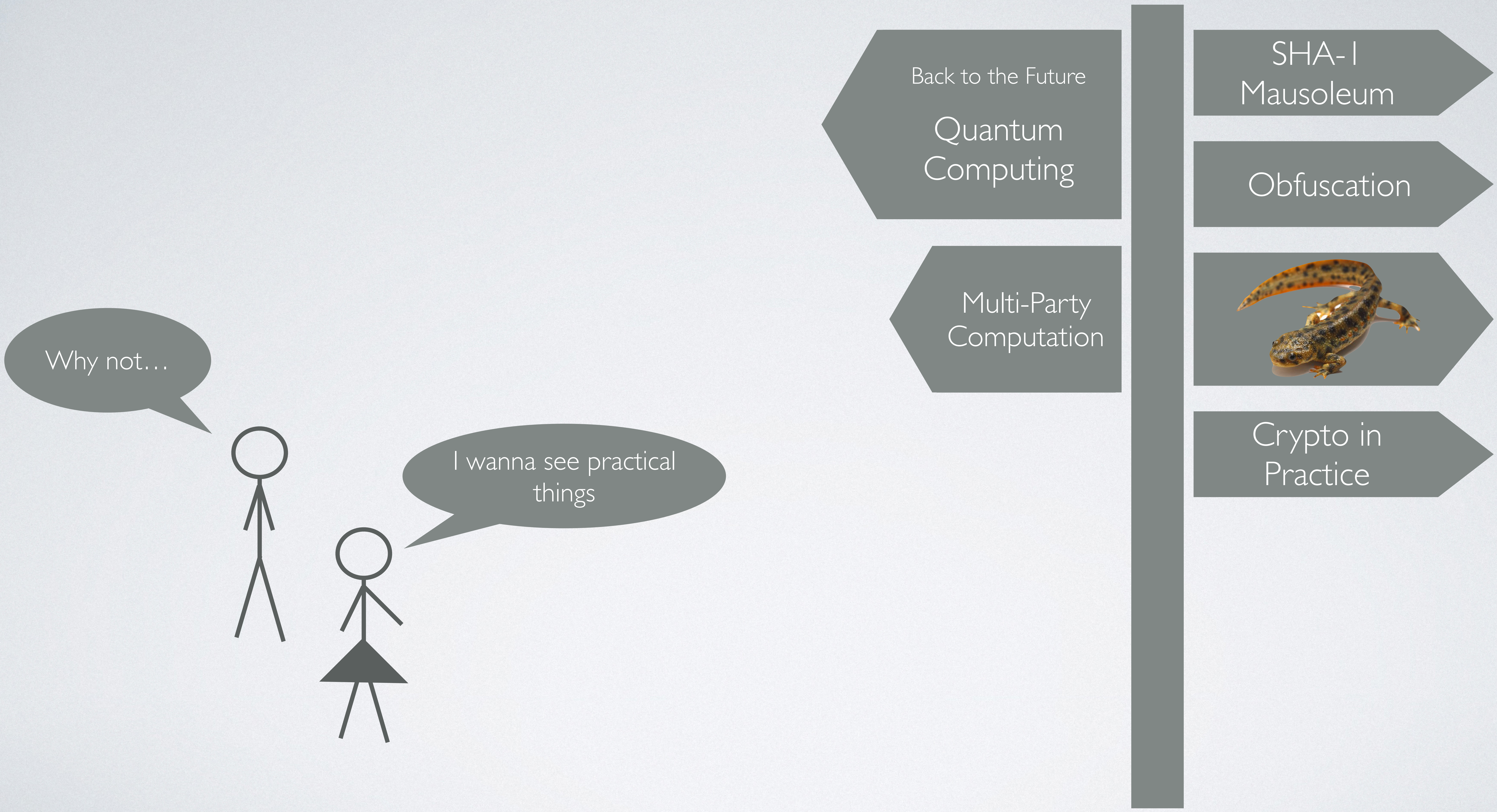
Siyao Guo
Northeastern University


Let's see
some crypto!


Fine…
Let's see some crypto!


Back to the Future
Quantum Computing
Multi-Party Computation
SHA-1 Mausoleum
Obfuscation
Crypto in Practice


Back to the Future
Quantum Computing
Multi-Party Computation
SHA-1 Mausoleum
Obfuscation
Crypto in Practice
I wanna see practical things


Why not…
I wanna see practical things
Back to the Future
Quantum Computing
Multi-Party Computation
SHA-1 Mausoleum
Obfuscation
Crypto in Practice

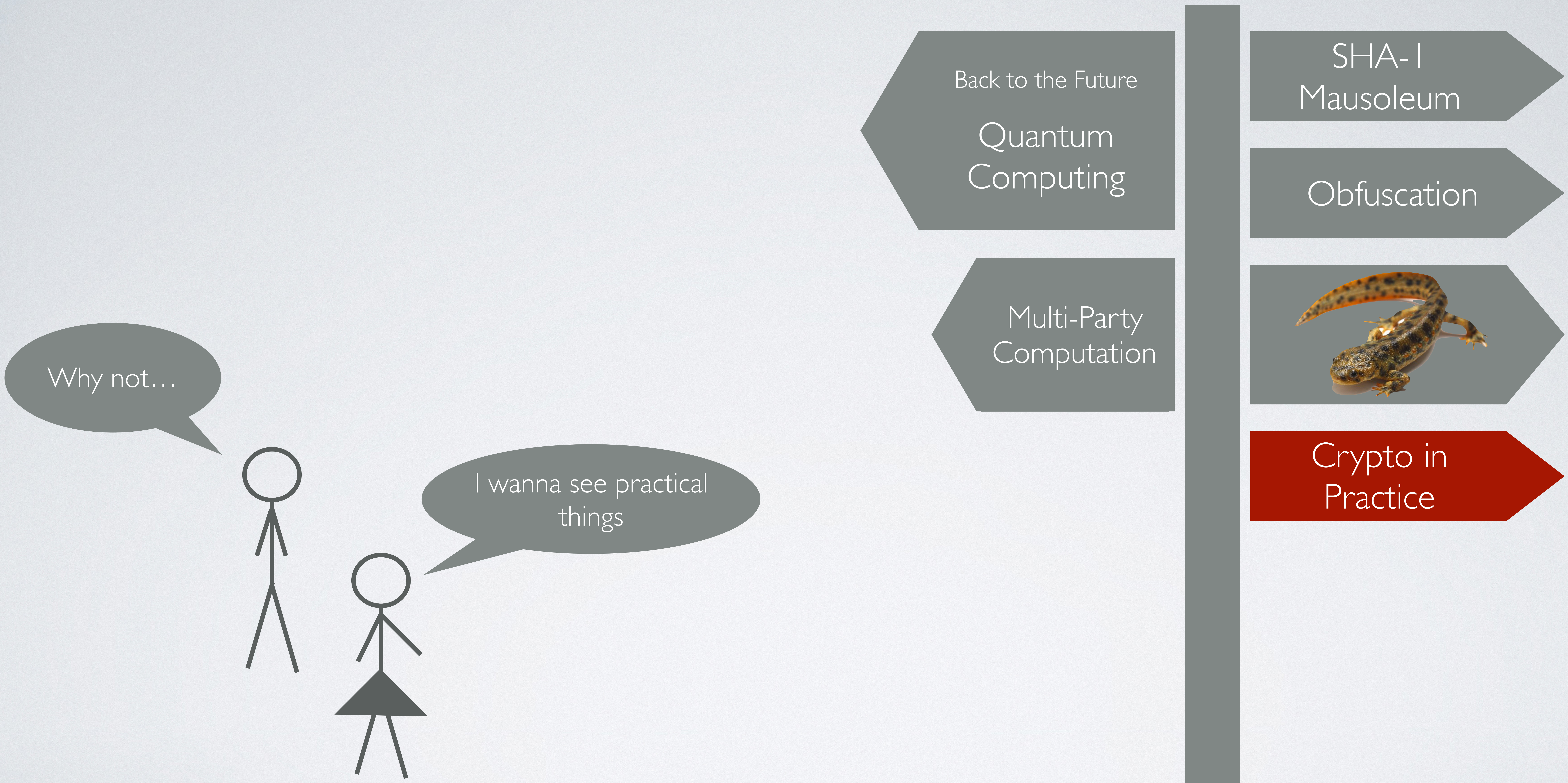
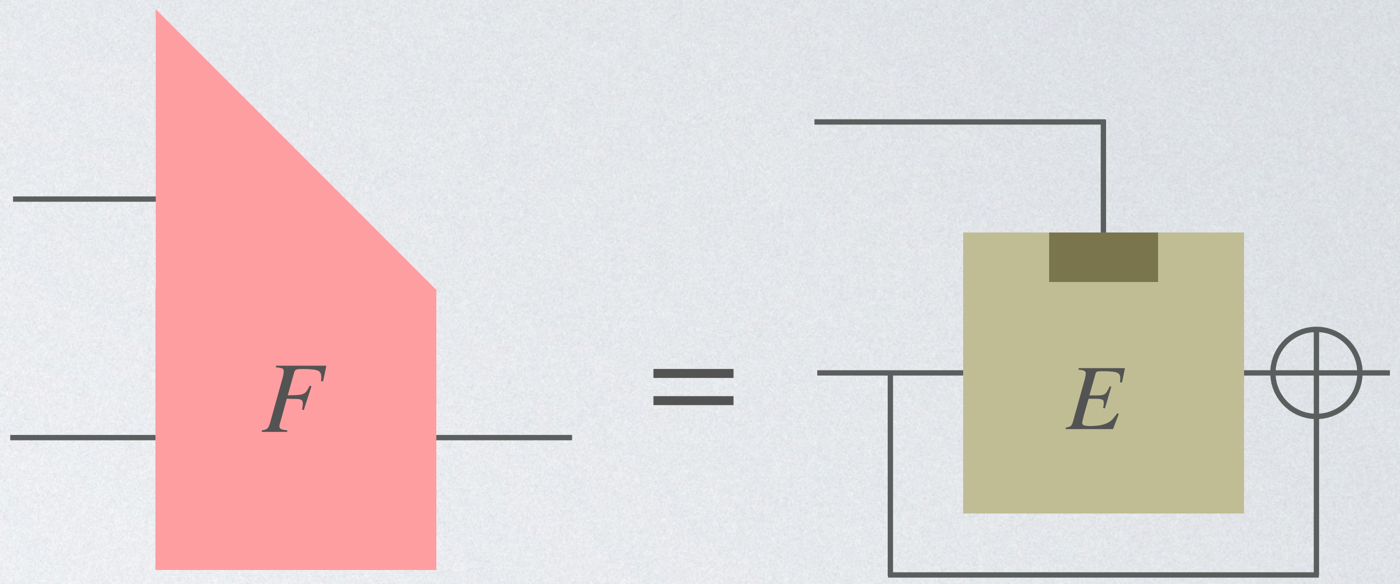
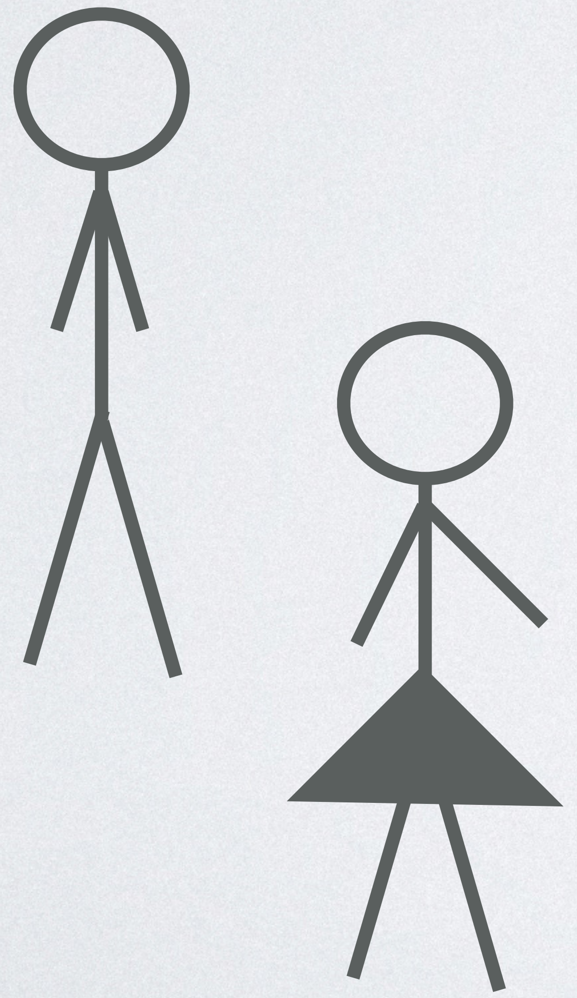

Why not…
I wanna see practical things
Back to the Future
Quantum Computing
Multi-Party Computation
SHA-1 Mausoleum
Obfuscation
Crypto in Practice

Exhibit A: Merkle-Damgard with Davies-Meyer (SHA-2)





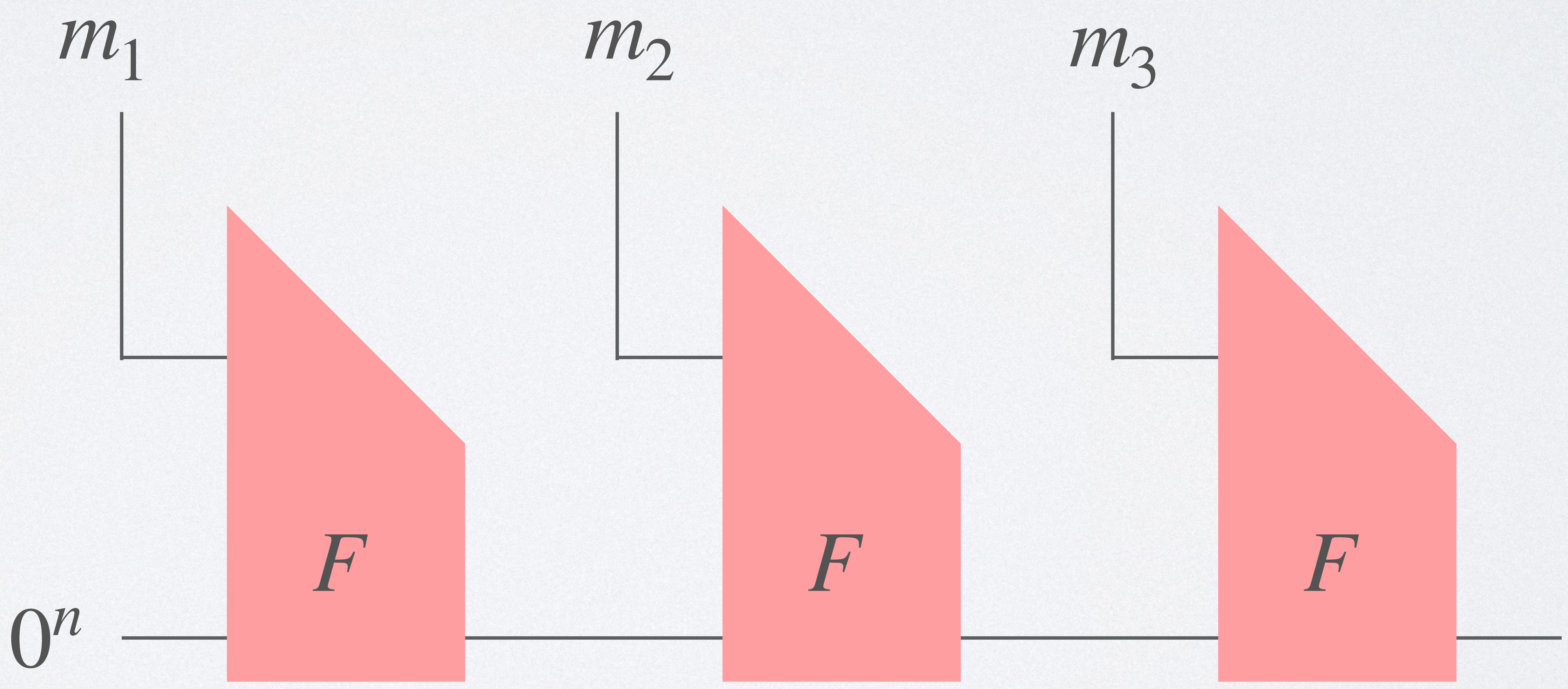
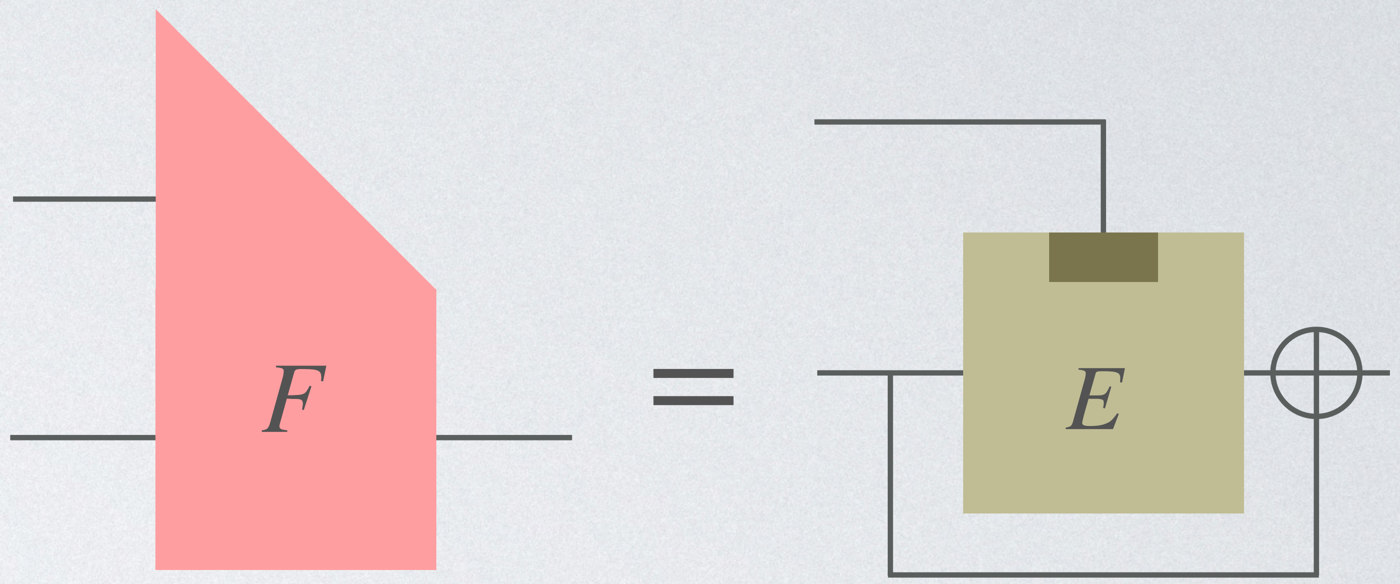
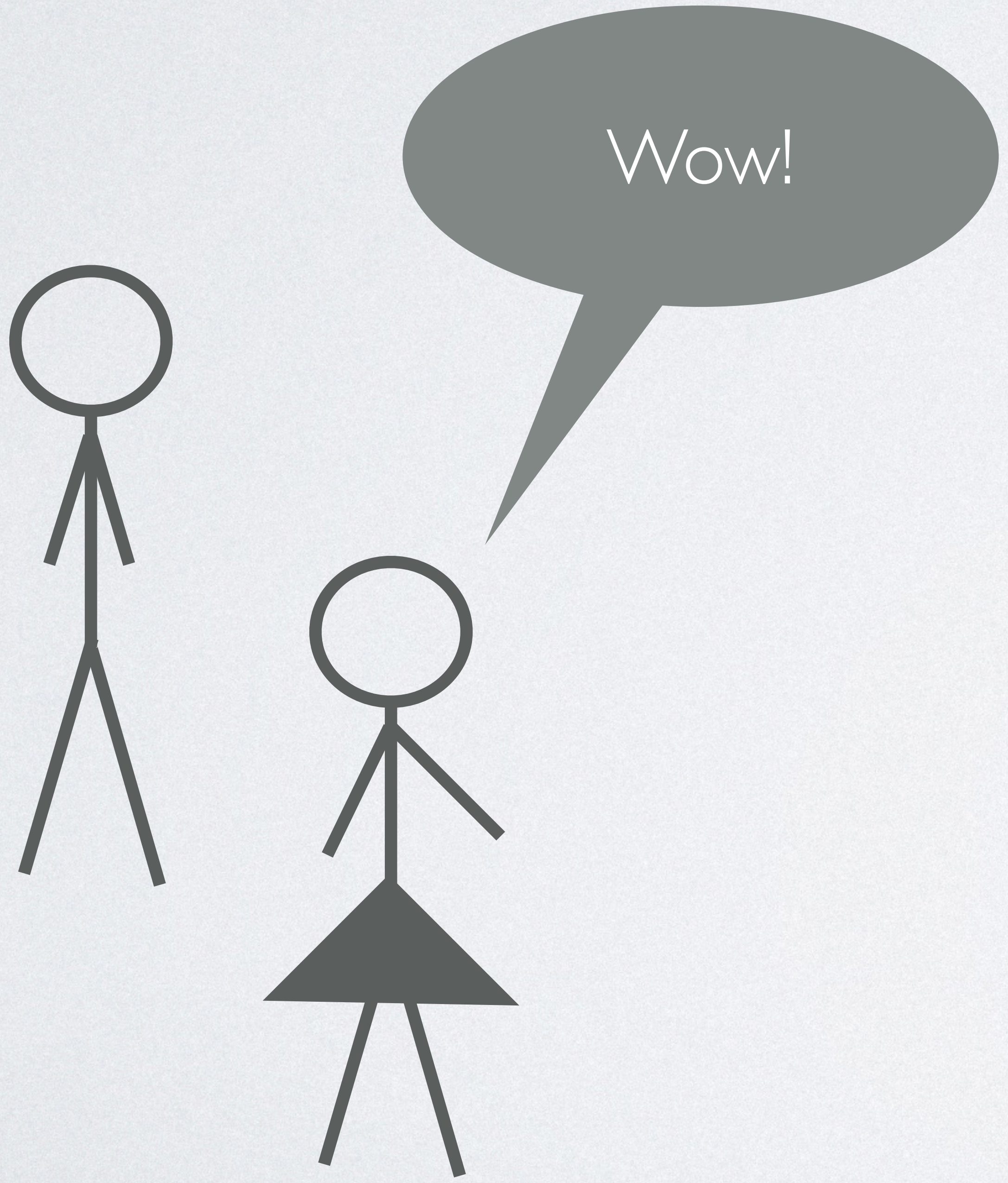


Exhibit A: Merkle-Damgard with Davies-Meyer (SHA-2)





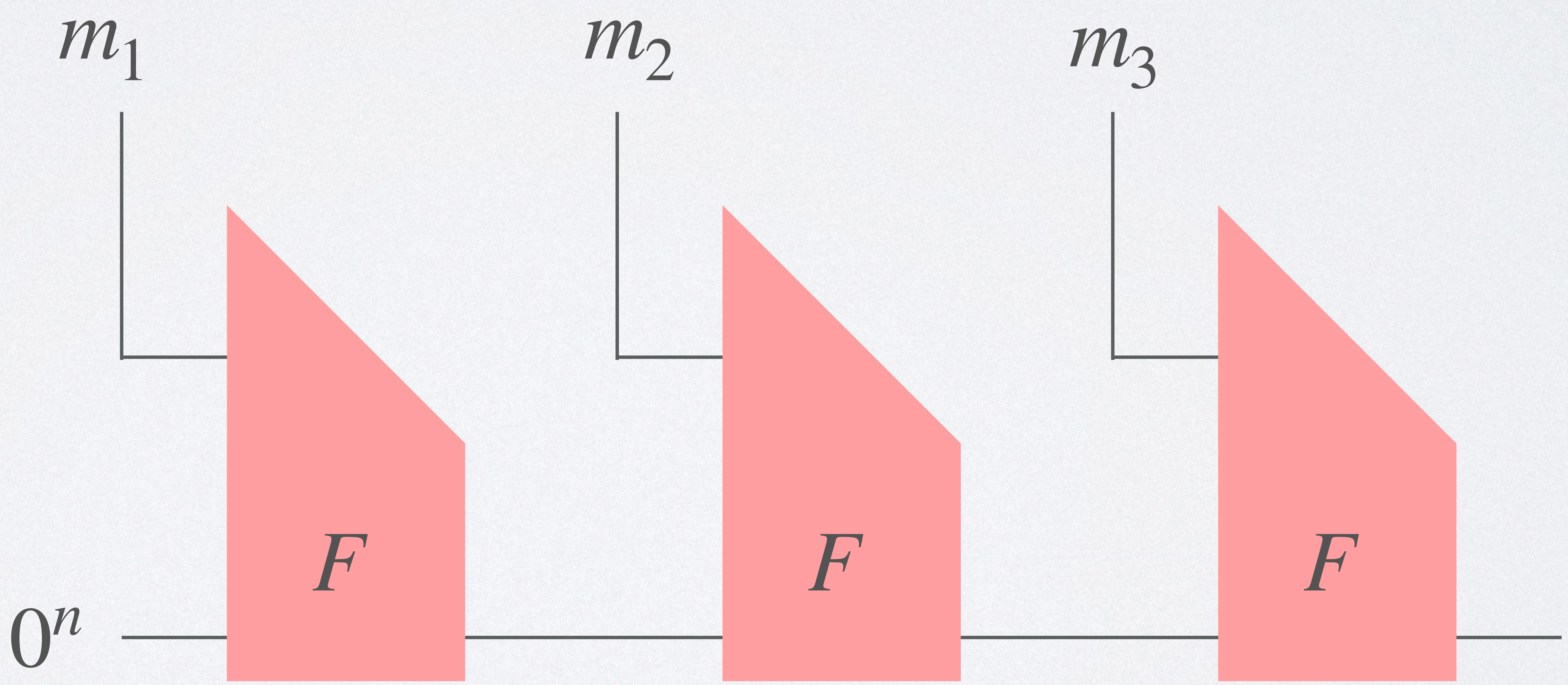
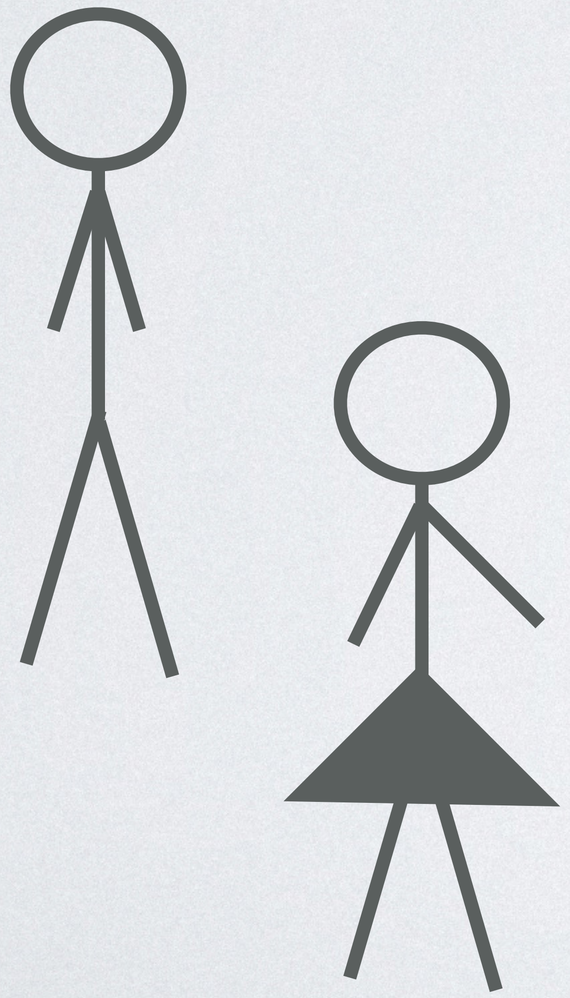
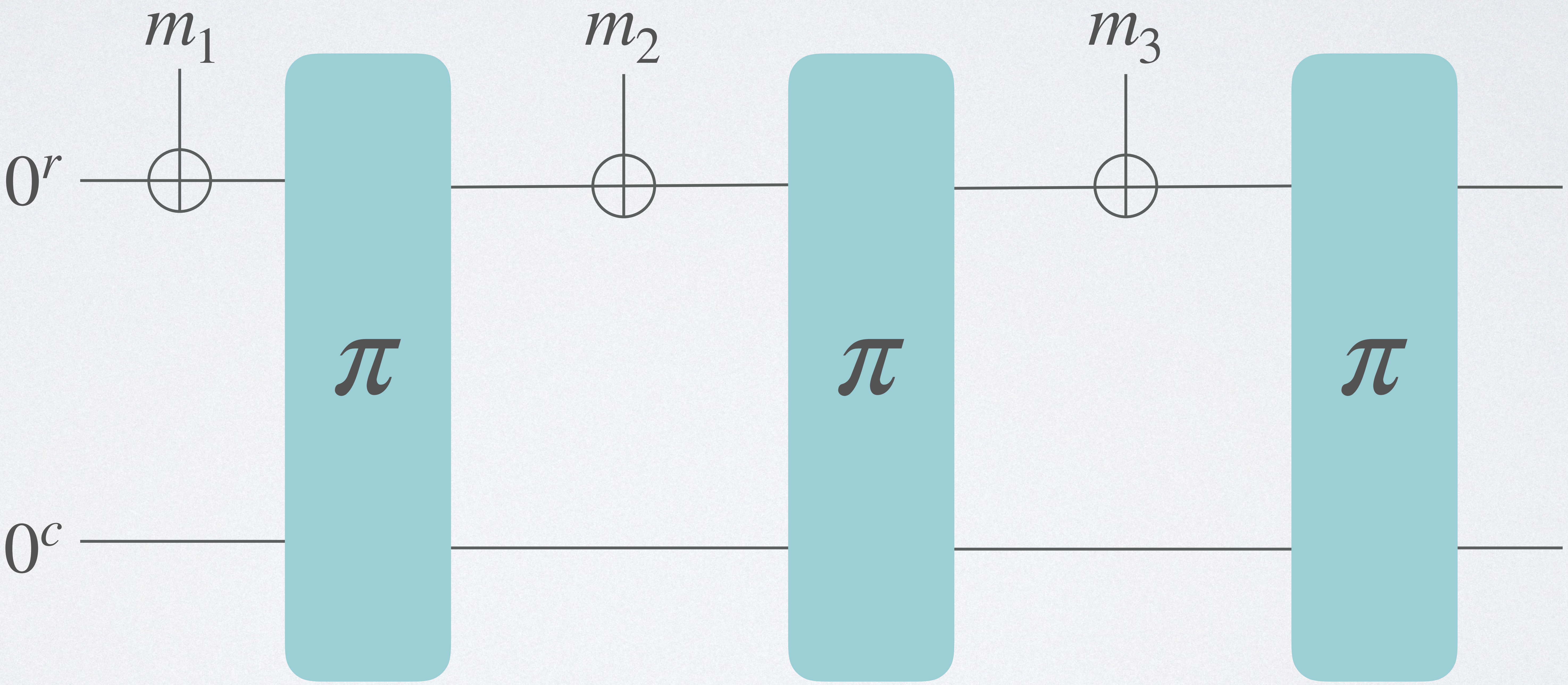


Exhibit 75: Sponge Construction (SHA-3)

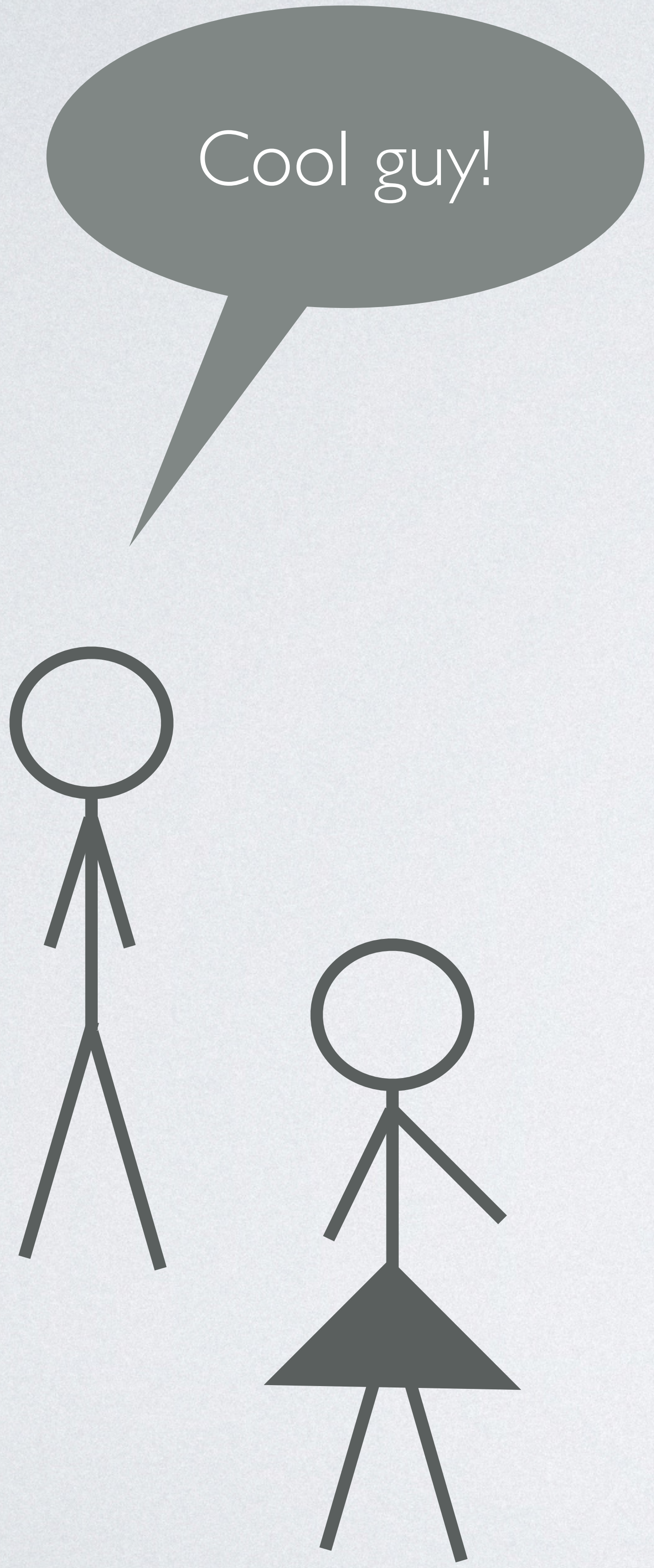


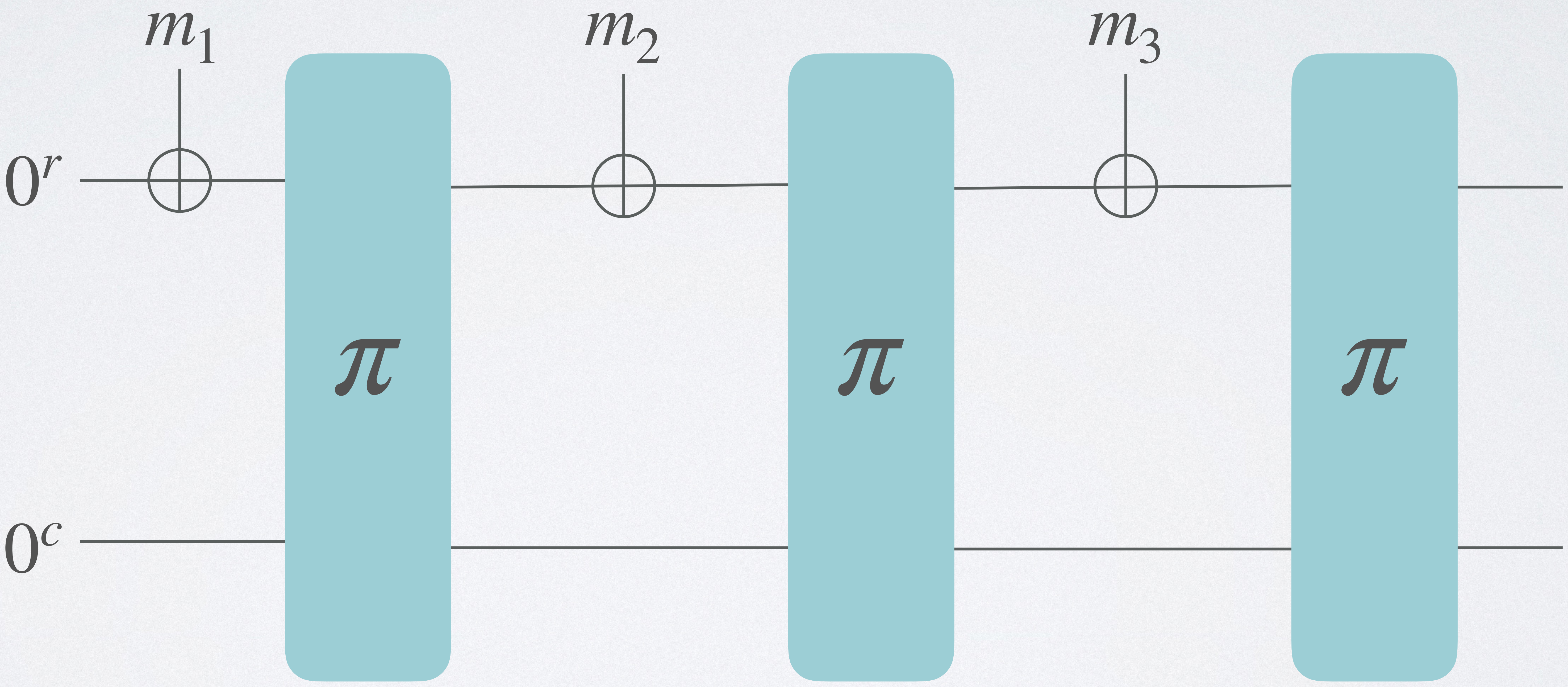


Used as:

- CRHF
- MAC
- PRF
- PRNG
- etc.

Exhibit 75: Sponge Construction (SHA-3)

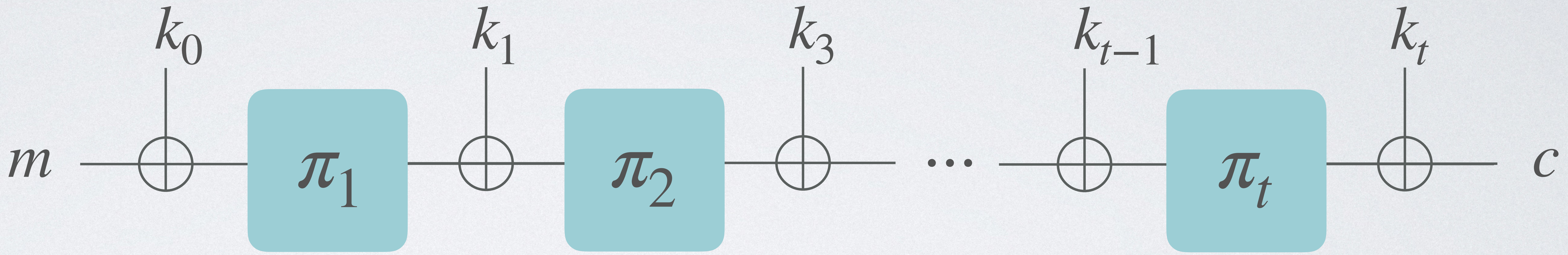


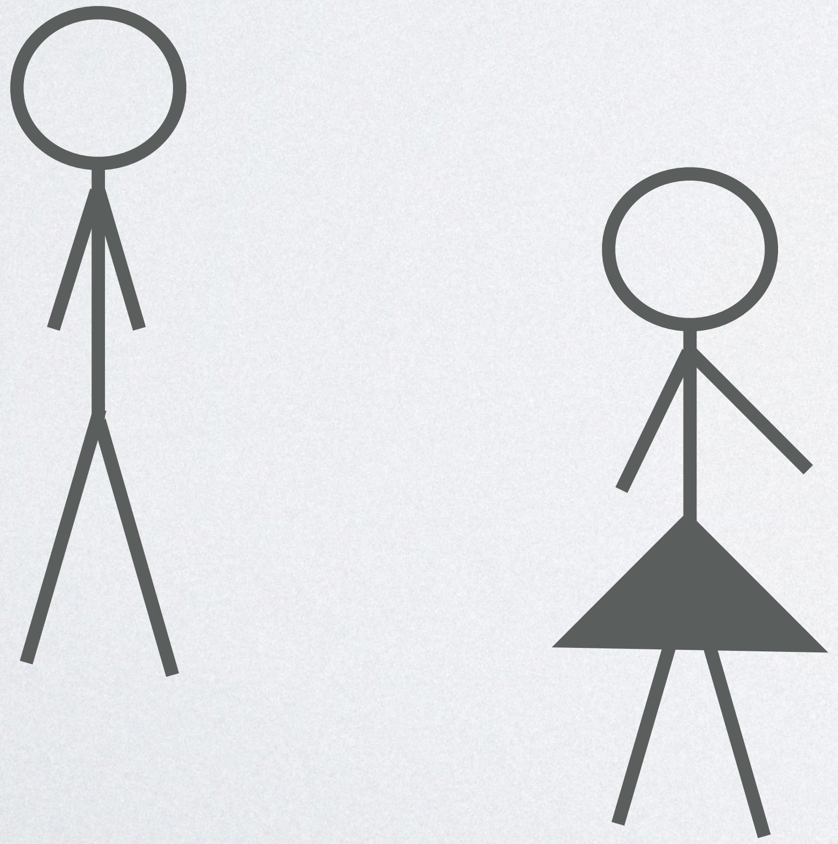


Used as:

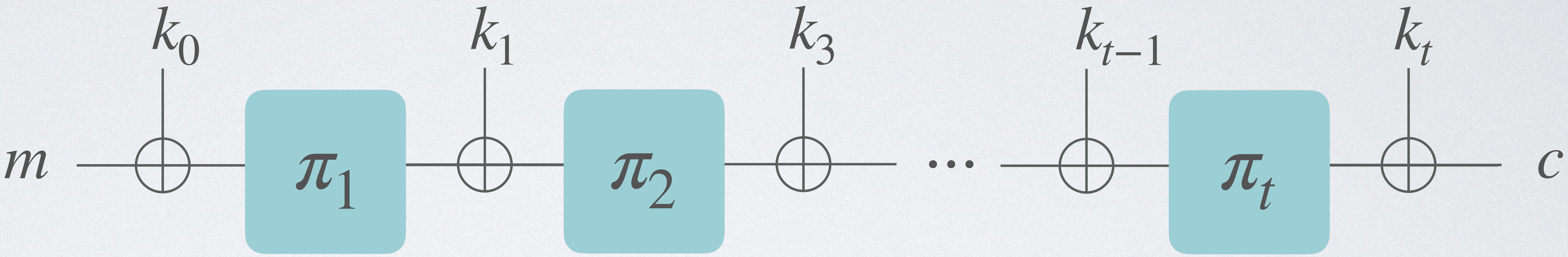
- CRHF
- MAC
- PRF
- PRNG
- etc.

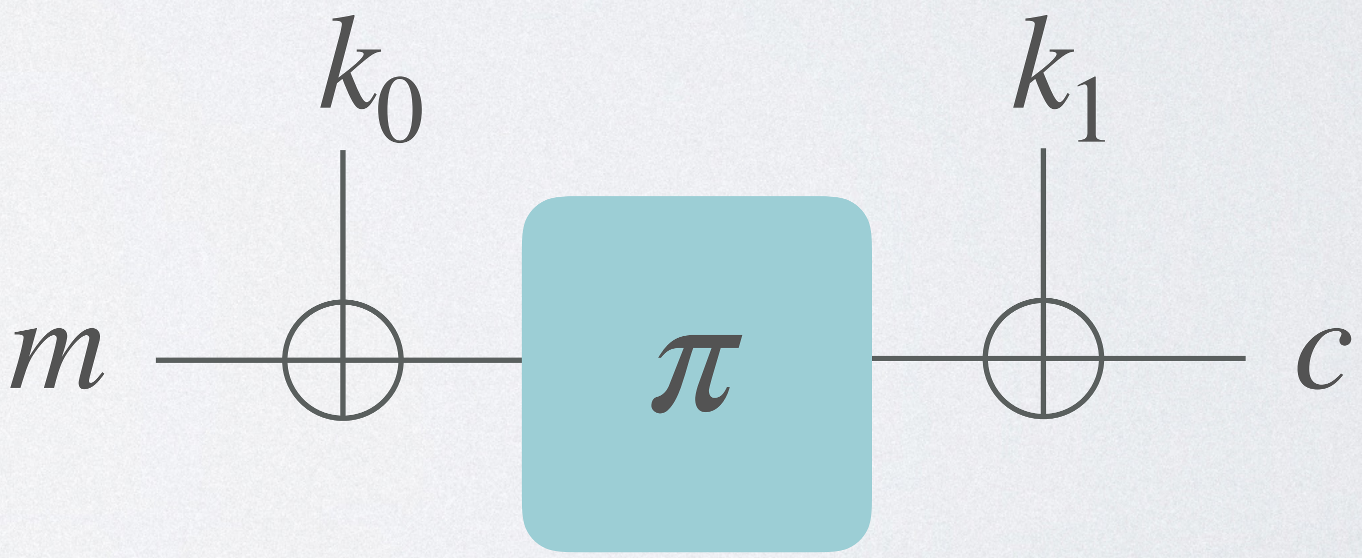
Item E12: Key-Alternating Ciphers (AES)



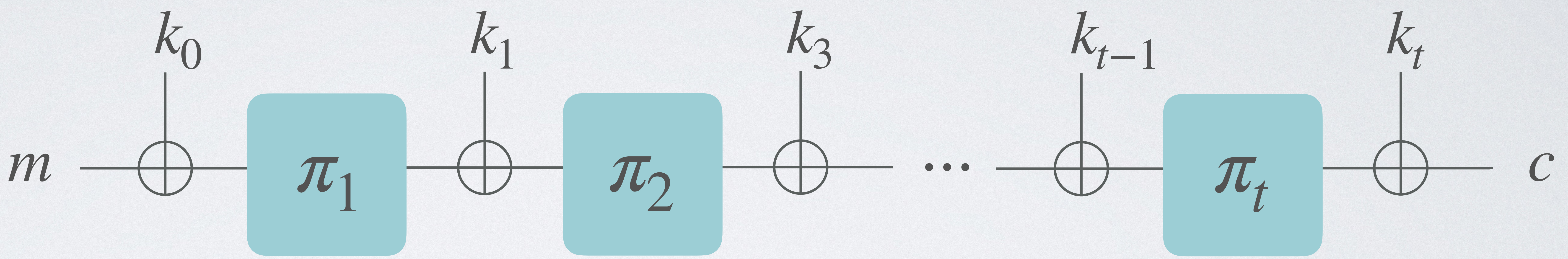


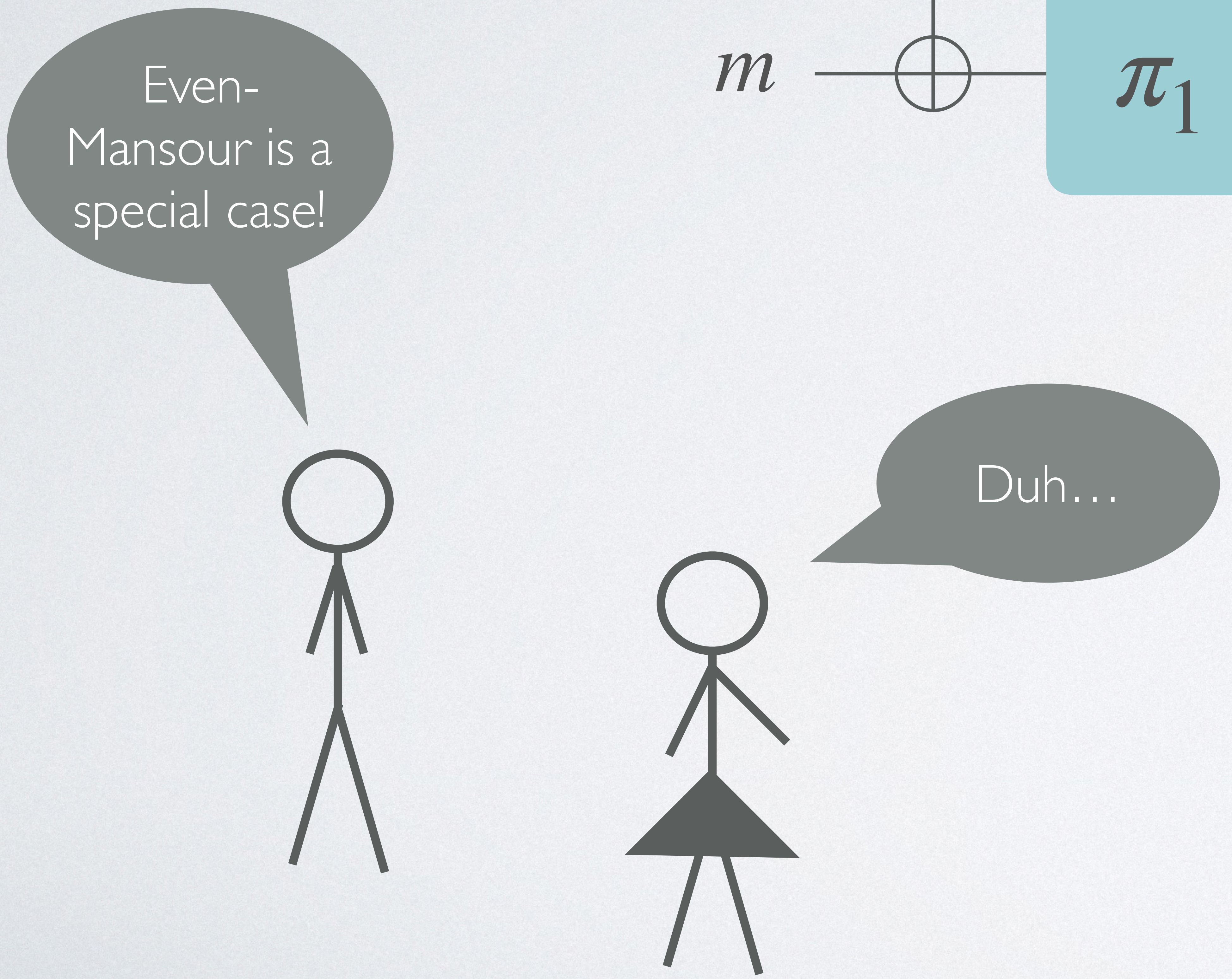
Item E12: Key-Alternating Ciphers (AES)





Item E12: Key-Alternating Ciphers (AES)





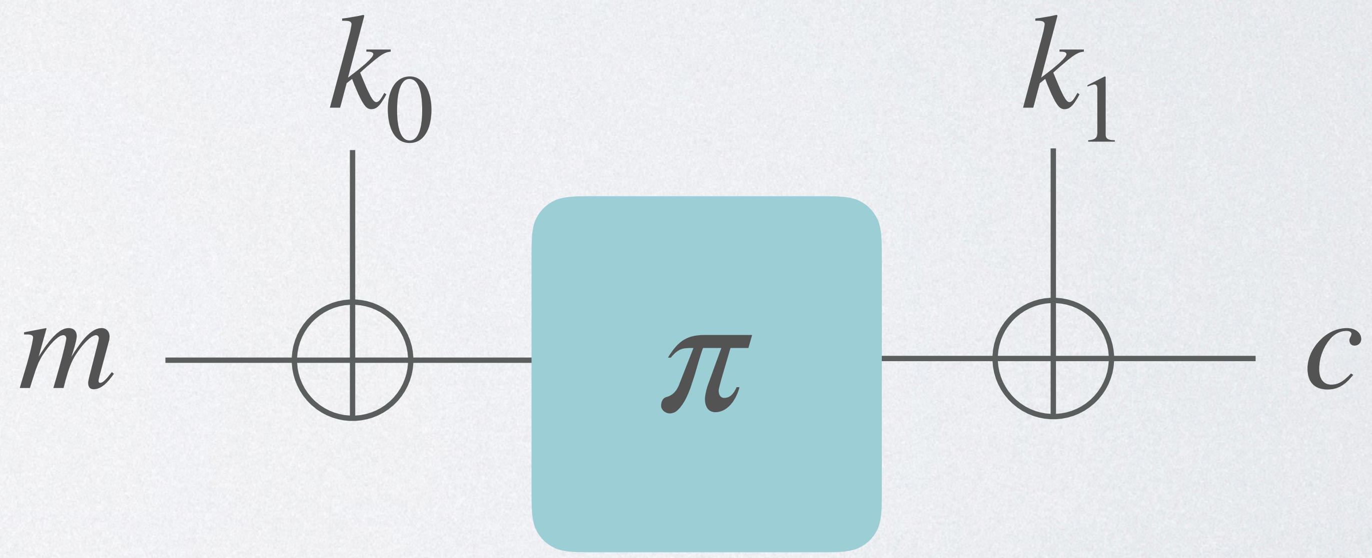


Exhibit S: Discrete Logarithms

Cyclic group $G = \langle g \rangle$ of order N

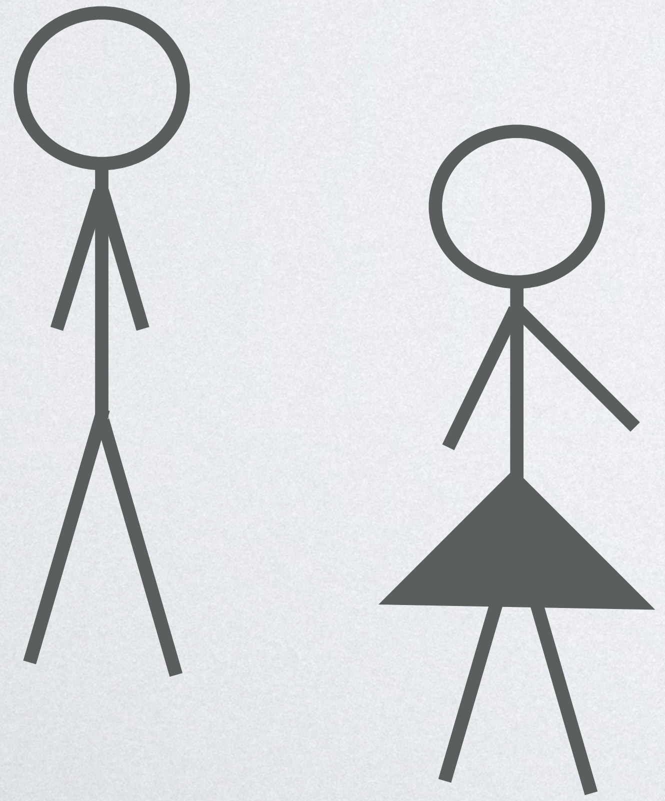
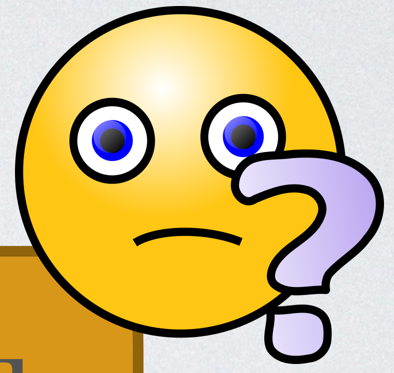


Exhibit S: Discrete Logarithms

Cyclic group $G = \langle g \rangle$ of order N

Exhibit S: Discrete Logarithms

Cyclic group $G = \langle g \rangle$ of order N



$x \leftarrow [N]$

$y = g^x$

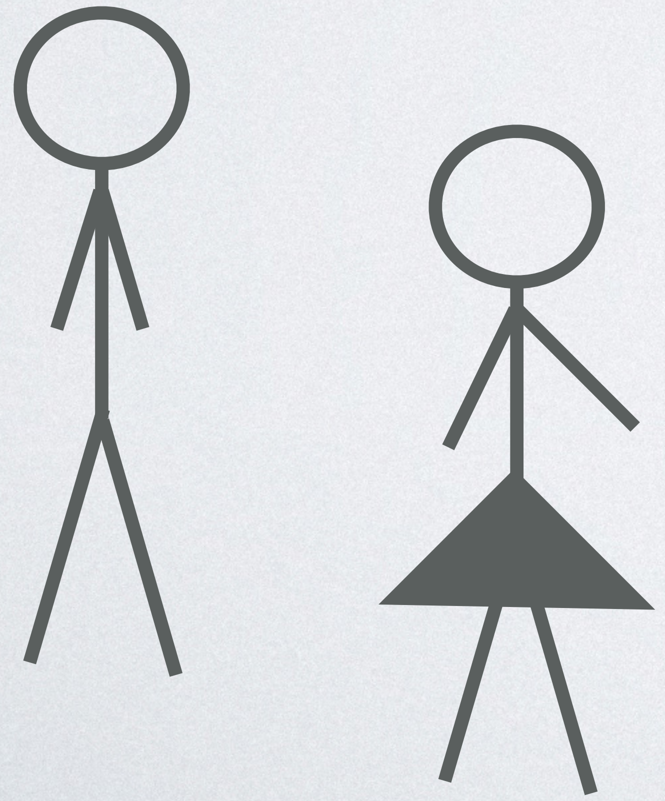


Exhibit S: Discrete Logarithms

Cyclic group $G = \langle g \rangle$ of order N

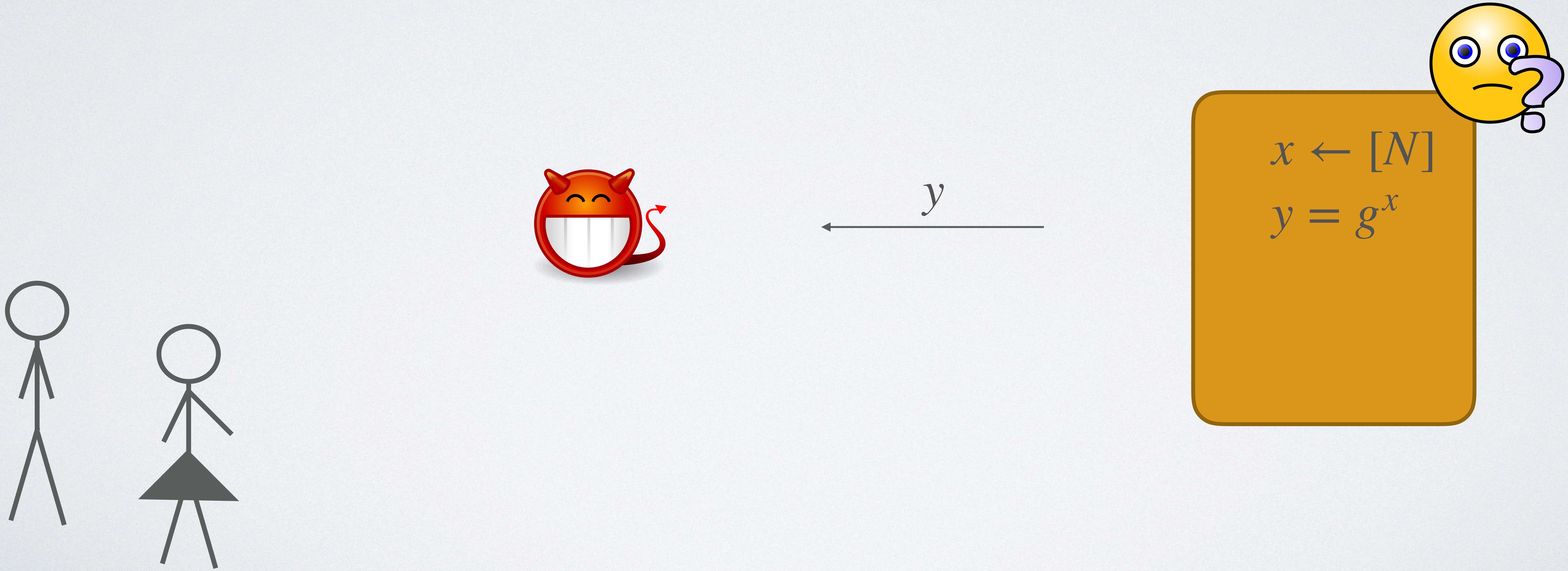


Exhibit S: Discrete Logarithms

Cyclic group $G = \langle g \rangle$ of order N

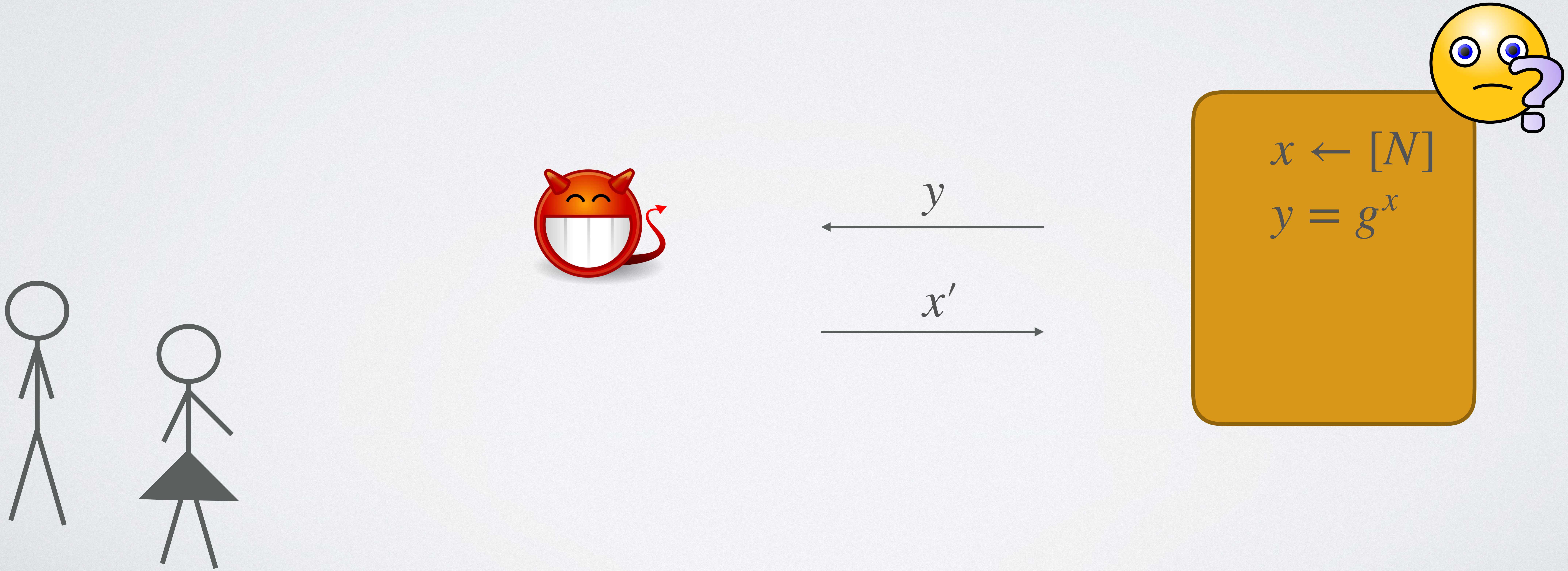
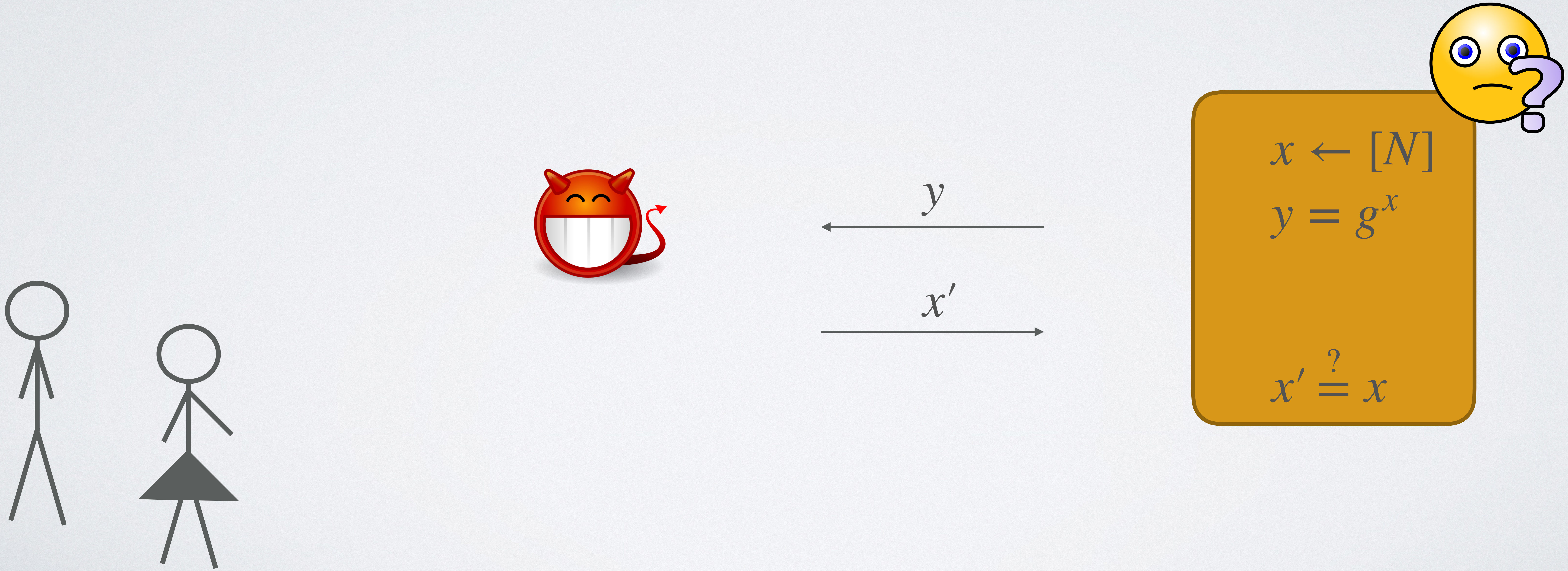
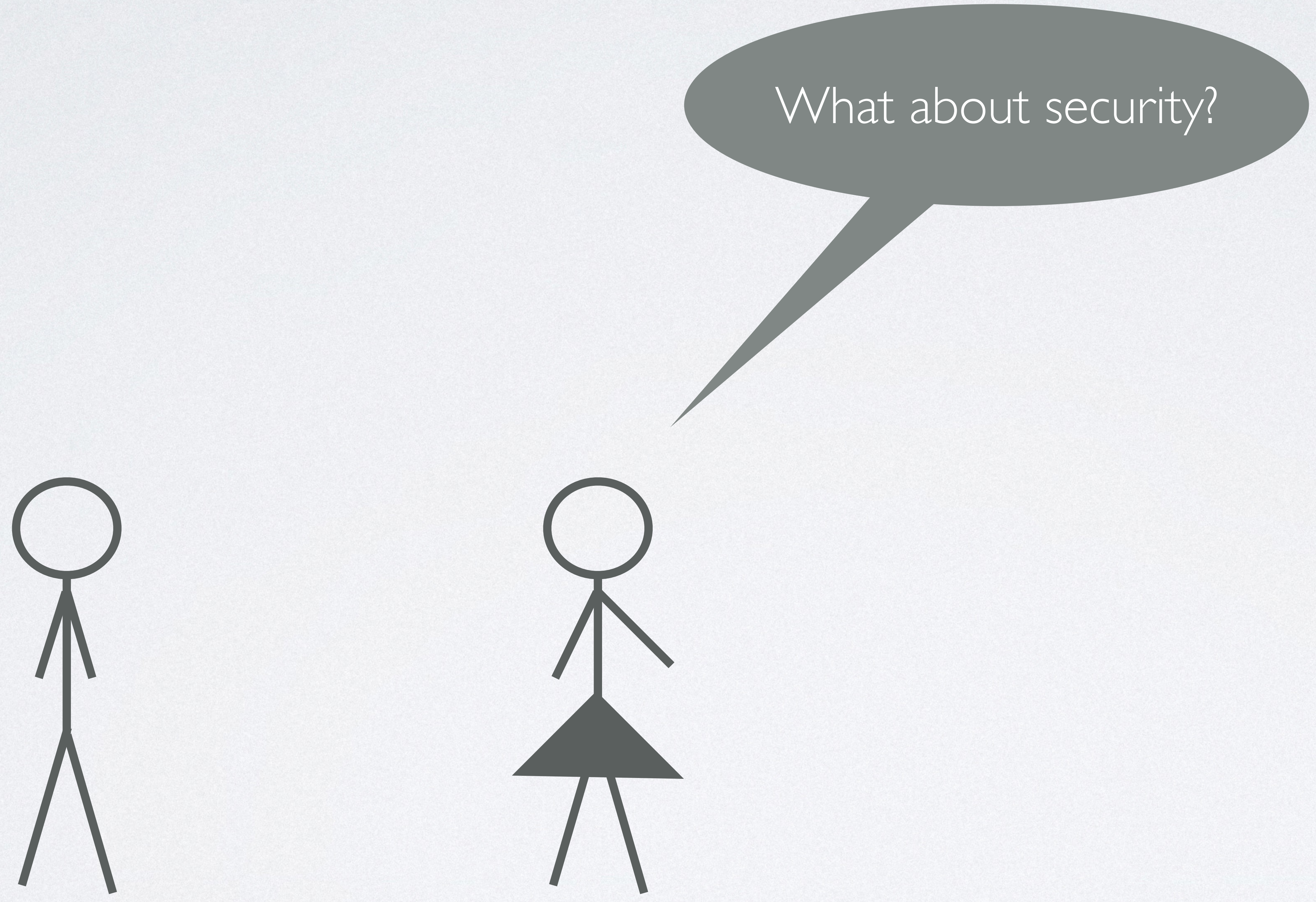
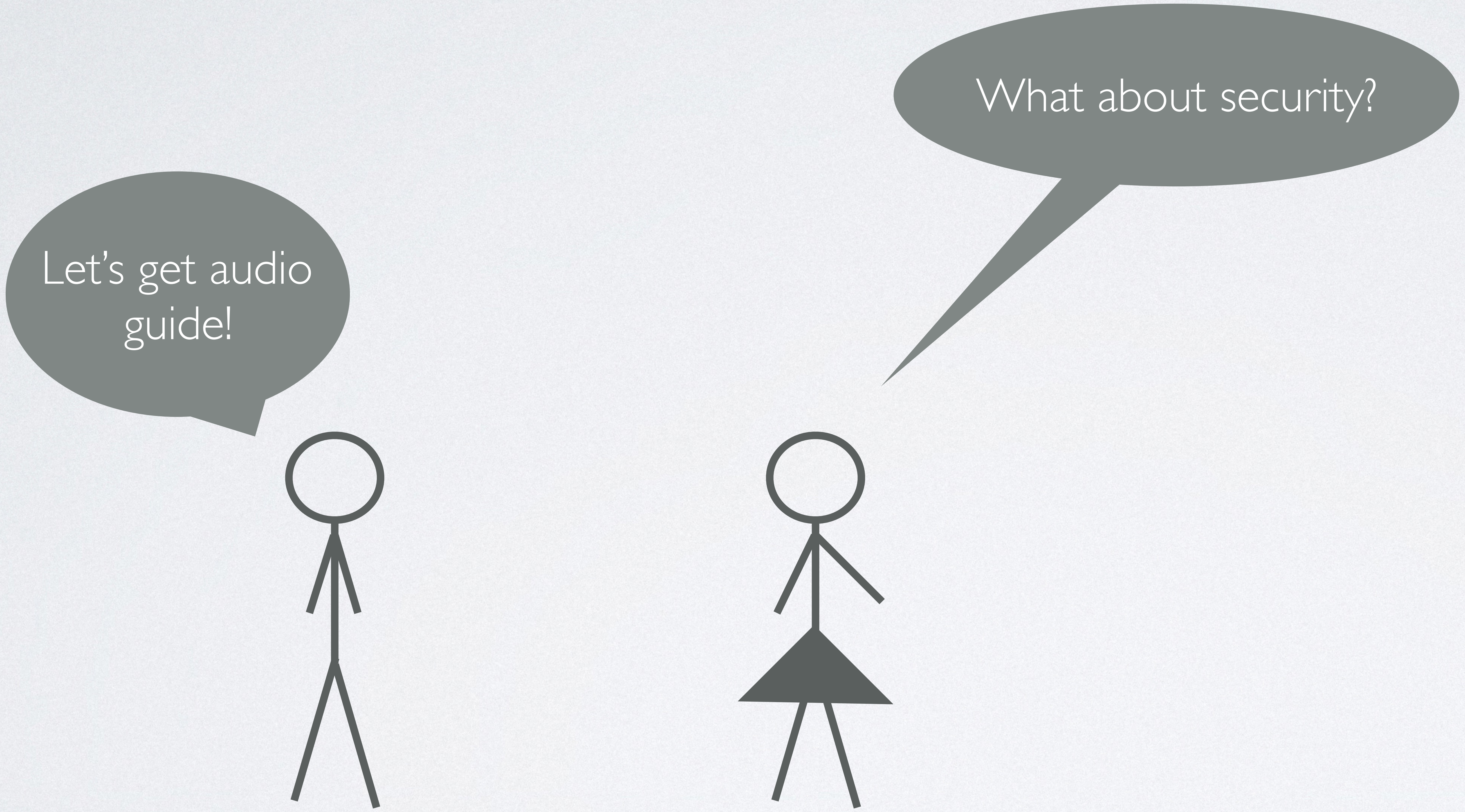


Exhibit S: Discrete Logarithms

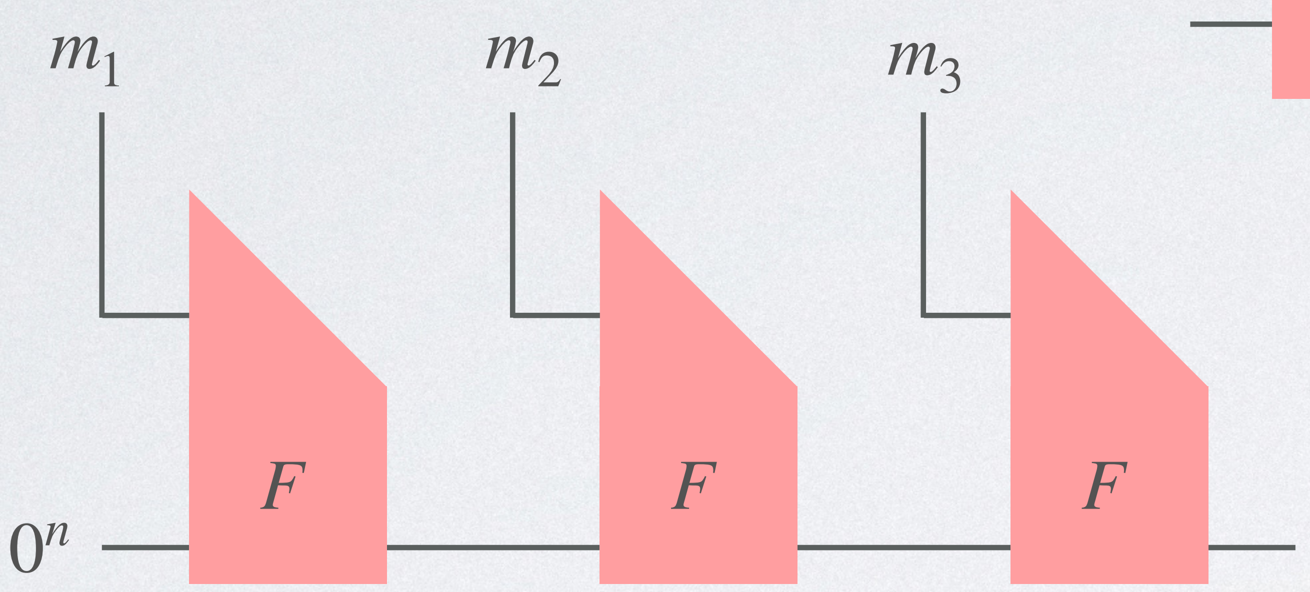
Cyclic group $G = \langle g \rangle$ of order N

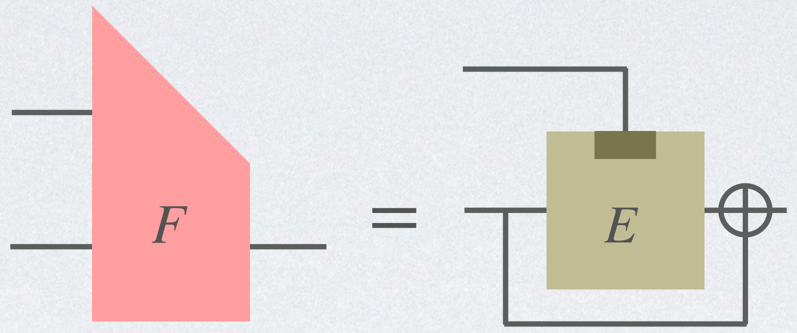


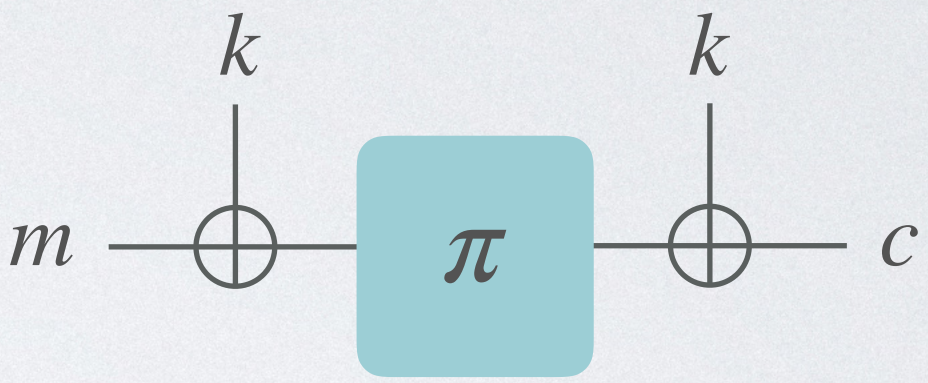

What about security?

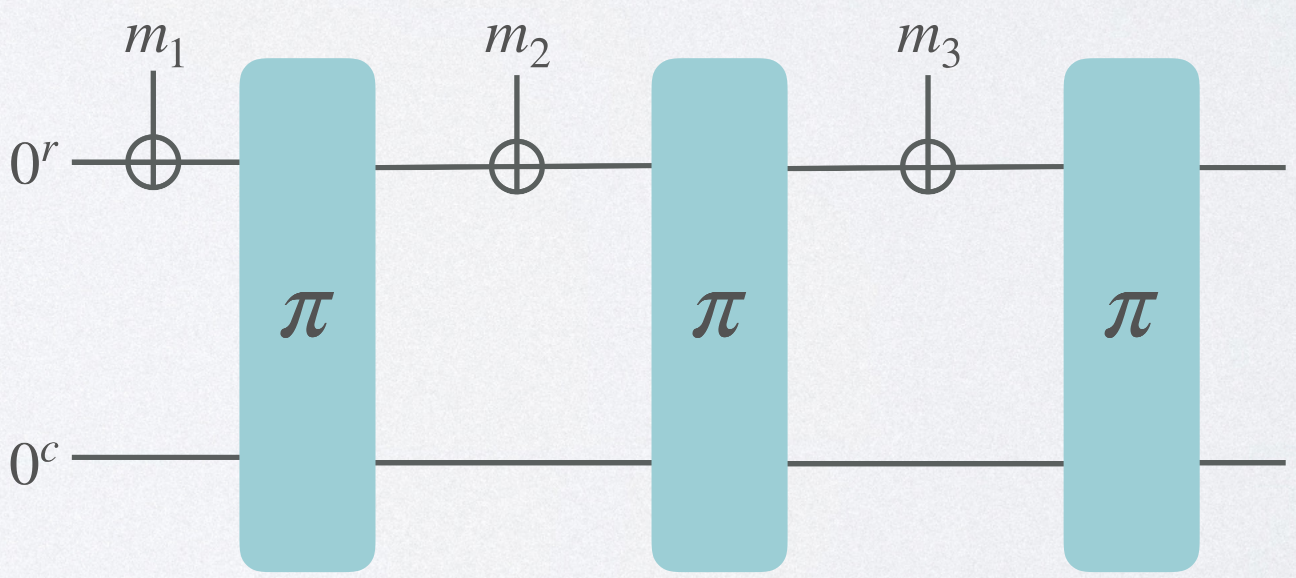

Let's get audio guide!
What about security?

Symmetric Crytpography

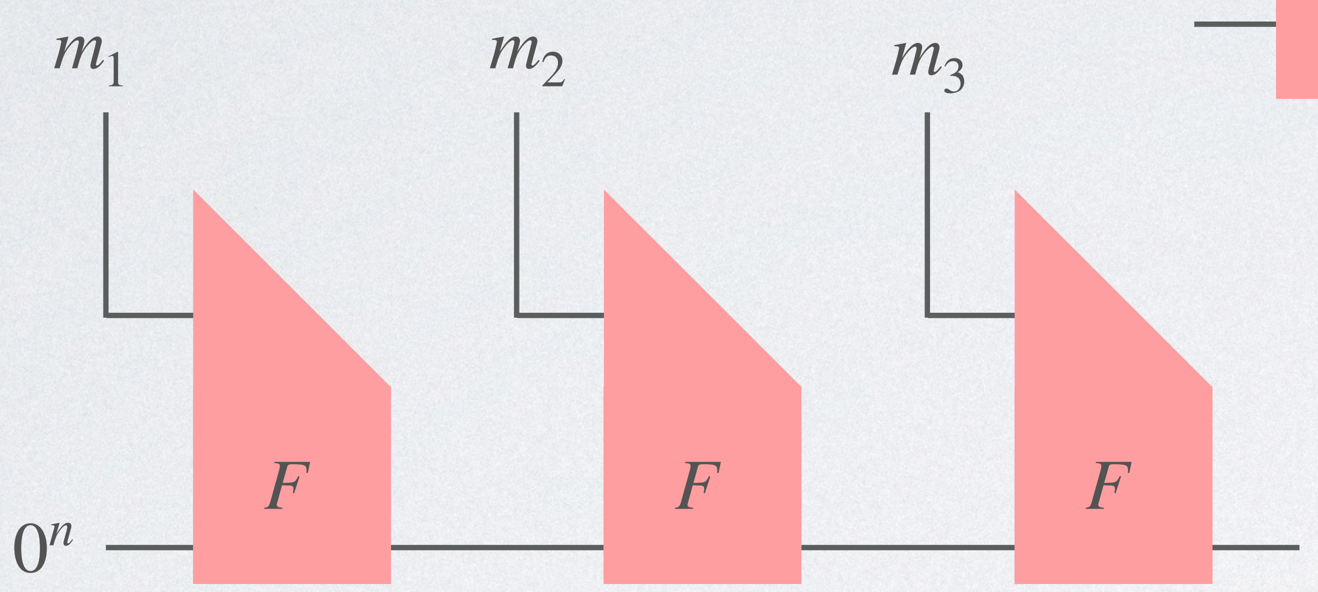


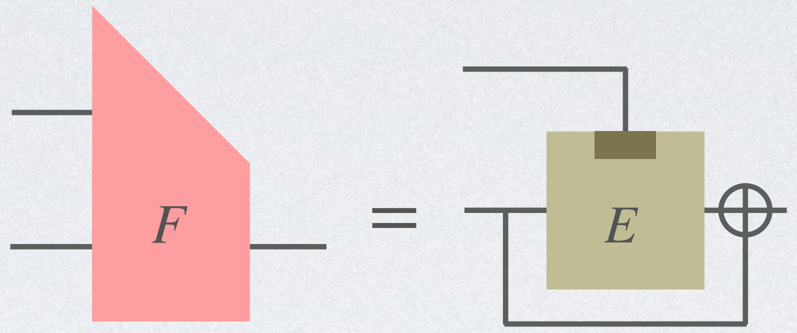


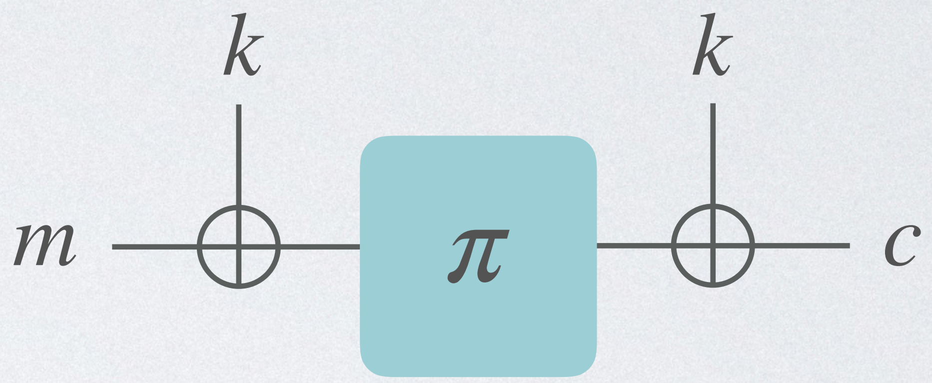


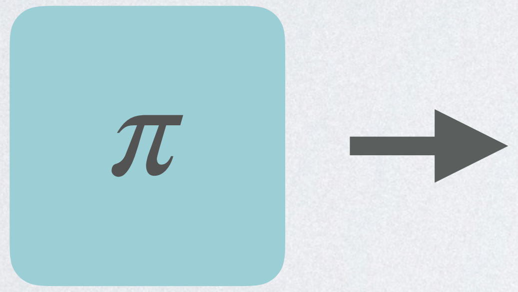


Symmetric Crytpography

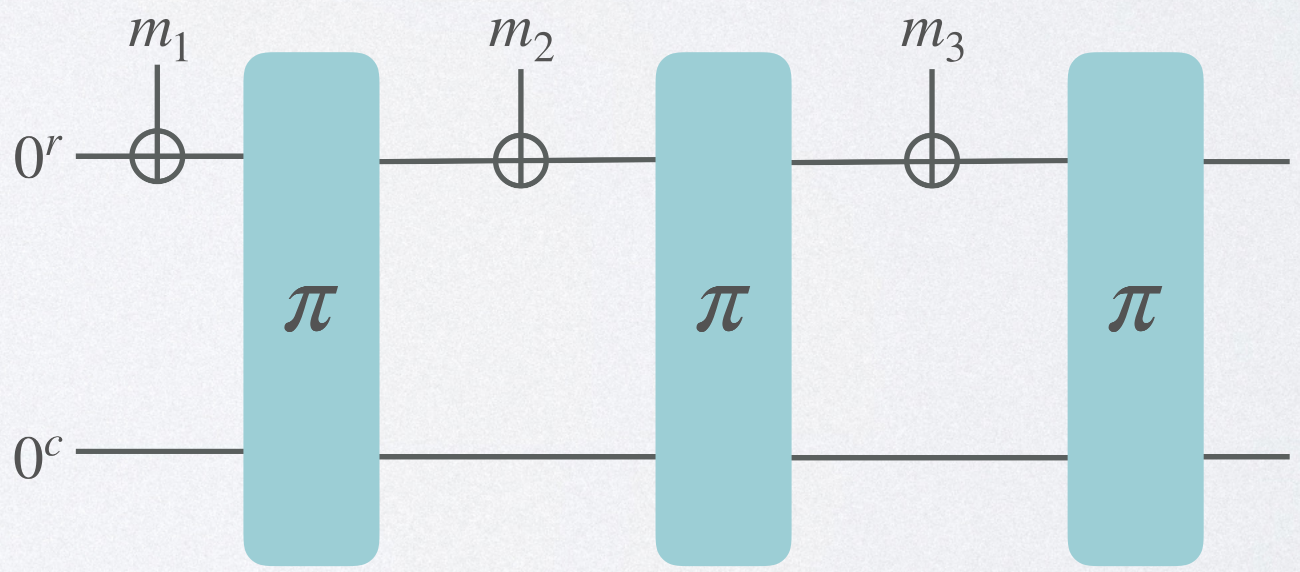




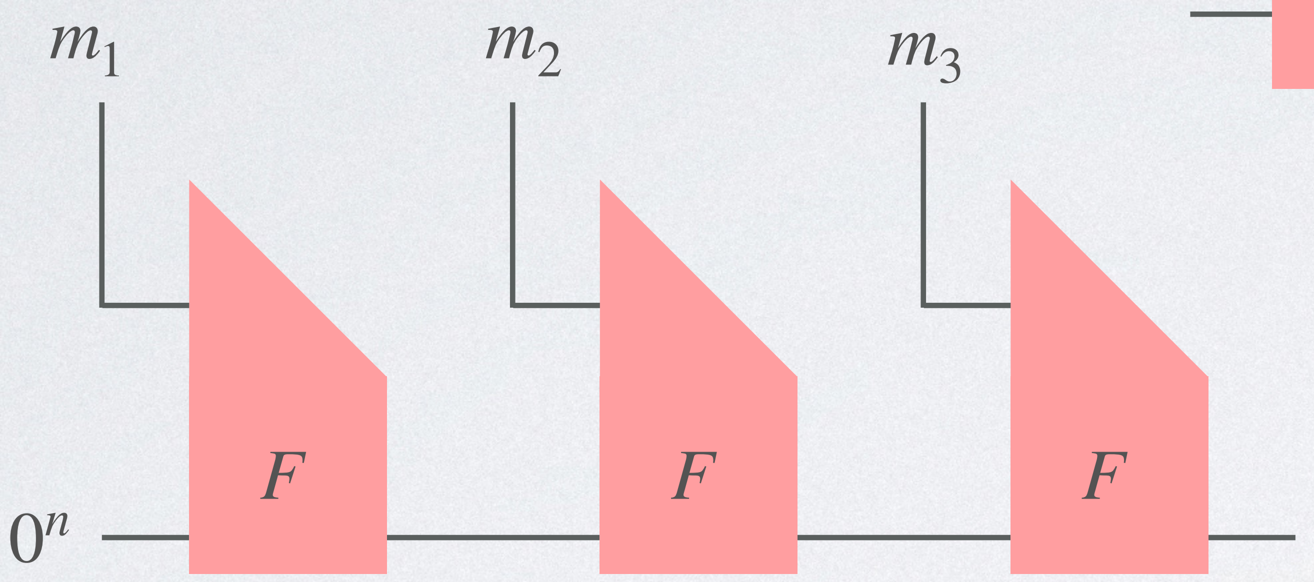


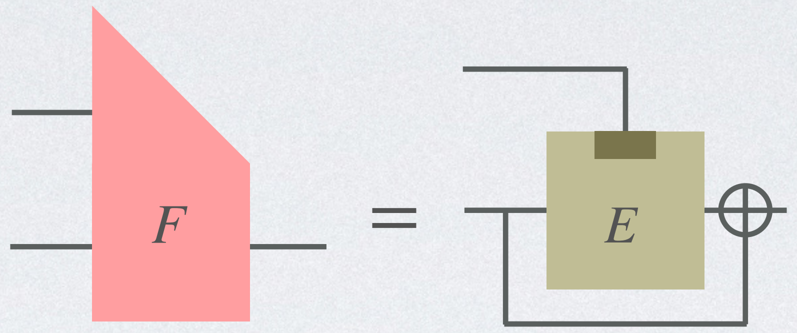


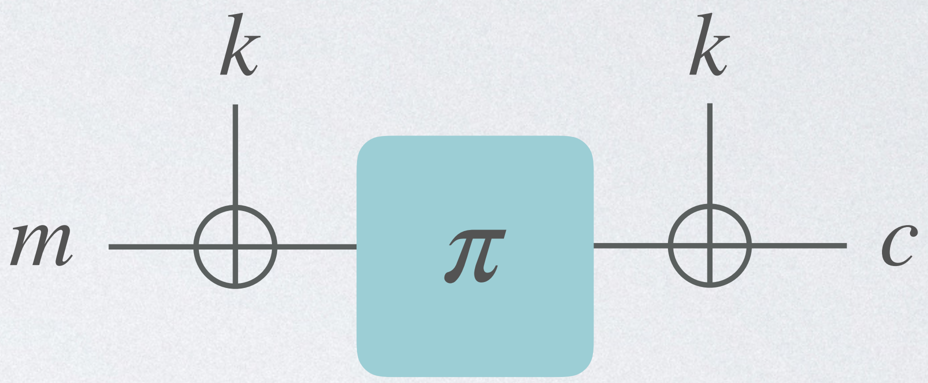
Random Permutation

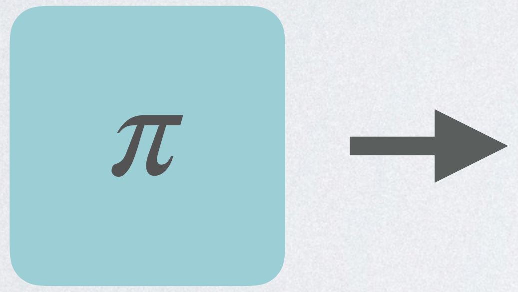


Symmetric Crytpography



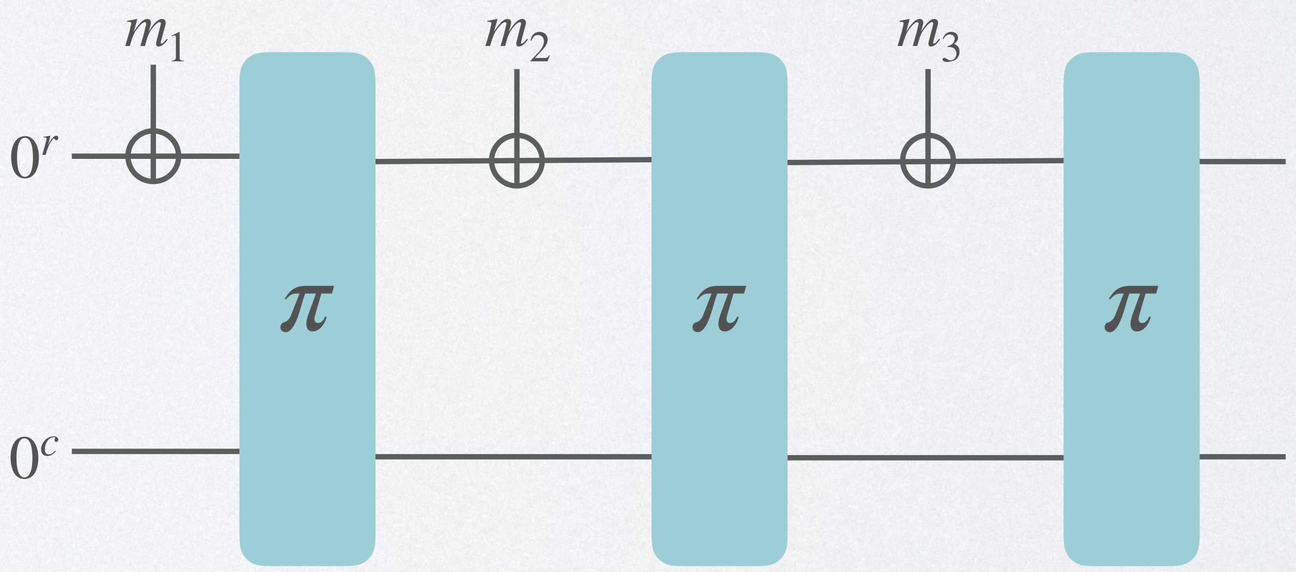






Random Permutation

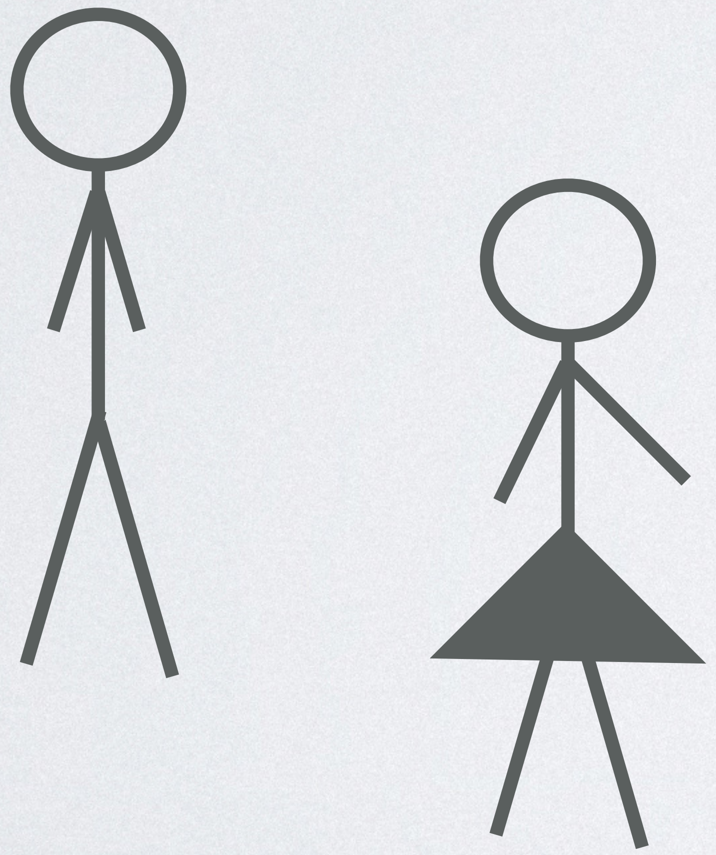
Ideal Cipher



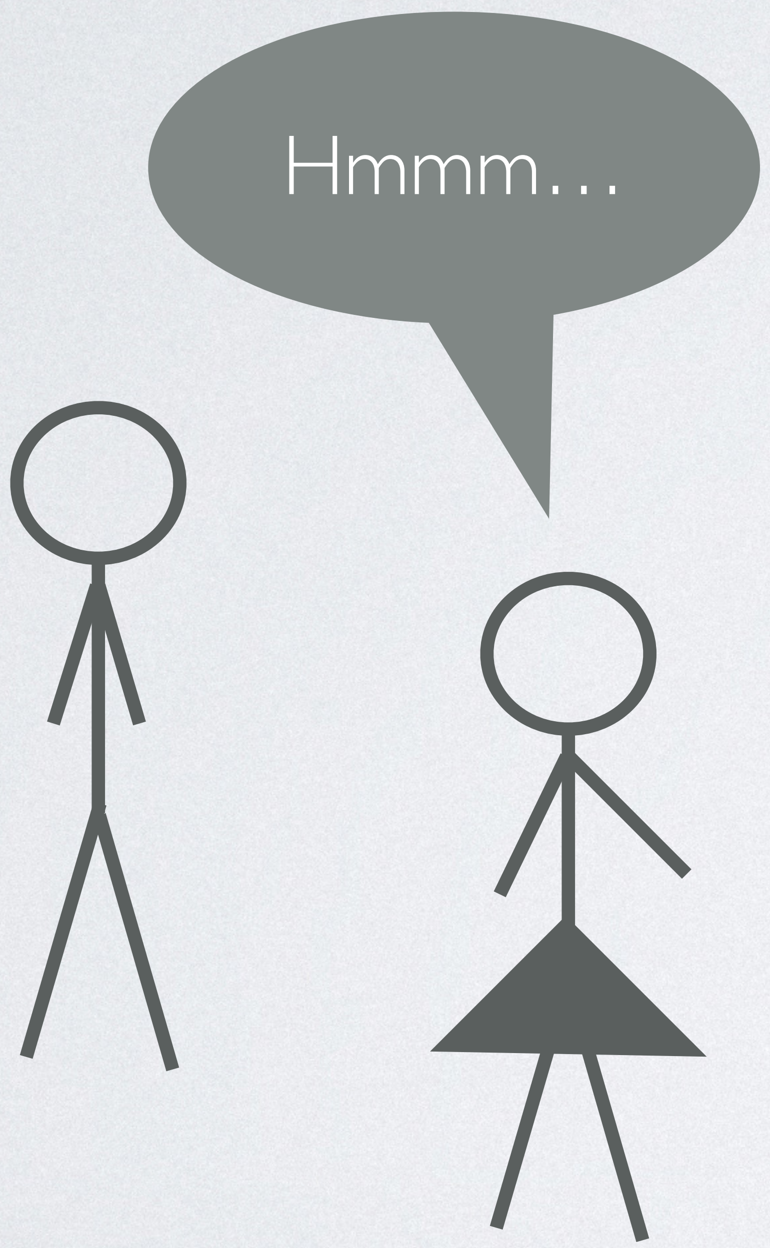
Idealized-Model Methodology

For "natural" applications:

Security in idealized model
=
Security in standard model
using best possible instantiation



Idealized-Model Methodology

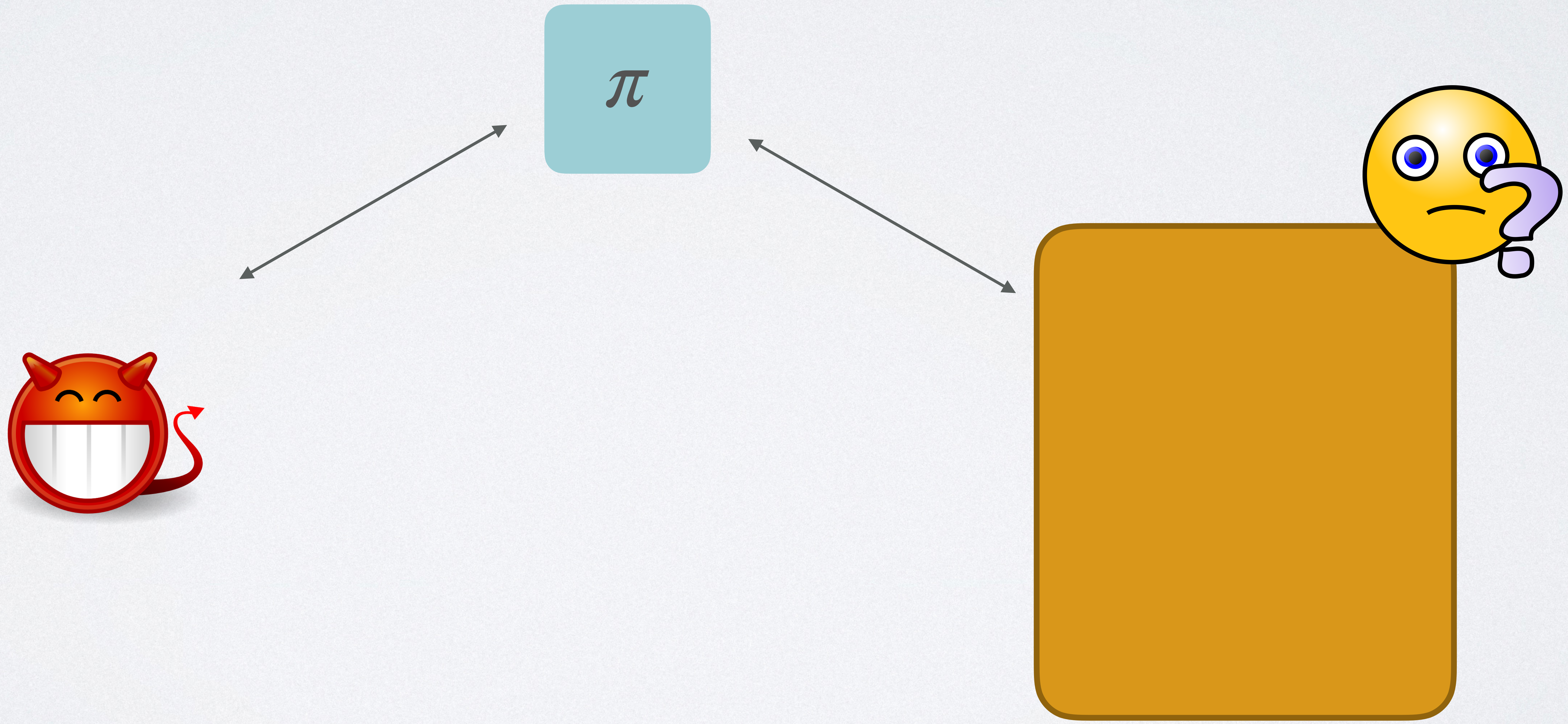


For "natural" applications:

Security in idealized model
=
Security in standard model
using best possible instantiation

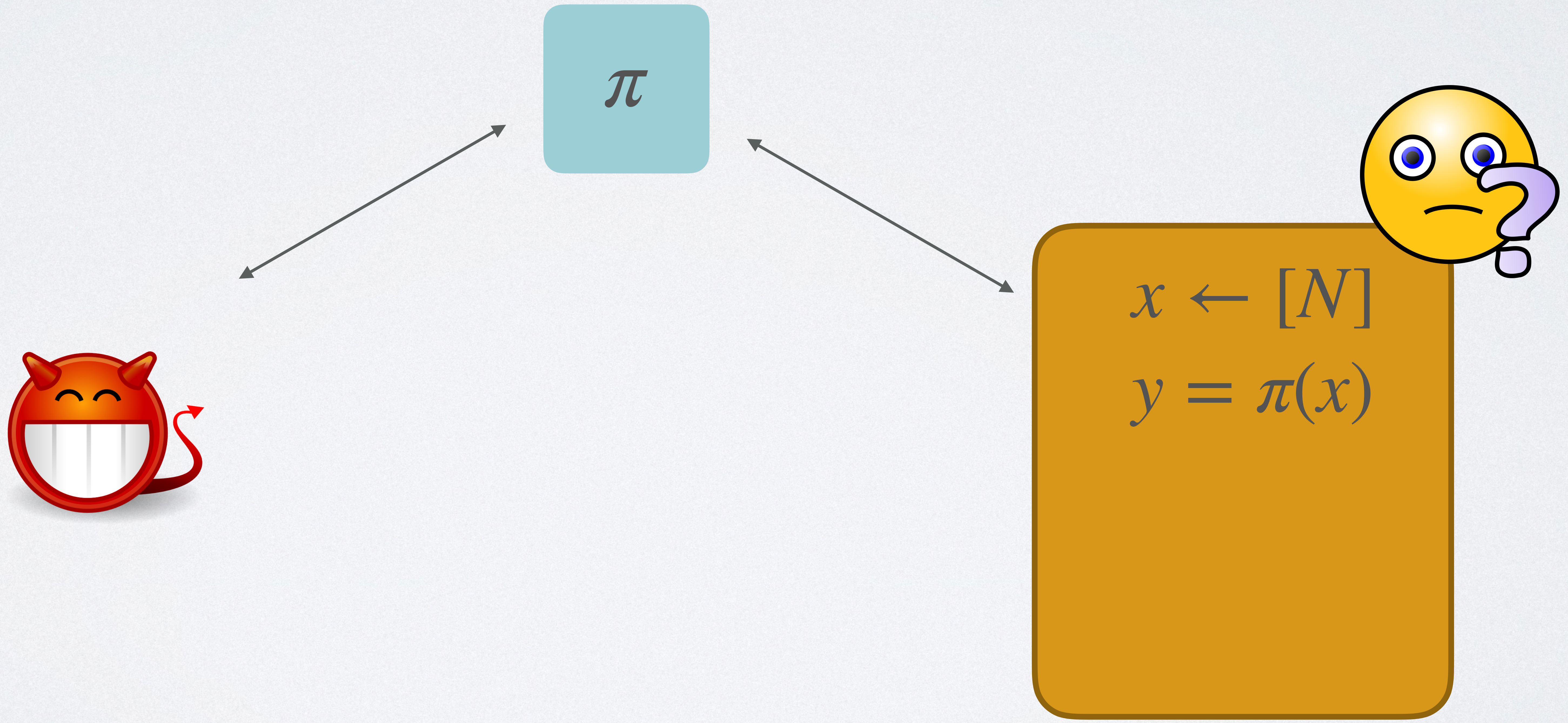
Toy Example: One-Way Permutations

Random permutation $\pi : [N] \to [N]$



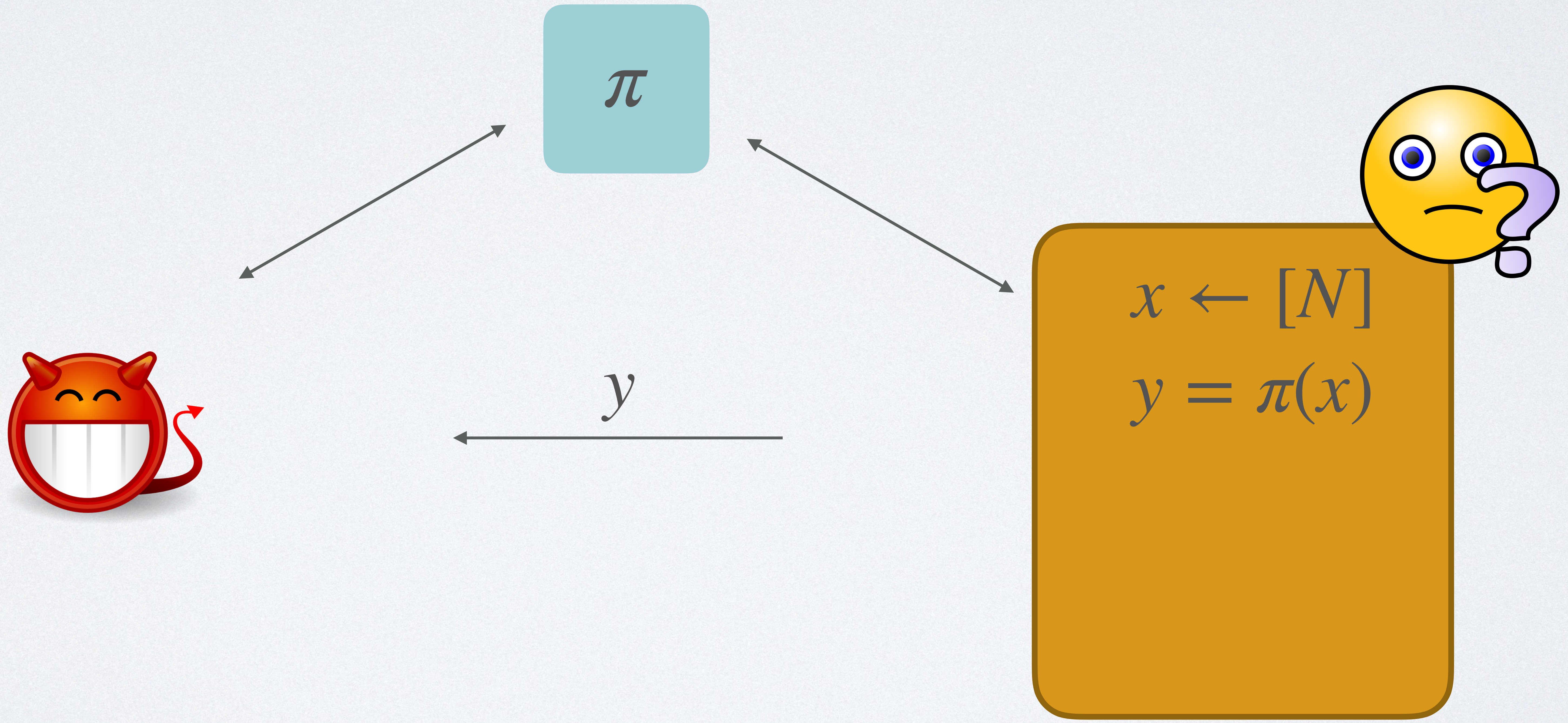
Toy Example:
One-Way Permutations

Random permutation $\pi : [N] \to [N]$



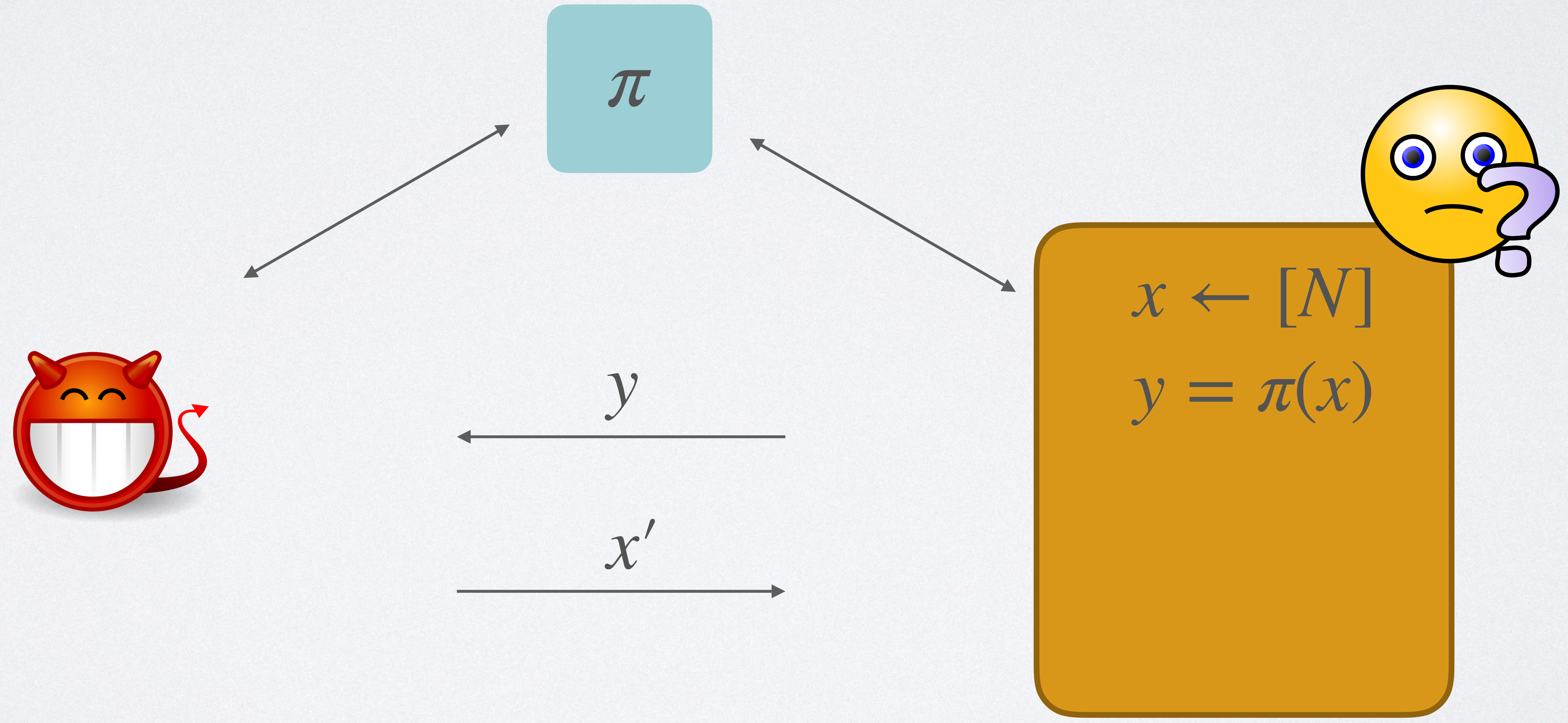
Toy Example:
One-Way Permutations

Random permutation $\pi : [N] \to [N]$



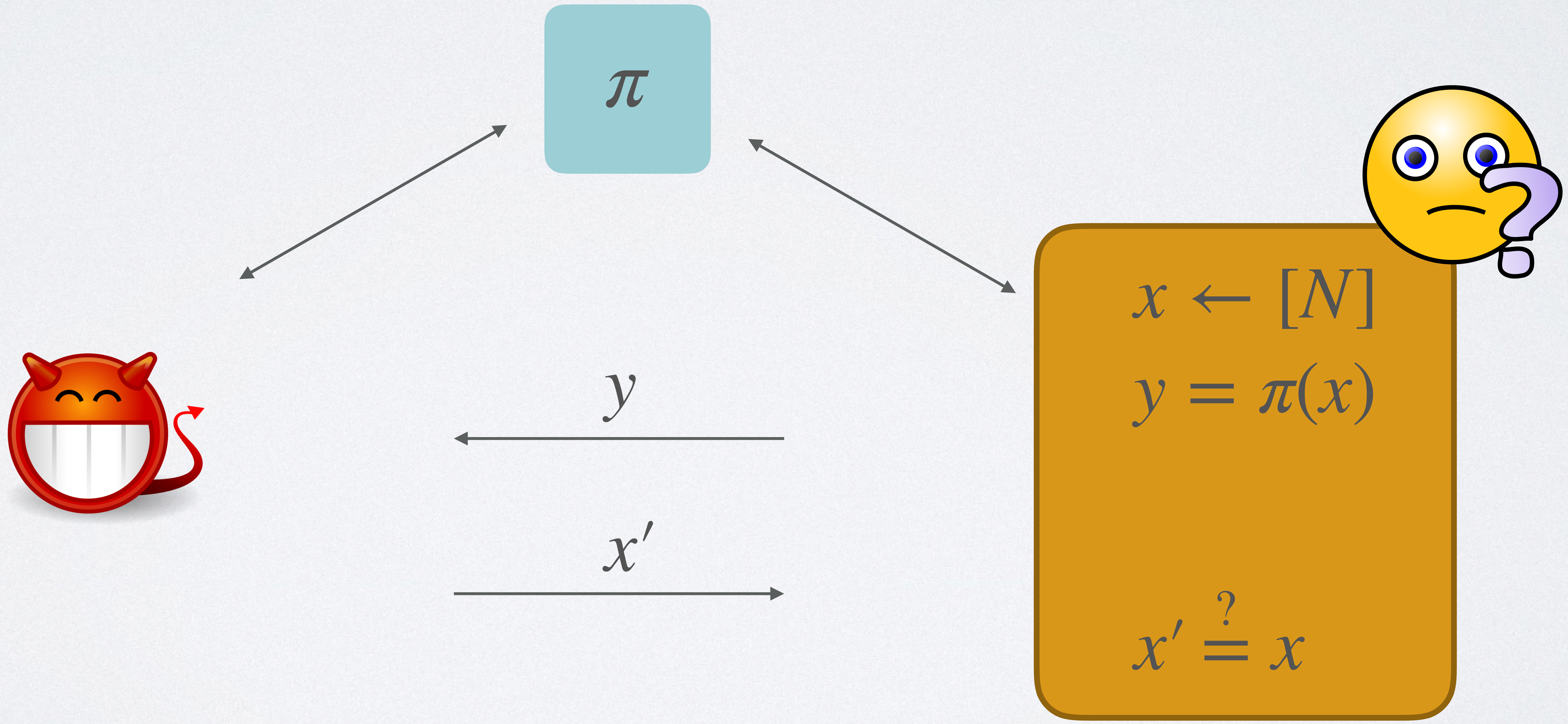
Toy Example: One-Way Permutations

Random permutation $\pi : [N] \to [N]$



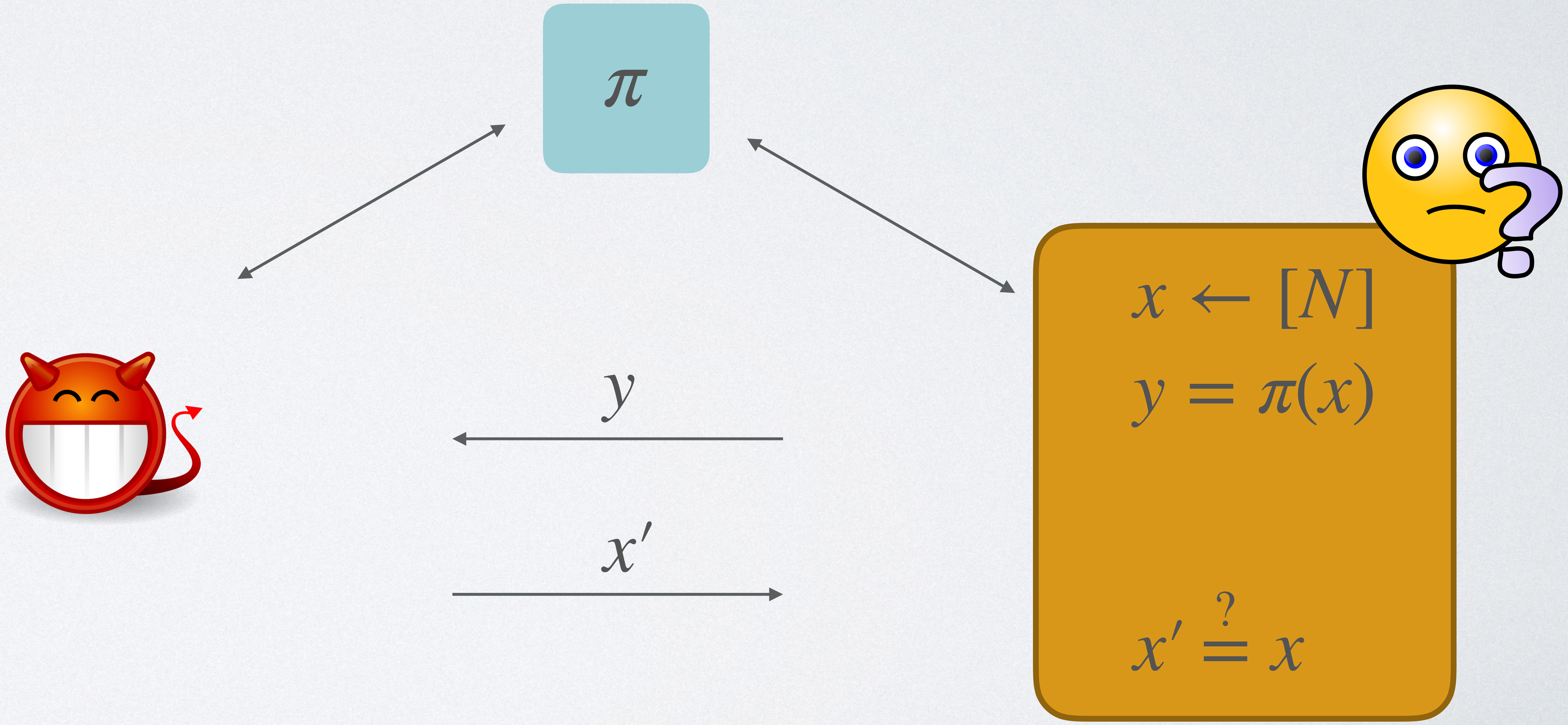
Toy Example: One-Way Permutations

Random permutation $\pi : [N] \to [N]$



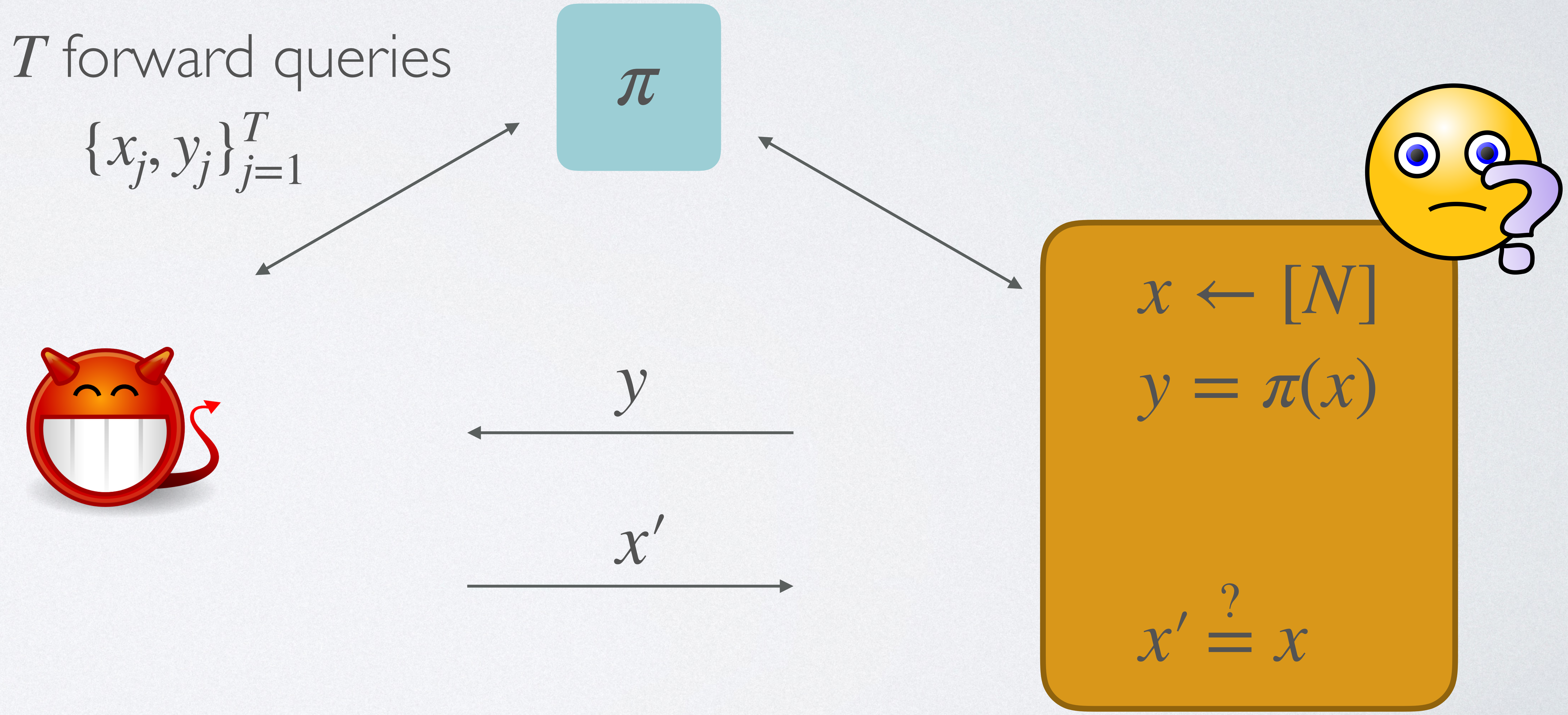
Toy Example: One-Way Permutations

Random permutation $\pi : [N] \to [N]$



Toy Example: One-Way Permutations

Random permutation $\pi : [N] \to [N]$

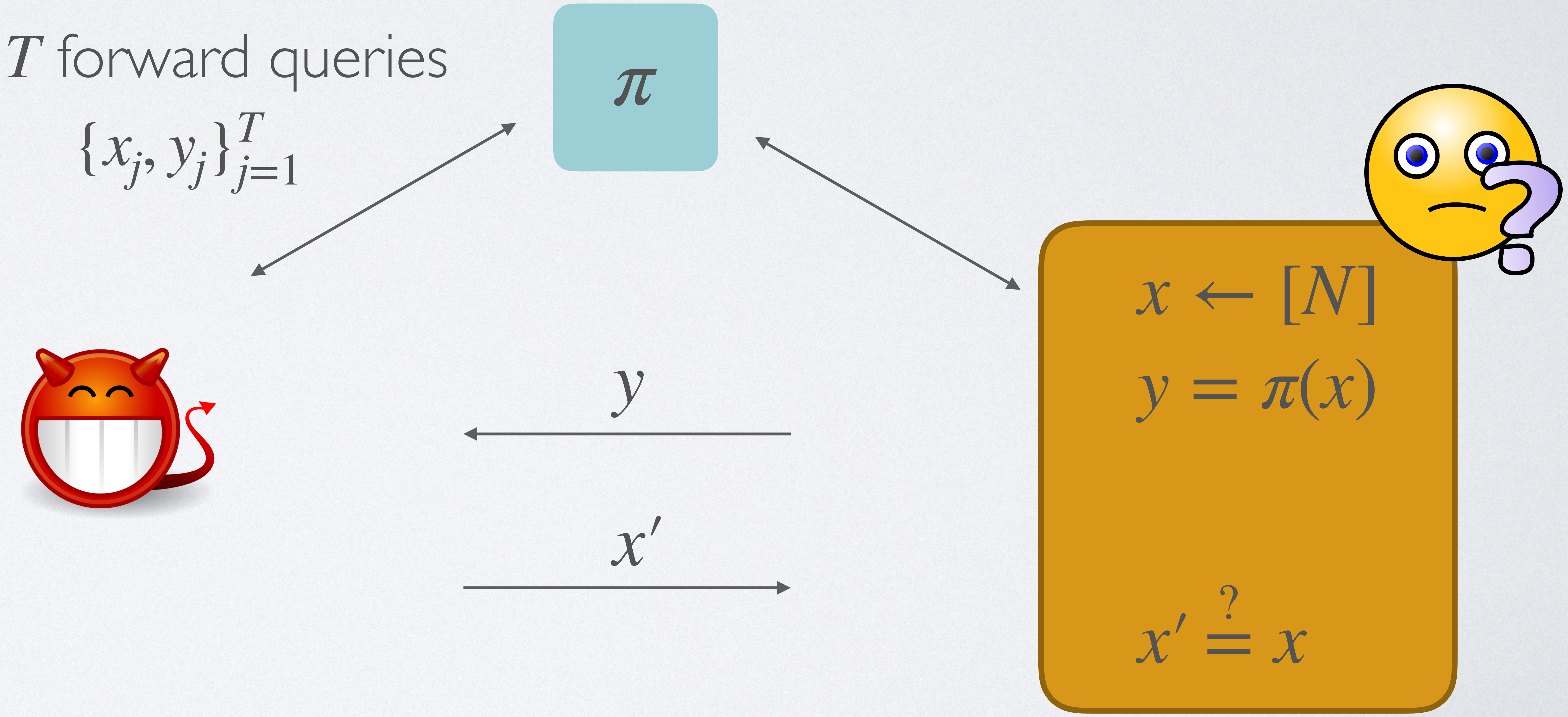


Toy Example:
One-Way Permutations

Random permutation $\pi : [N] \to [N]$

Event **BAD**:

$\exists j : \quad x_j = x$



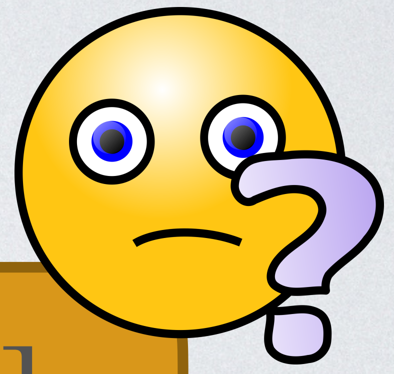
Toy Example: One-Way Permutations

Random permutation $\pi : [N] \to [N]$

T forward queries

$\{x_j, y_j\}_{j=1}^{T}$

π



$x \leftarrow [N]$

$y = \pi(x)$

y

x'

$x' \stackrel{?}{=} x$

Event **BAD**:

$\exists j : \quad x_j = x$

$$\text{P[BAD]} \leq \frac{T}{N}$$

Toy Example: One-Way Permutations

Random permutation $\pi : [N] \to [N]$

Conclusion:

One-Way Permutations secure up to ***N* queries**

Event BAD:

$\exists j : \quad x_j = x$

$\text{P[BAD]} \leq \frac{T}{N}$

x'

$x \leftarrow [N]$

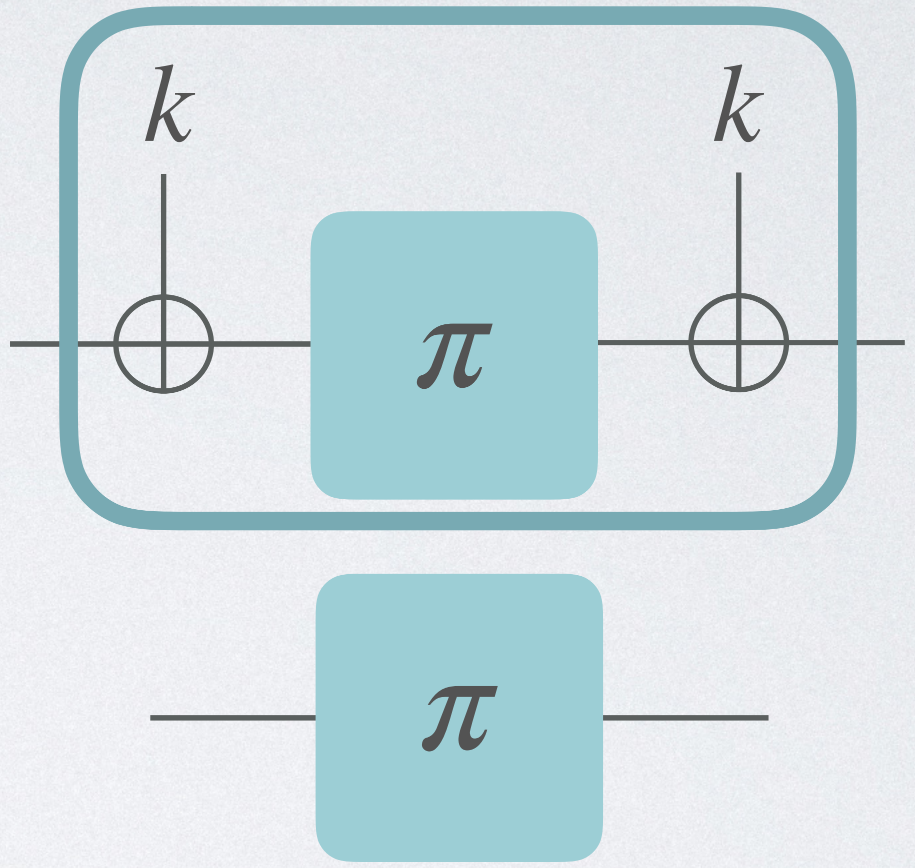
$y = \pi(x)$

$x' \stackrel{?}{=} x$

EM Cipher with Random Permutation

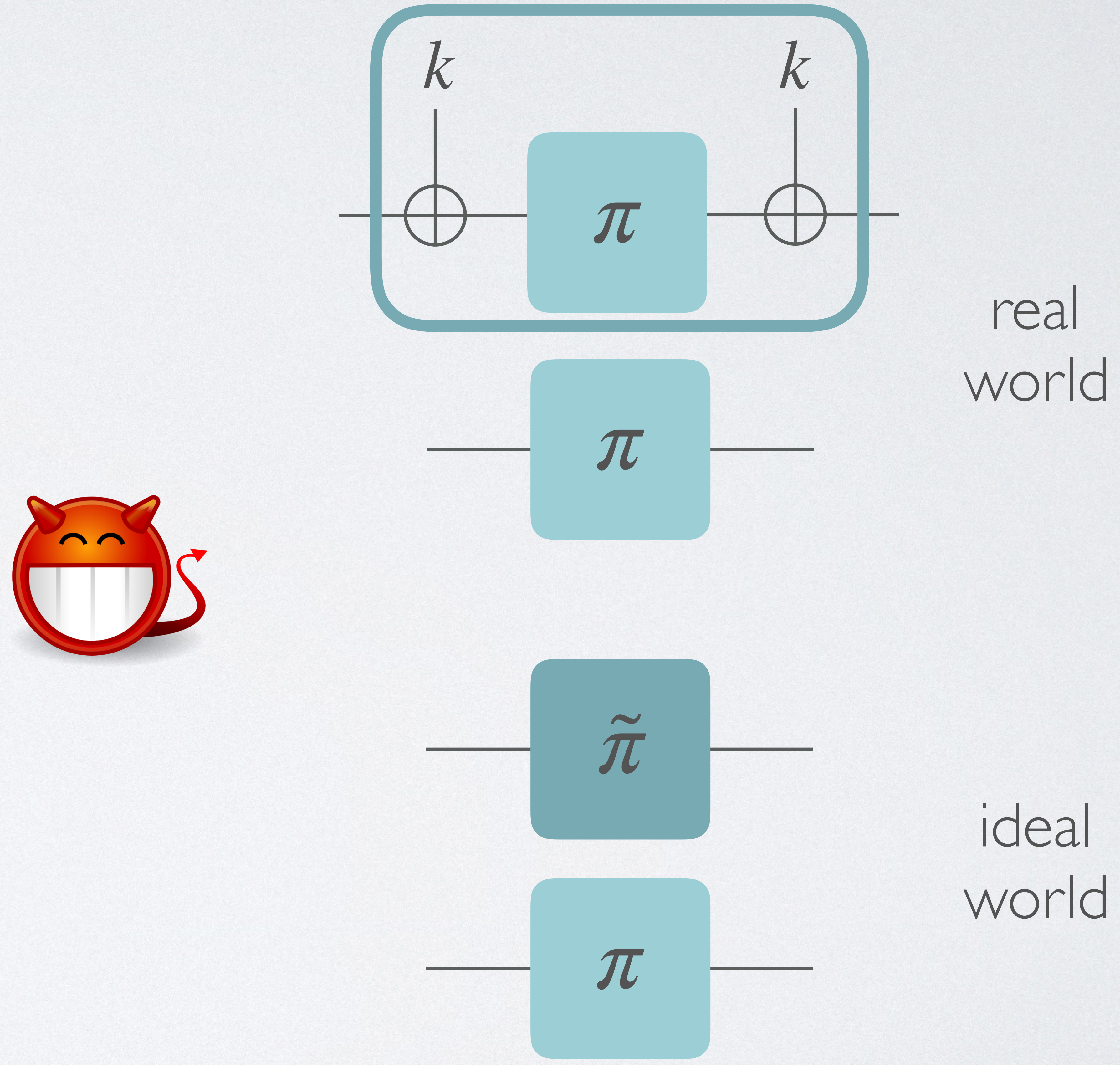
EM Cipher with Random Permutation

EM Cipher with Random Permutation



real world

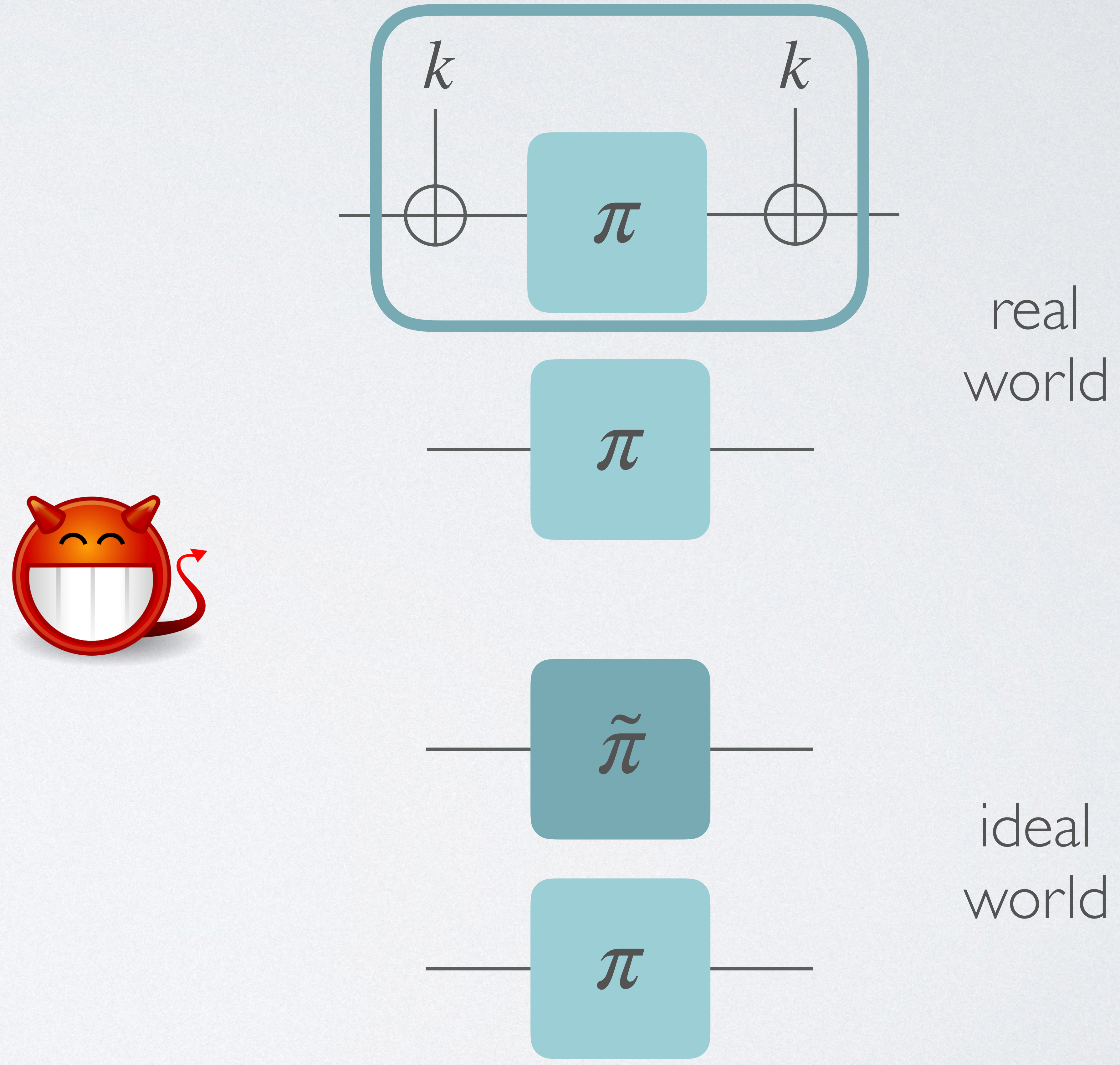
EM Cipher with Random Permutation



EM Cipher with Random Permutation

q construction queries

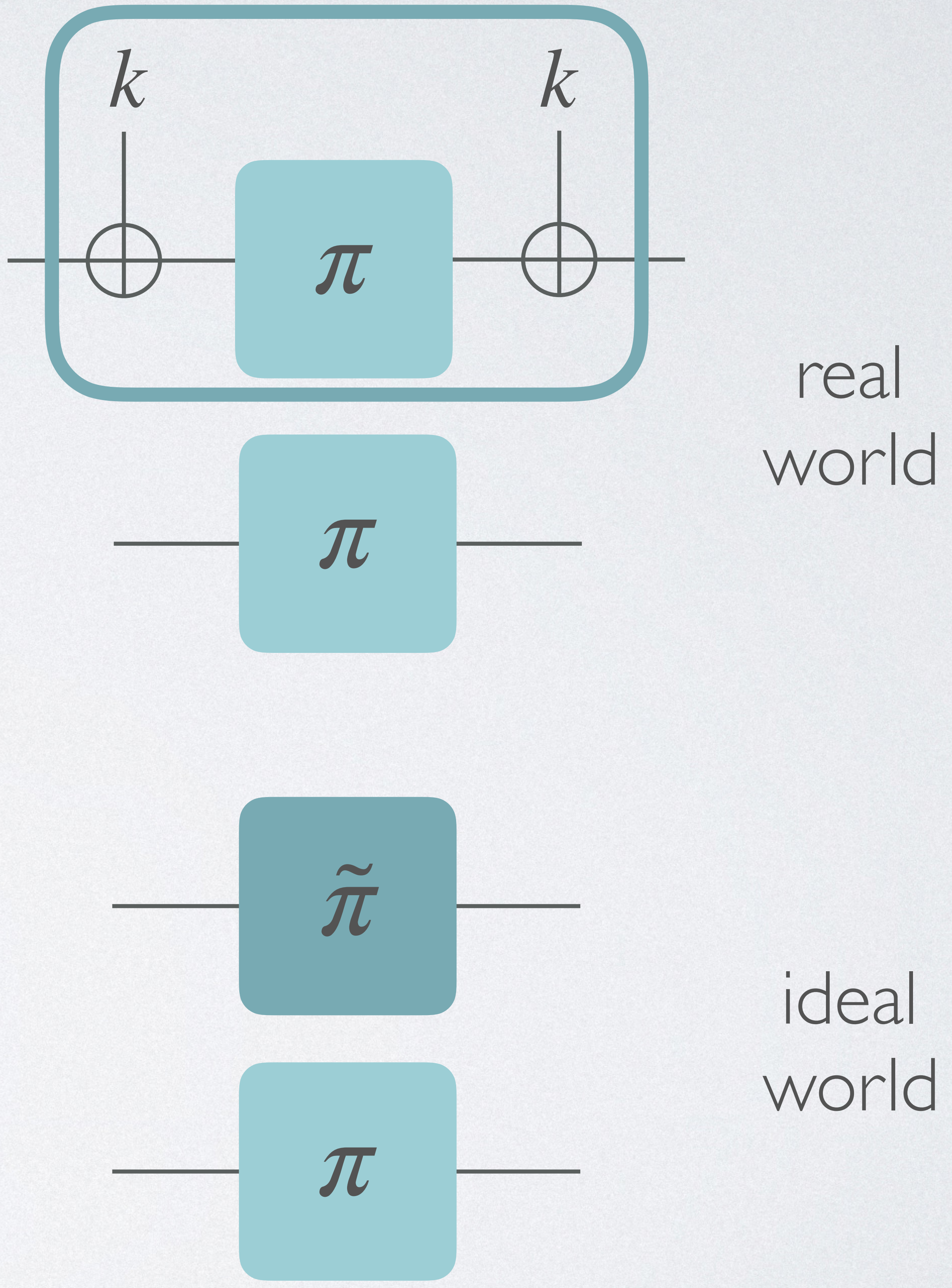
$\{u_i, v_i\}_{i=1}^{q}$



EM Cipher with Random Permutation

q construction queries
$\{u_i, v_i\}_{i=1}^{q}$

T primitive queries
$\{x_j, y_j\}_{j=1}^{T}$



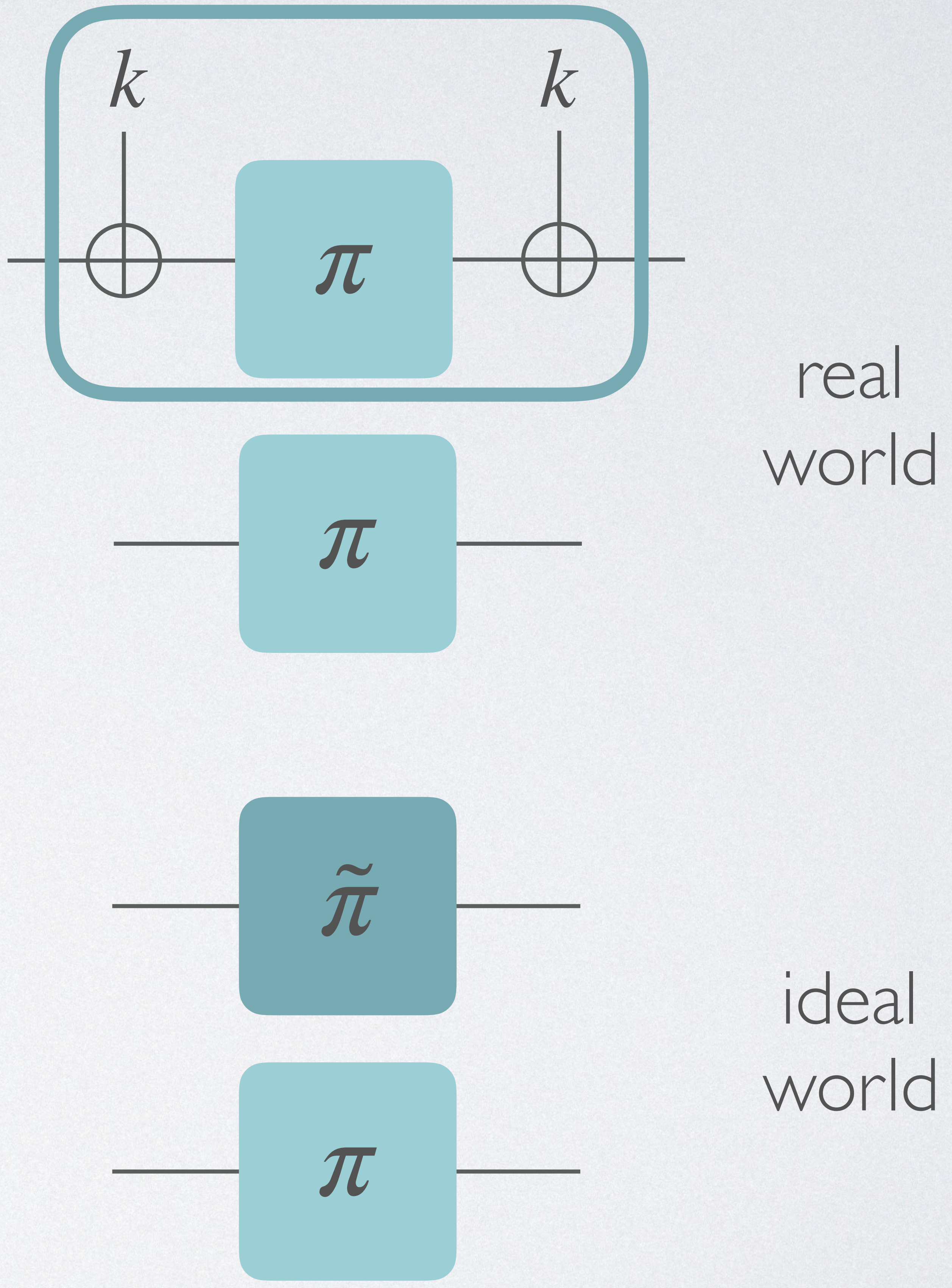
EM Cipher with Random Permutation

q construction queries
$\{u_i, v_i\}_{i=1}^{q}$

T primitive queries
$\{x_j, y_j\}_{j=1}^{T}$

Event BAD:

$\exists i, j : \; u_i \oplus k = x_j \; \vee \; v_i \oplus k = y_j$



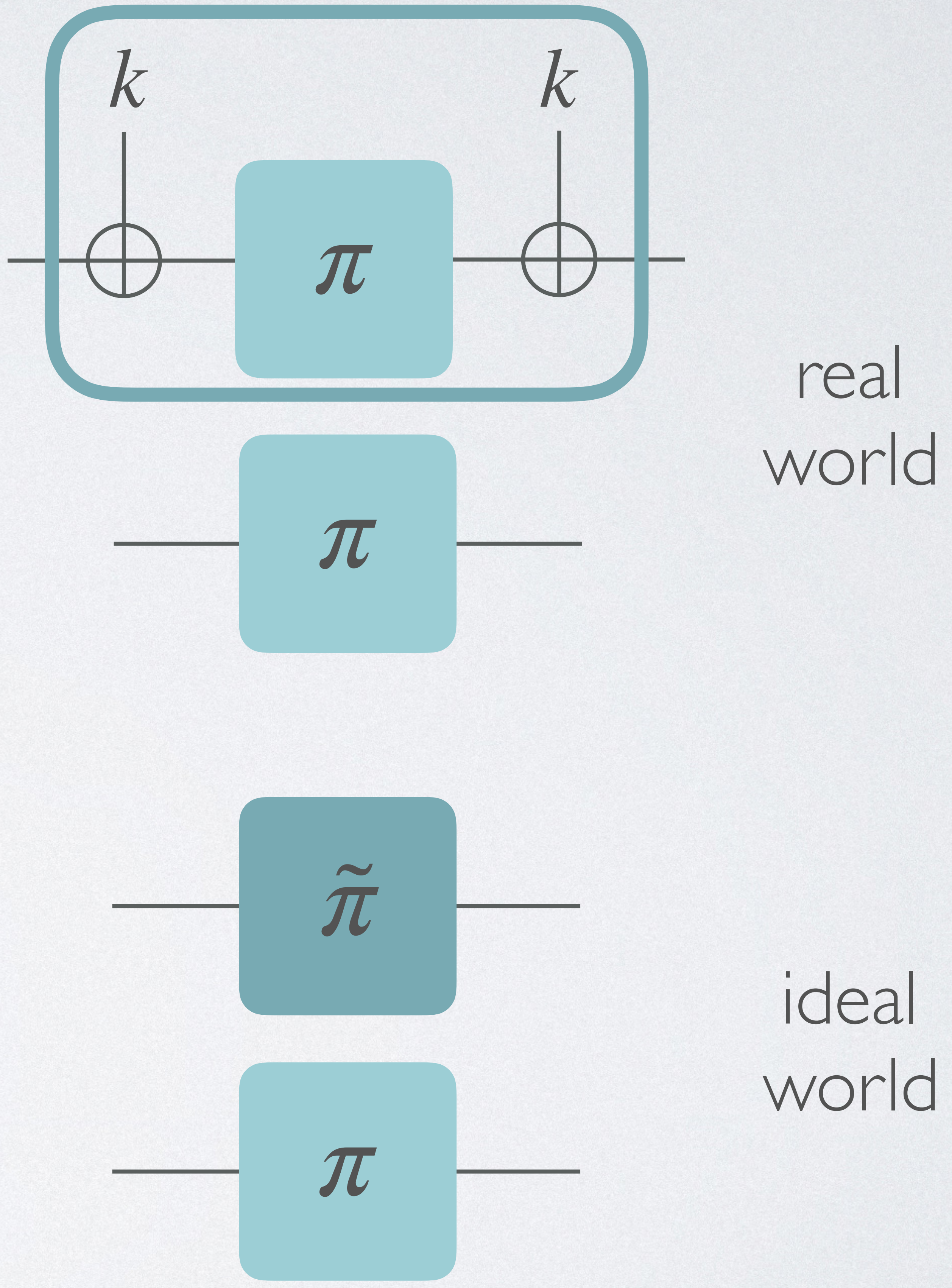
EM Cipher with Random Permutation

q construction queries

$\{u_i, v_i\}_{i=1}^{q}$

T primitive queries

$\{x_j, y_j\}_{j=1}^{T}$



Event BAD:

$$\exists i, j : \; u_i \oplus k = x_j \;\vee\; v_i \oplus k = y_j$$

$$\mathrm{P[BAD]} \leq \frac{qT}{N}$$

EM Cipher with Random Permutation

q construction que

$\{u_i, v_i\}_{i=1}^{q}$

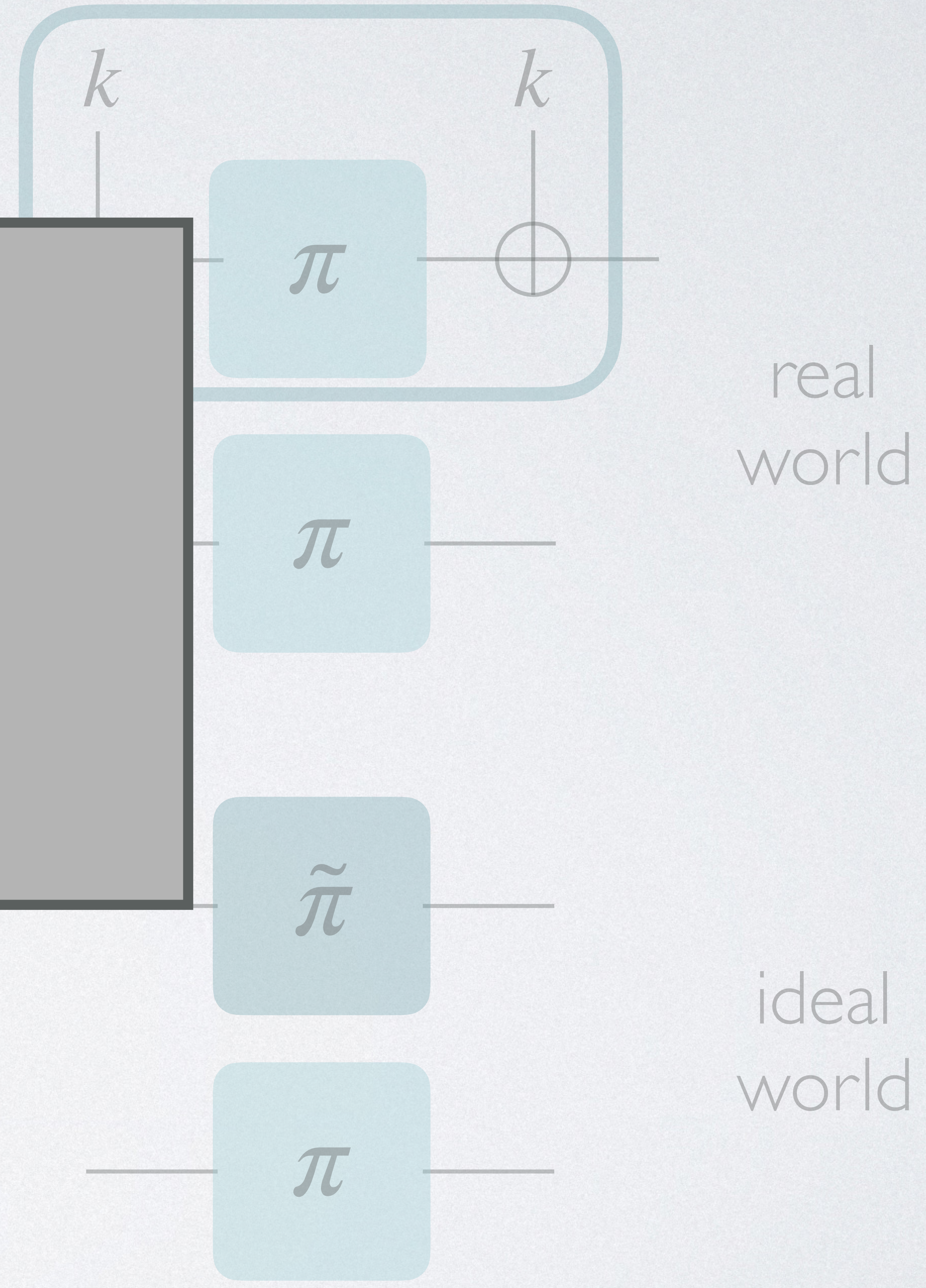
Conclusion:

Even-Mansour secure up to **birthday bound**

Event BAD:

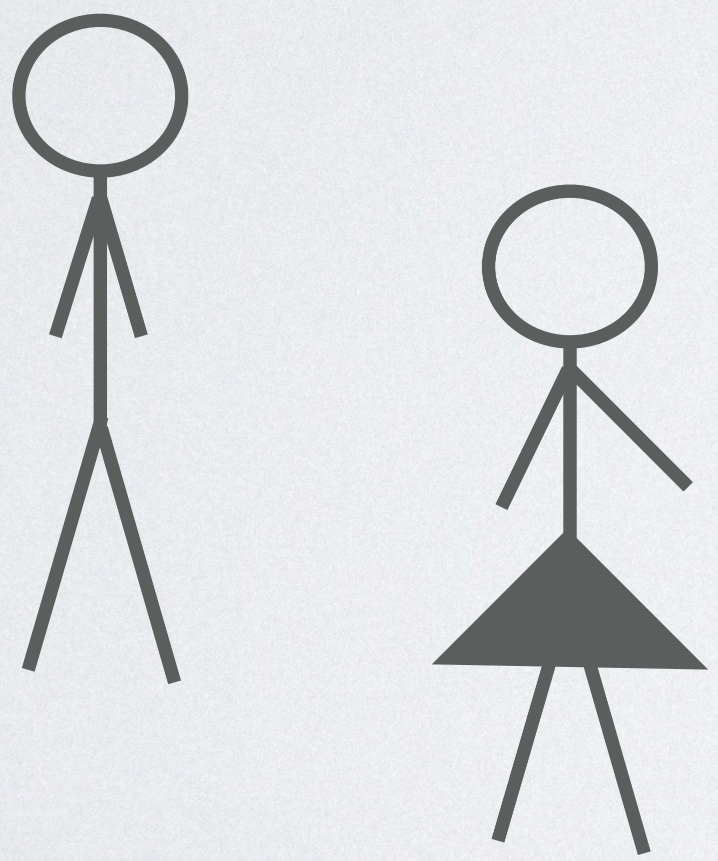
$\exists i, j: \ u_i \oplus k = x_j \vee v_i \oplus k = y_j$

P[BAD] $\leq \frac{qT}{2^n}$



Discrete Logarithms

Rule out generic algorithms via analysis in the
Generic Group Model

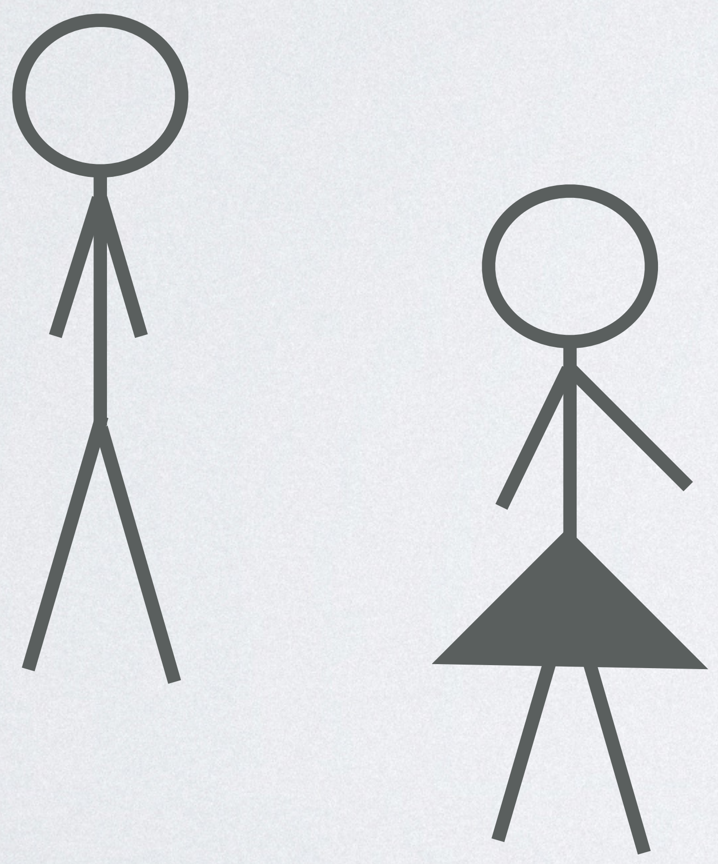


Discrete Logarithms

Rule out generic algorithms via analysis in the
Generic Group Model

G represented by random injection

$$\sigma : [N] \to [M]$$

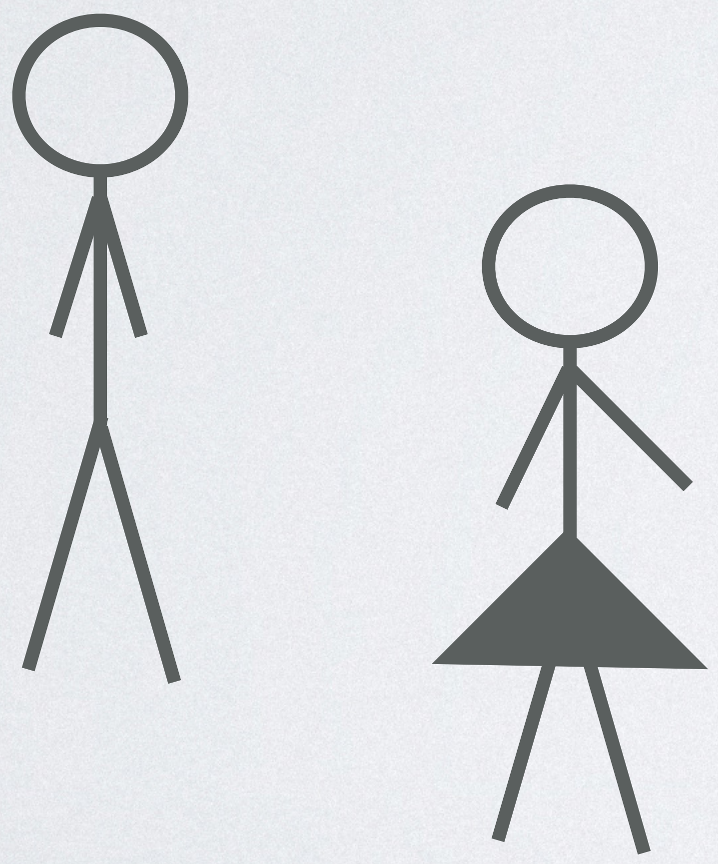


Discrete Logarithms

Rule out generic algorithms via analysis in the
Generic Group Model

G represented by random injection

$$\sigma : [N] \to [M]$$

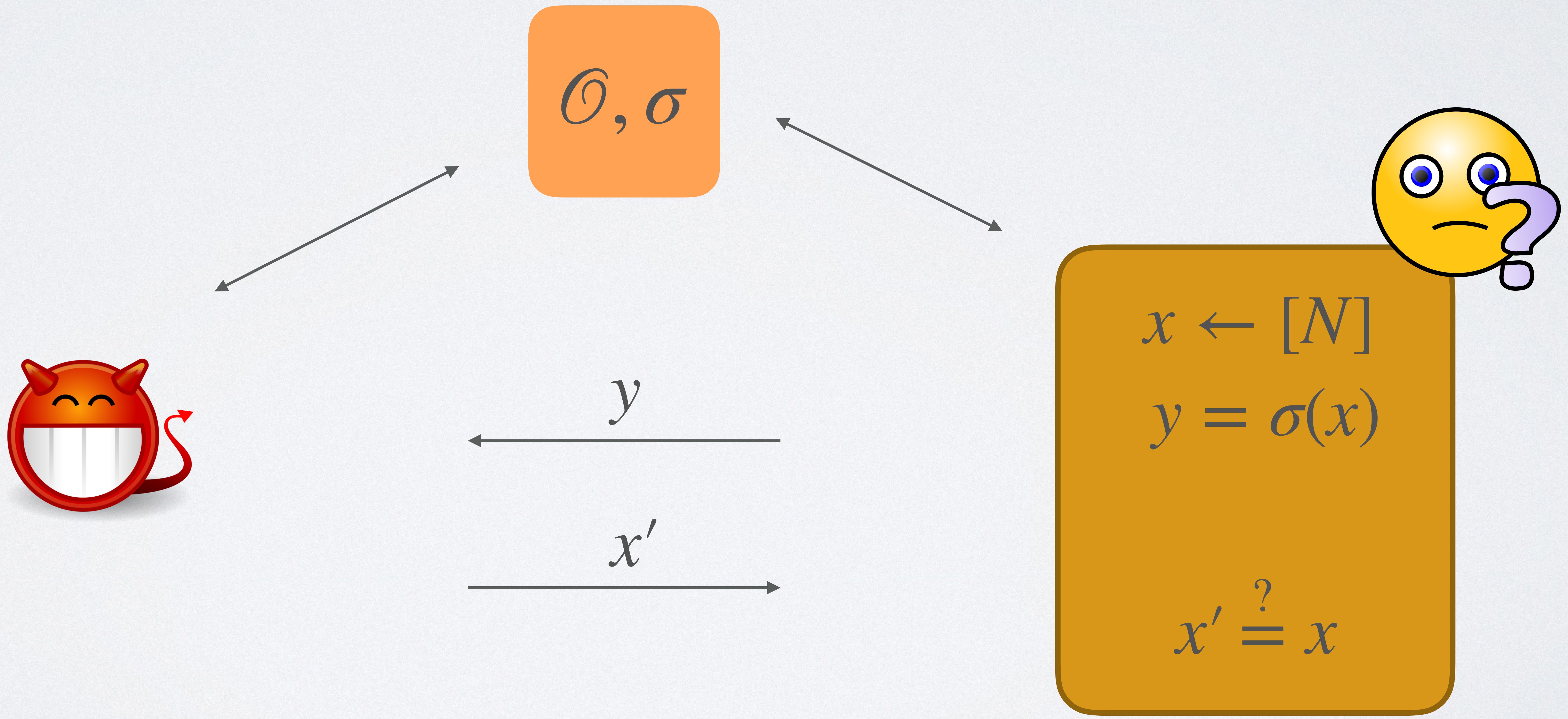


Group operation oracle:

$$\mathcal{O} : \ (\sigma(s), \sigma(s')) \ \mapsto \ \sigma(s + s')$$

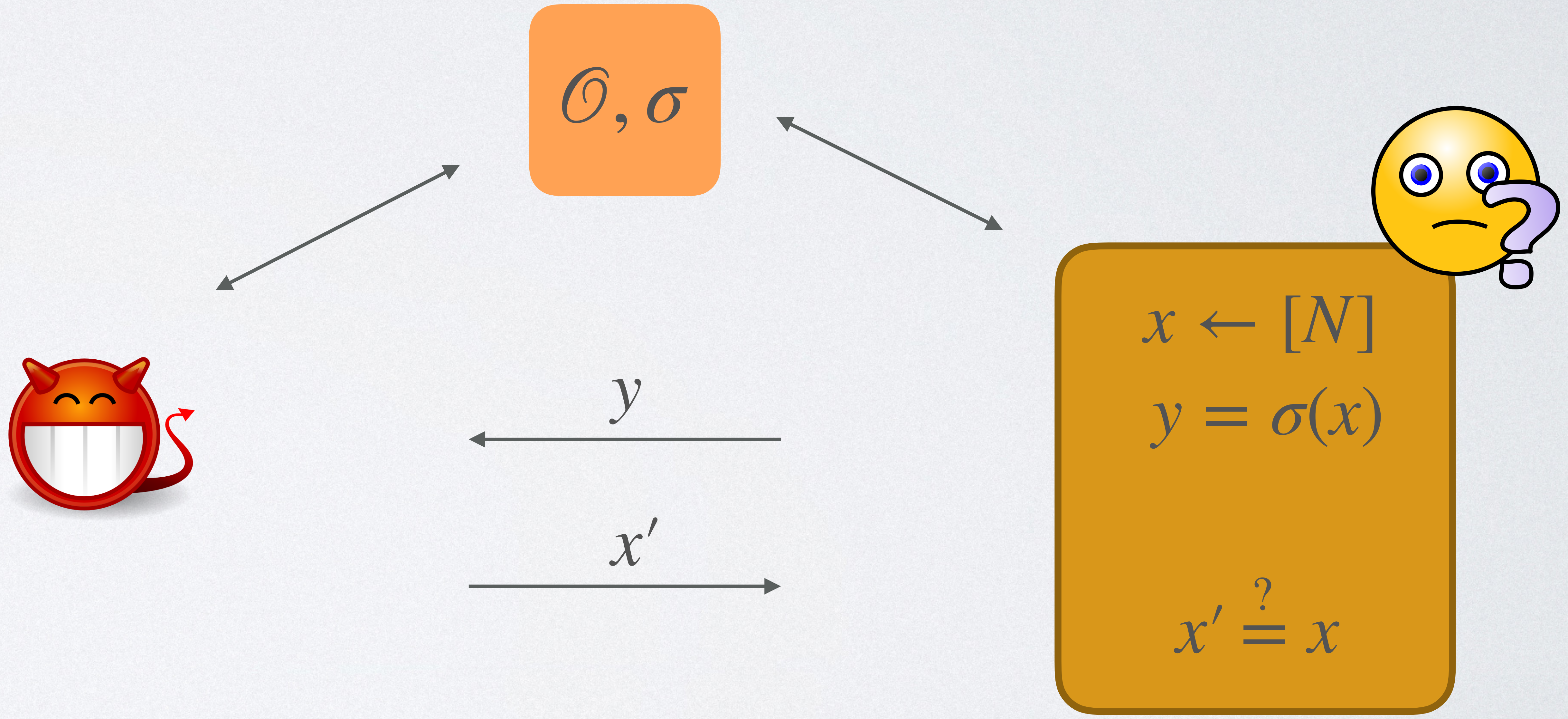
Discrete Logarithms in the GGM

Random injection $\sigma : [N] \to [M]$



Discrete Logarithms in the GGM

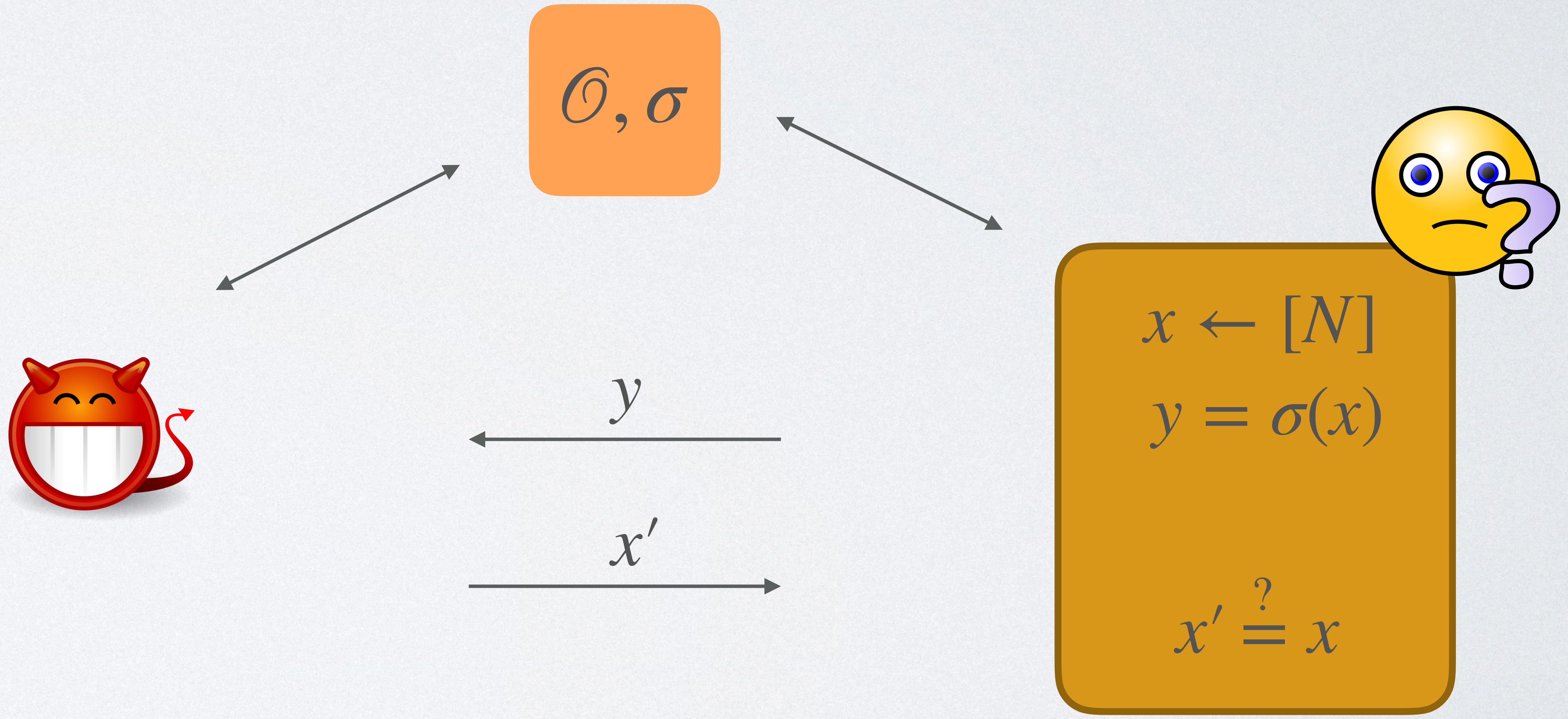
Random injection $\sigma : [N] \to [M]$



Discrete Logarithms in the GGM

Random injection $\sigma : [N] \to [M]$

Shoup '97



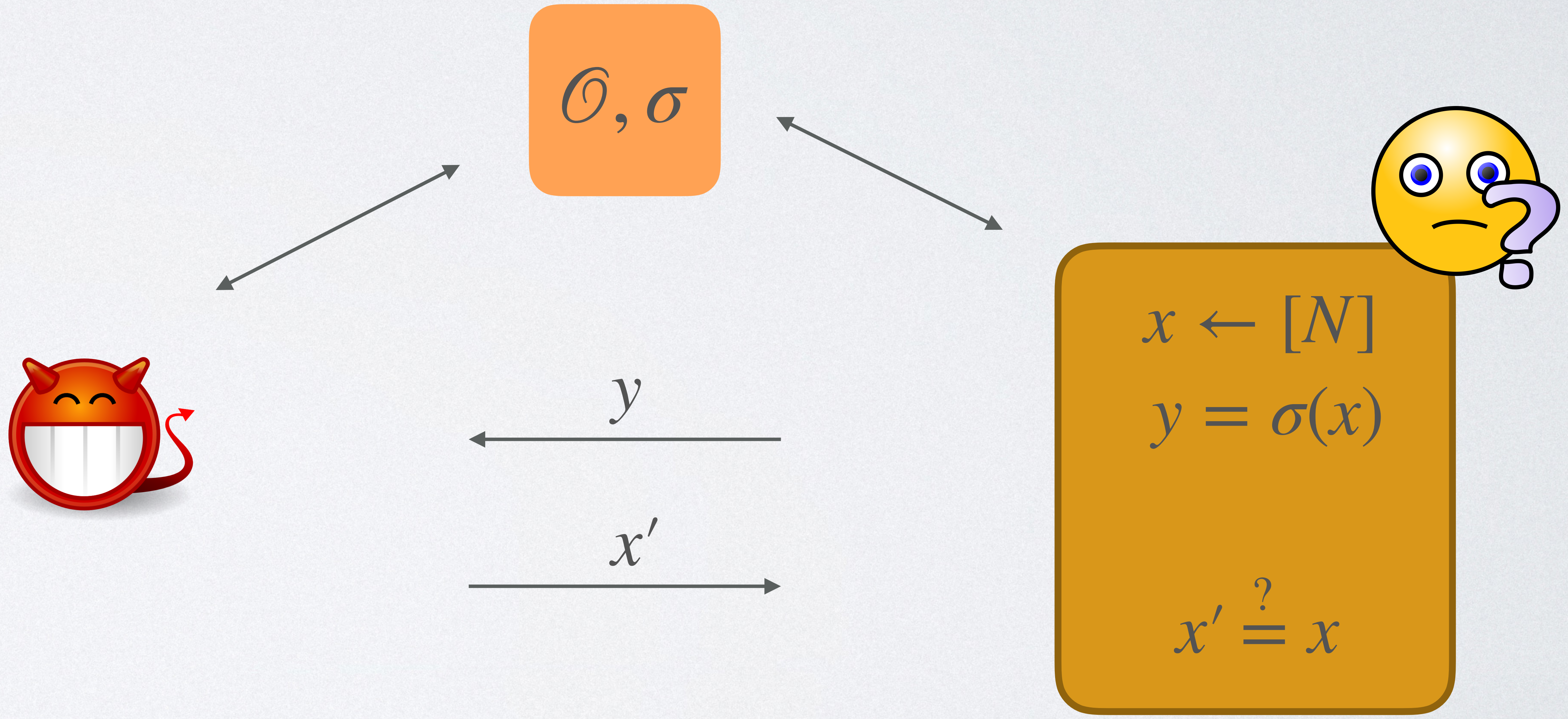
Discrete Logarithms in the GGM

Random injection $\sigma : [N] \to [M]$

By making queries to $\mathcal{O}$:

Shoup '97

$\mathscr{A}$ "generates" degree-1 polynomials in X



Discrete Logarithms in the GGM

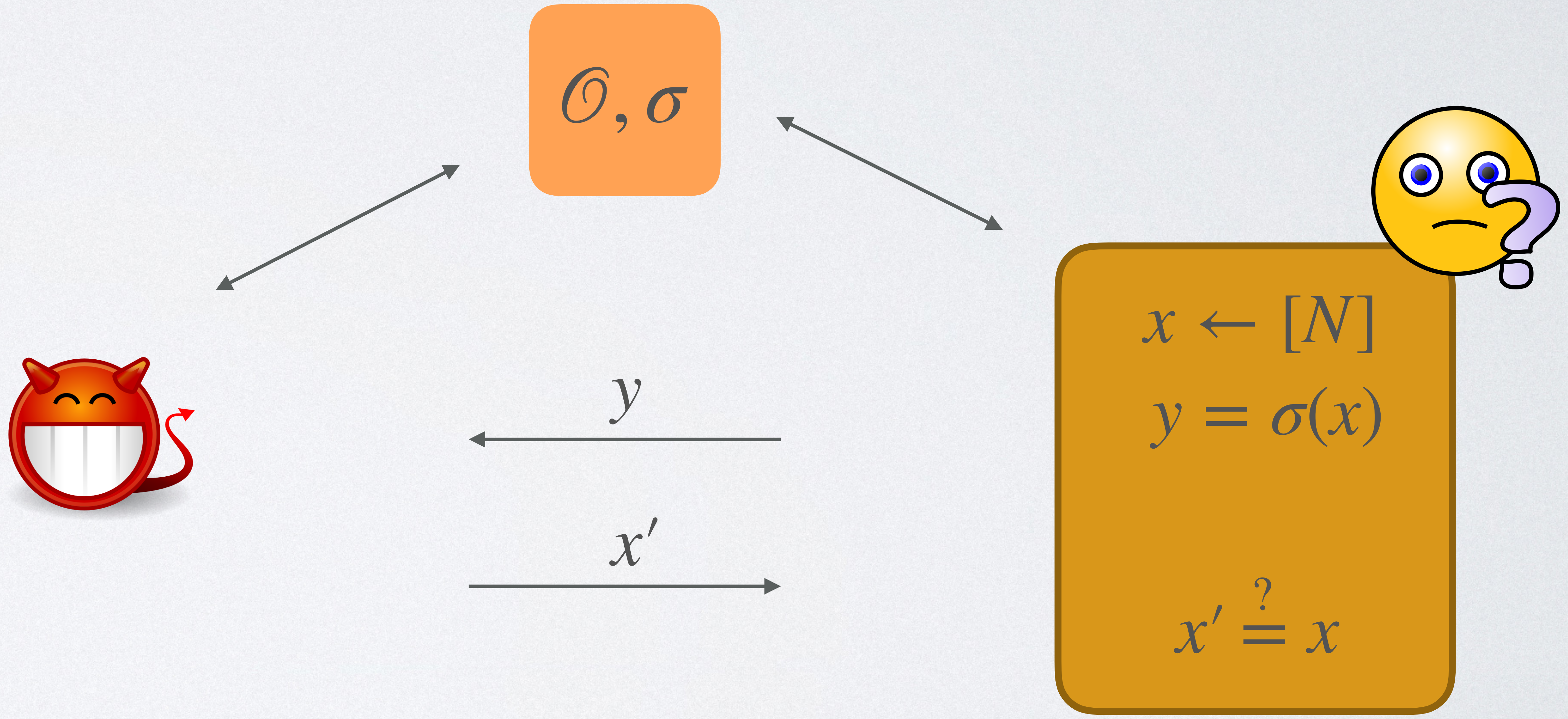
Random injection $\sigma : [N] \rightarrow [M]$

By making queries to $\mathcal{O}$:

Shoup '97

$\mathscr{A}$ "generates" degree-1 polynomials in X

Event **BAD**: two polynomials collide at $X = x$



Discrete Logarithms in the GGM

Random injection $\sigma : [N] \to [M]$

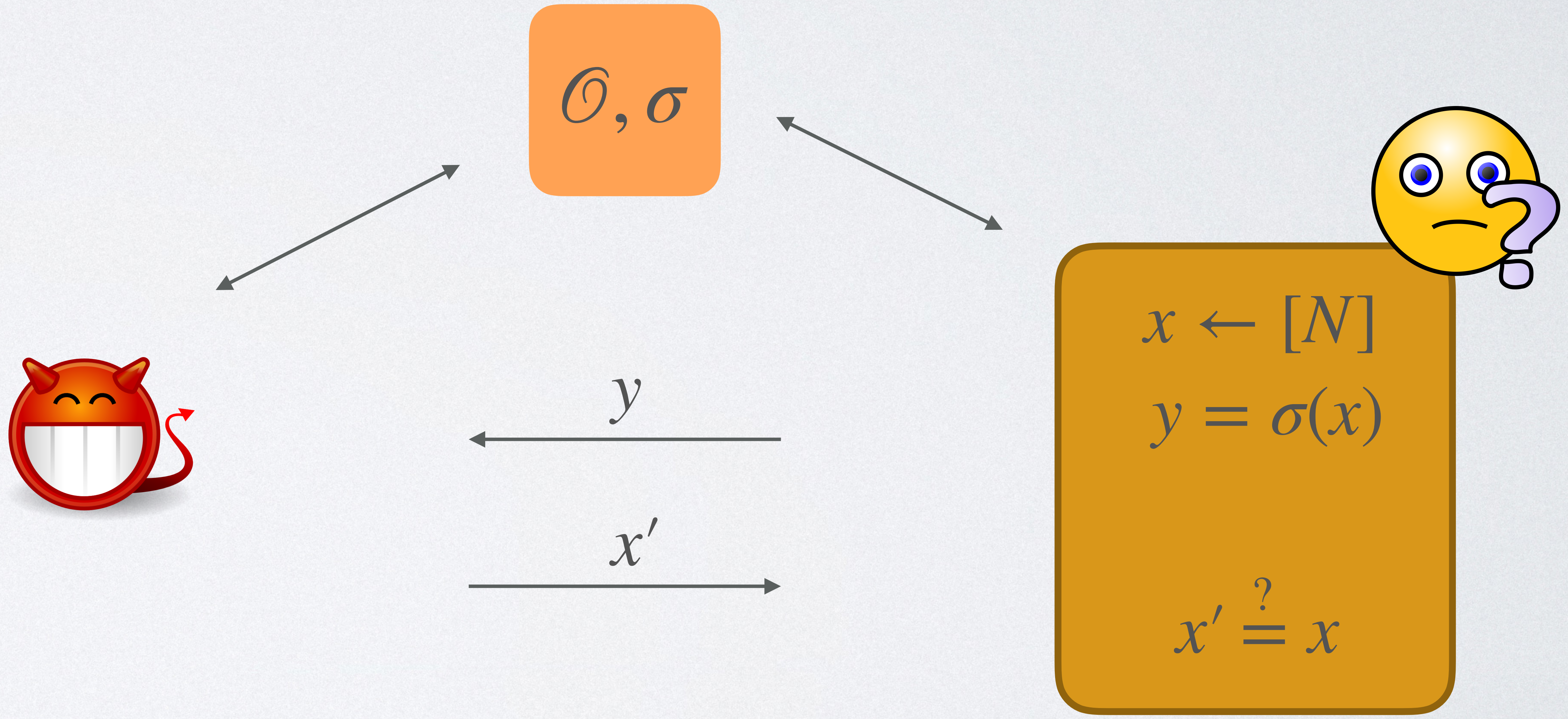
By making queries to $\mathcal{O}$:

Shoup '97

$\mathscr{A}$ "generates" degree-1 polynomials in X

Event **BAD**: two polynomials collide at $X = x$

$$\text{P[BAD]} \leq \frac{T^2}{N}$$



Discrete Logarithms in the GGM

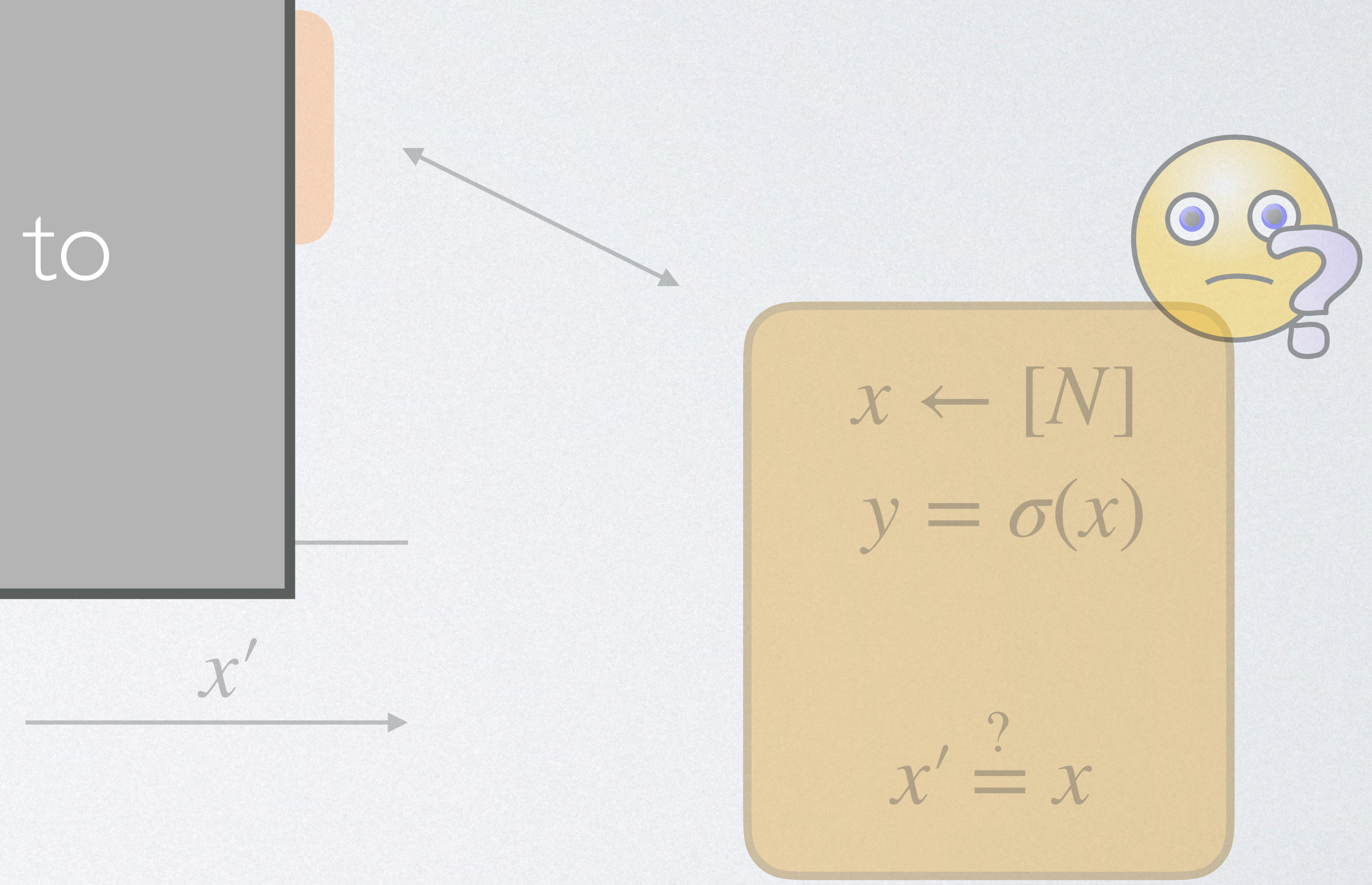
Random injection $\sigma : [N] \to [M]$

By making queries to $\mathcal{O}$:

$\mathscr{A}$ "generates" degree-1

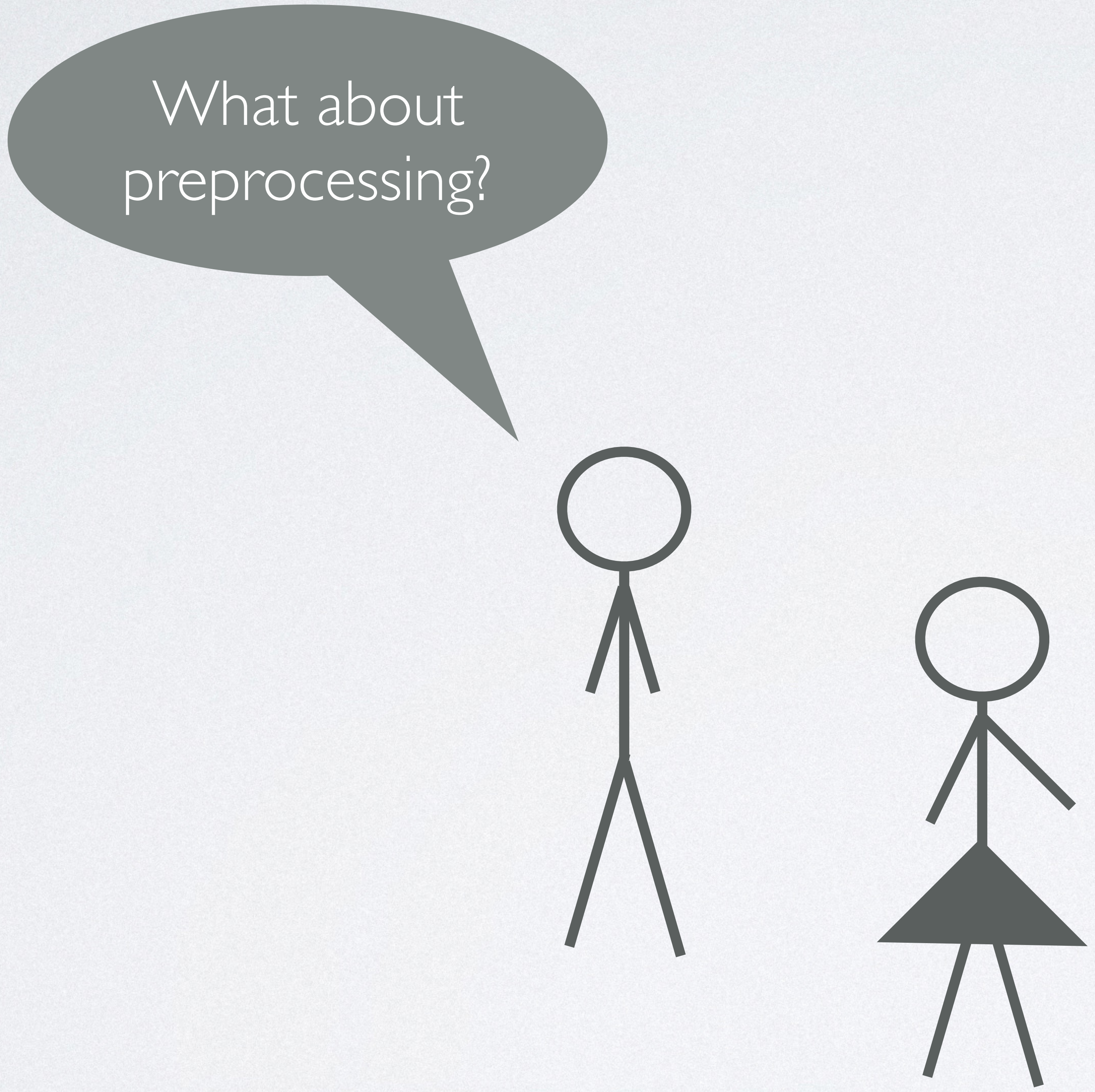
Event **BAD**: two polynom

P[BAD] $\leq \frac{T^2}{N}$



Conclusion:

Discrete logarithm secure up to
birthday bound
in GGM.


What about preprocessing?

Preprocessing Attacks

Preprocessing Attacks

In practice:

- security parameter fixed
- dedicated attacker may perform **precomputation** to speed up online attack

Preprocessing Attacks

In practice:

- security parameter fixed
- dedicated attacker may perform **precomputation** to speed up online attack

Preprocessing Attacks

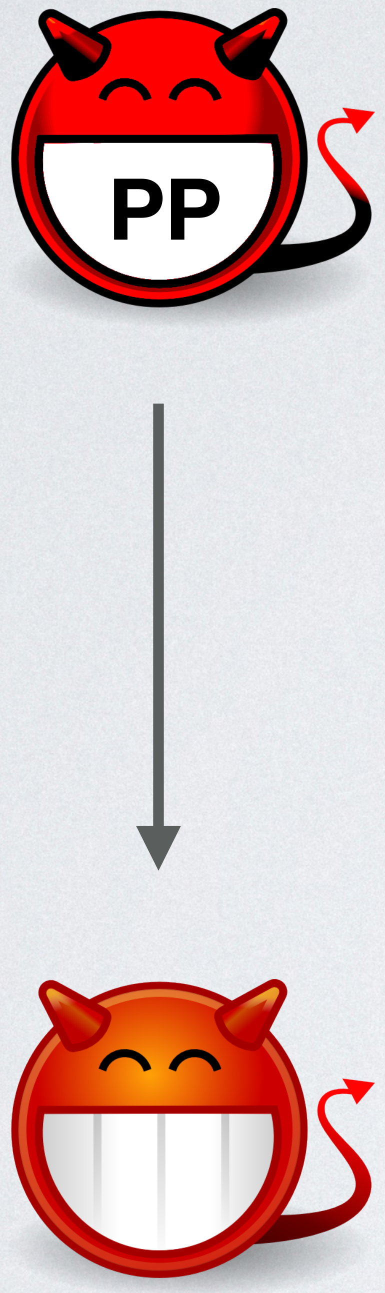
In practice:

- security parameter fixed
- dedicated attacker may perform **precomputation** to speed up online attack
- models non-uniformity

Attacking One-Way Permutations

Hellman '80

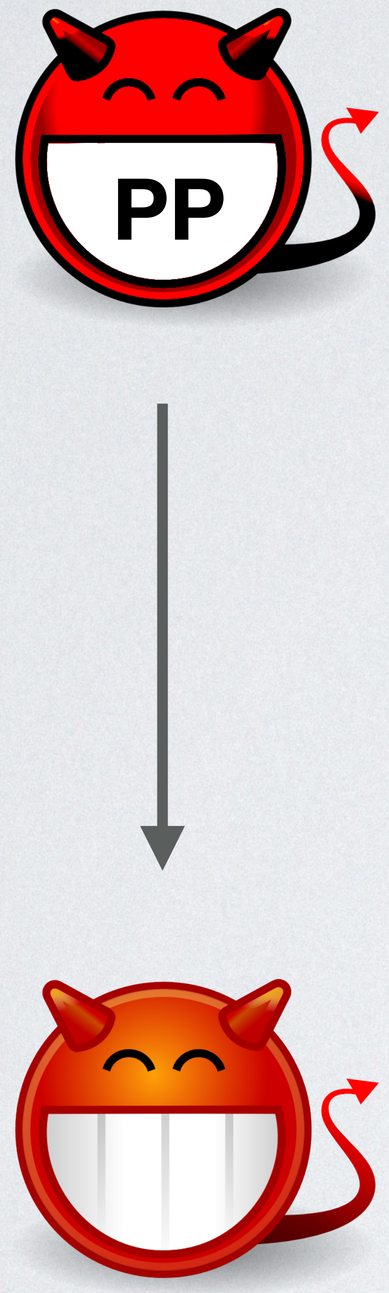
Permutation $\pi : [N] \to [N]$



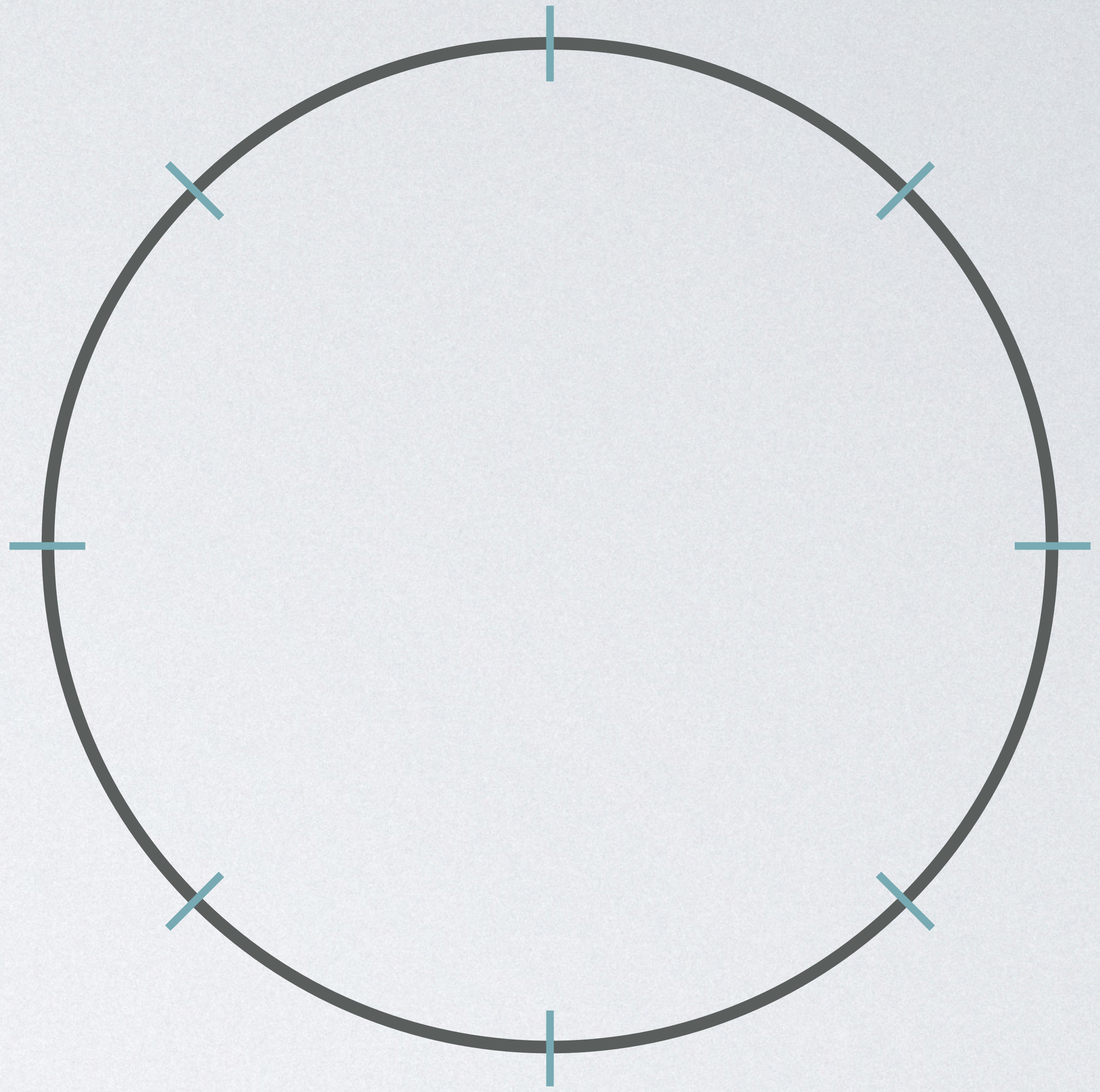
Attacking One-Way Permutations

Hellman '80

Permutation $\pi : [N] \to [N]$



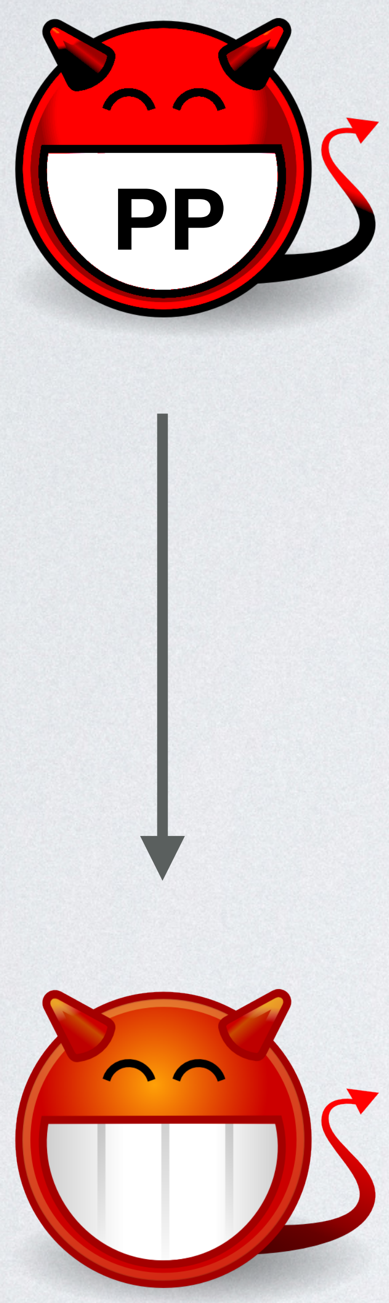
For every cycle of length at least S,
store points x_i at distance N/S



Attacking One-Way Permutations

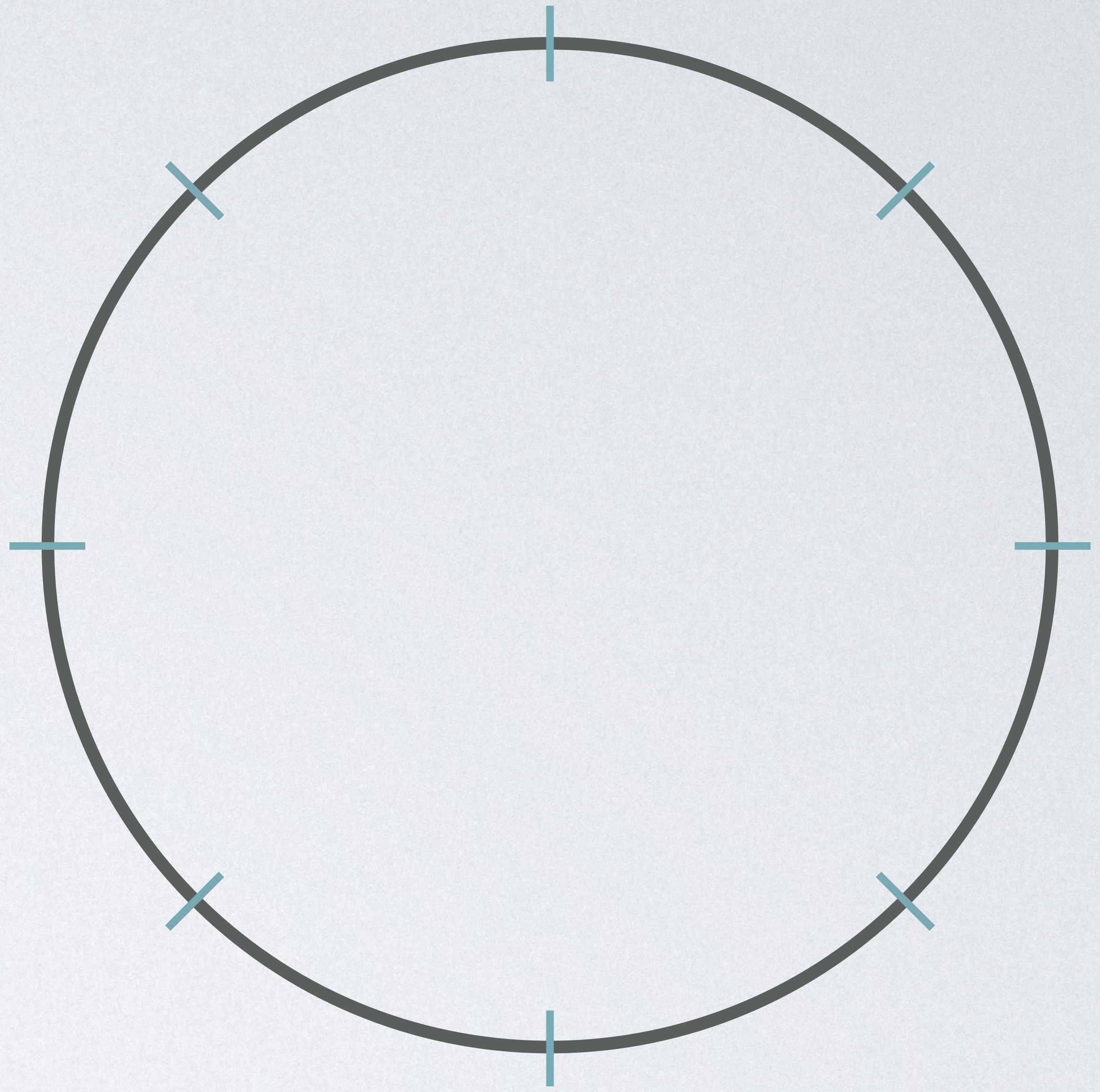
Hellman '80

Permutation $\pi : [N] \to [N]$



For every cycle of length at least S,
store points x_i at distance N/S

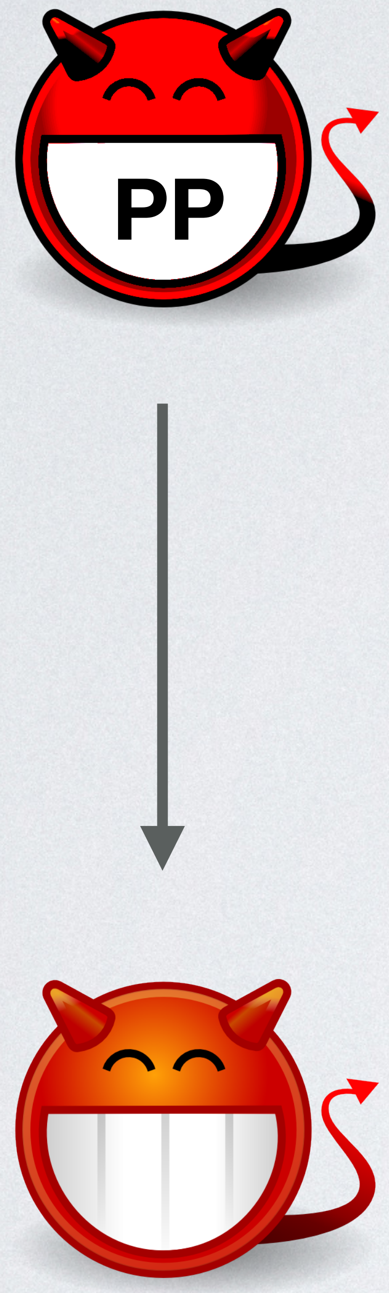
Advice: $z = (x_1, \ldots, x_S)$



Attacking One-Way Permutations

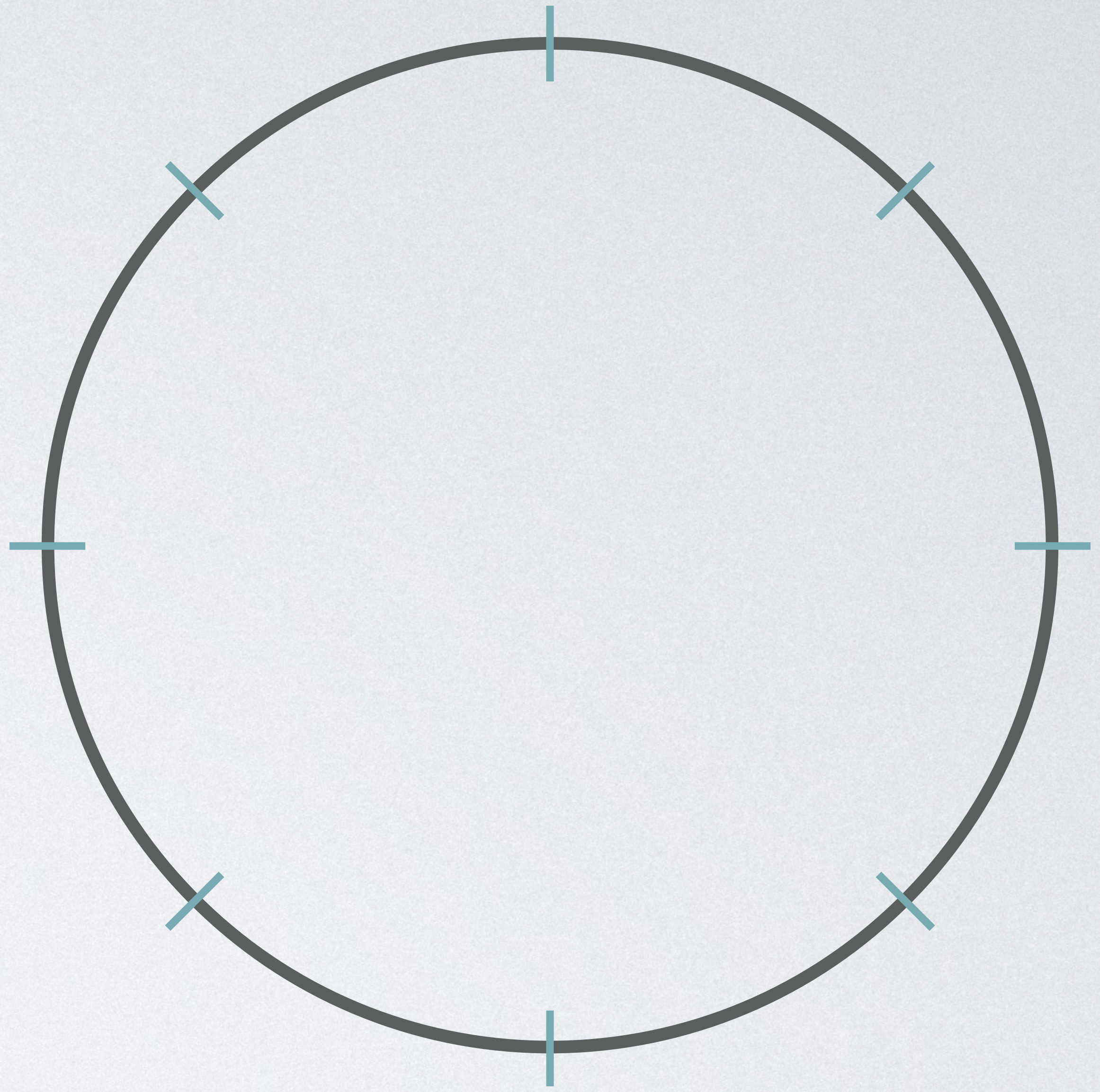
Hellman '80

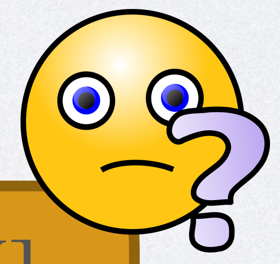
Permutation $\pi : [N] \to [N]$



For every cycle of length at least S,
store points x_i at distance N/S

Advice: $z = (x_1, \ldots, x_S)$





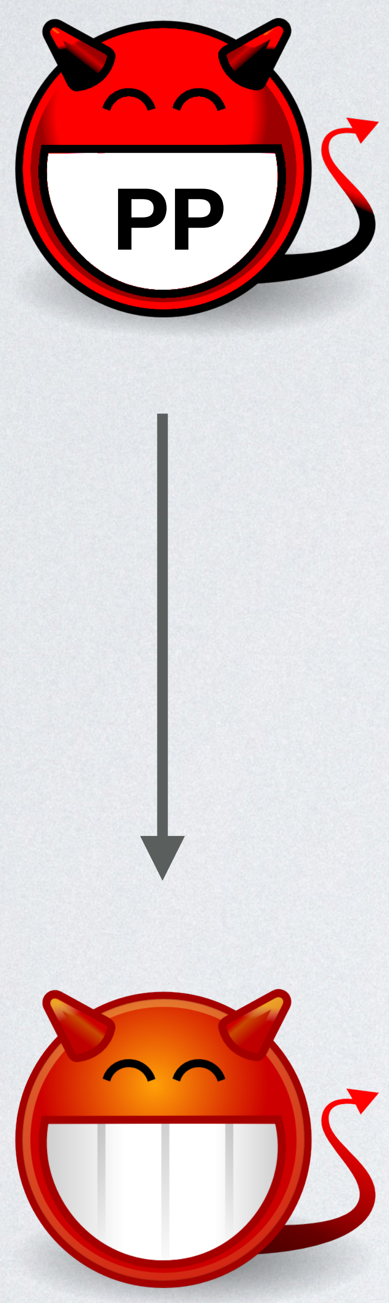
$x \leftarrow [N]$
$y = \pi(x)$

$x' \stackrel{?}{=} x$

Attacking One-Way Permutations

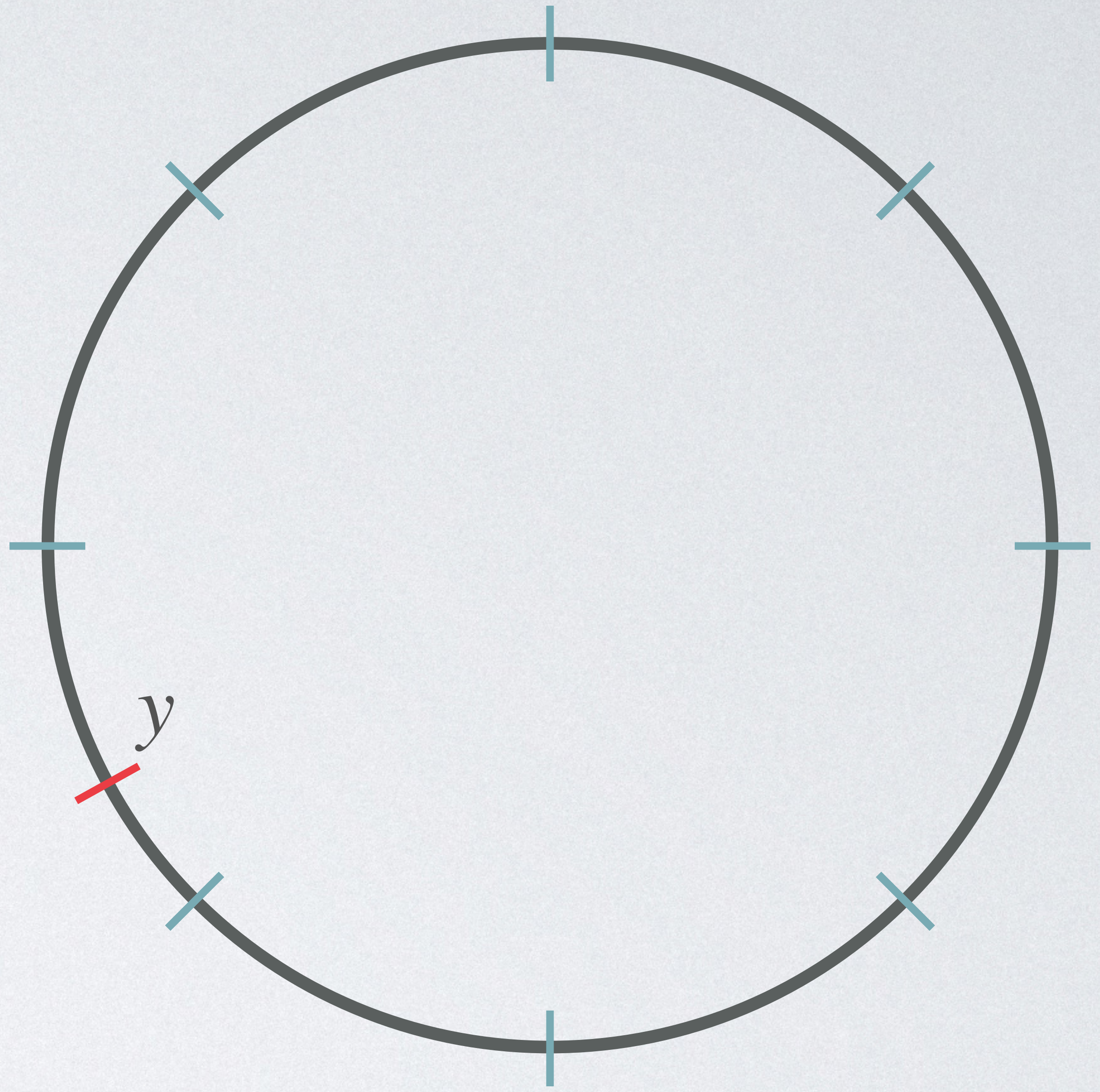
Hellman '80

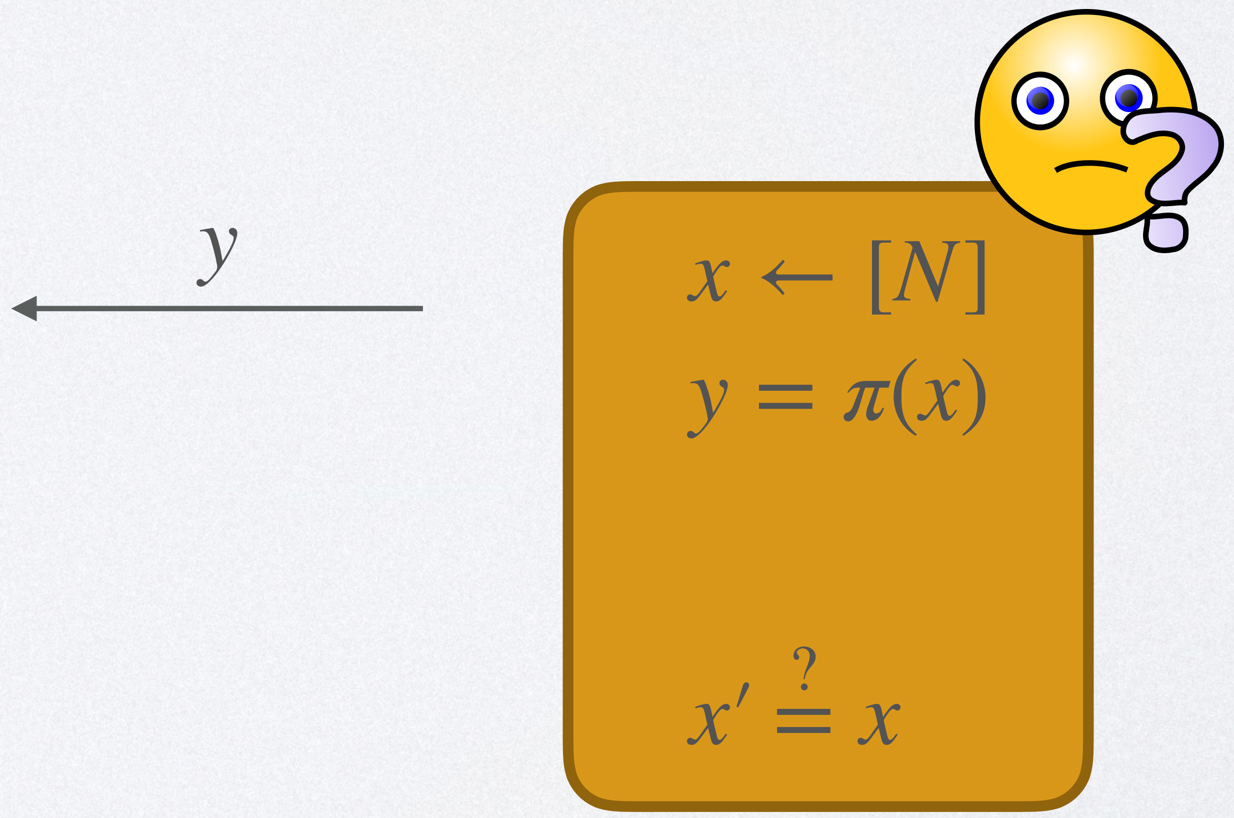
Permutation $\pi : [N] \to [N]$



For every cycle of length at least S,
store points x_i at distance N/S

Advice: $z = (x_1, \ldots, x_S)$



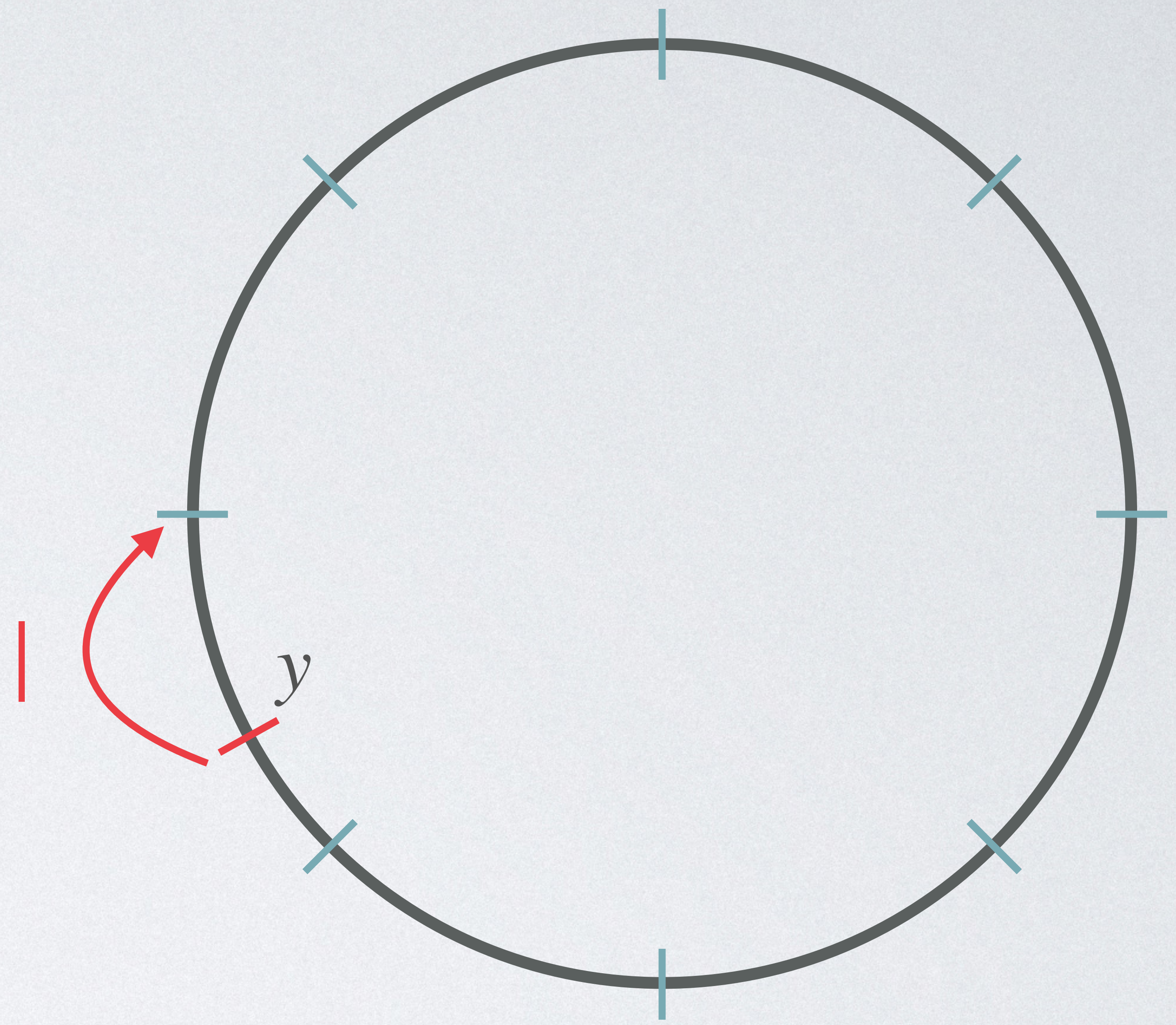


Attacking One-Way Permutations

Hellman '80

Permutation $\pi : [N] \to [N]$

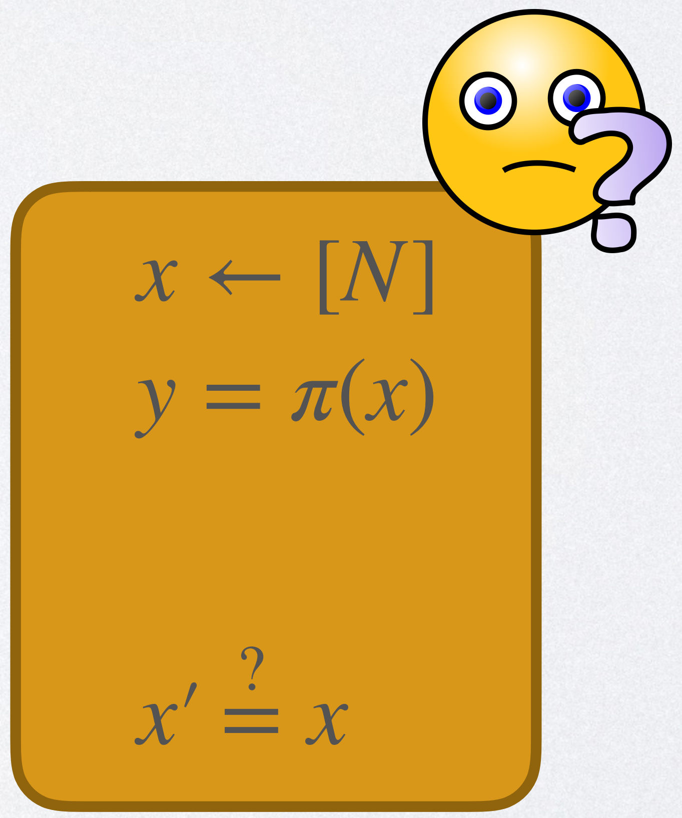
For every cycle of length at least S,
store points x_i at distance N/S



Advice: $z = (x_1, \ldots, x_S)$

Start at y and apply π until hit x_j,

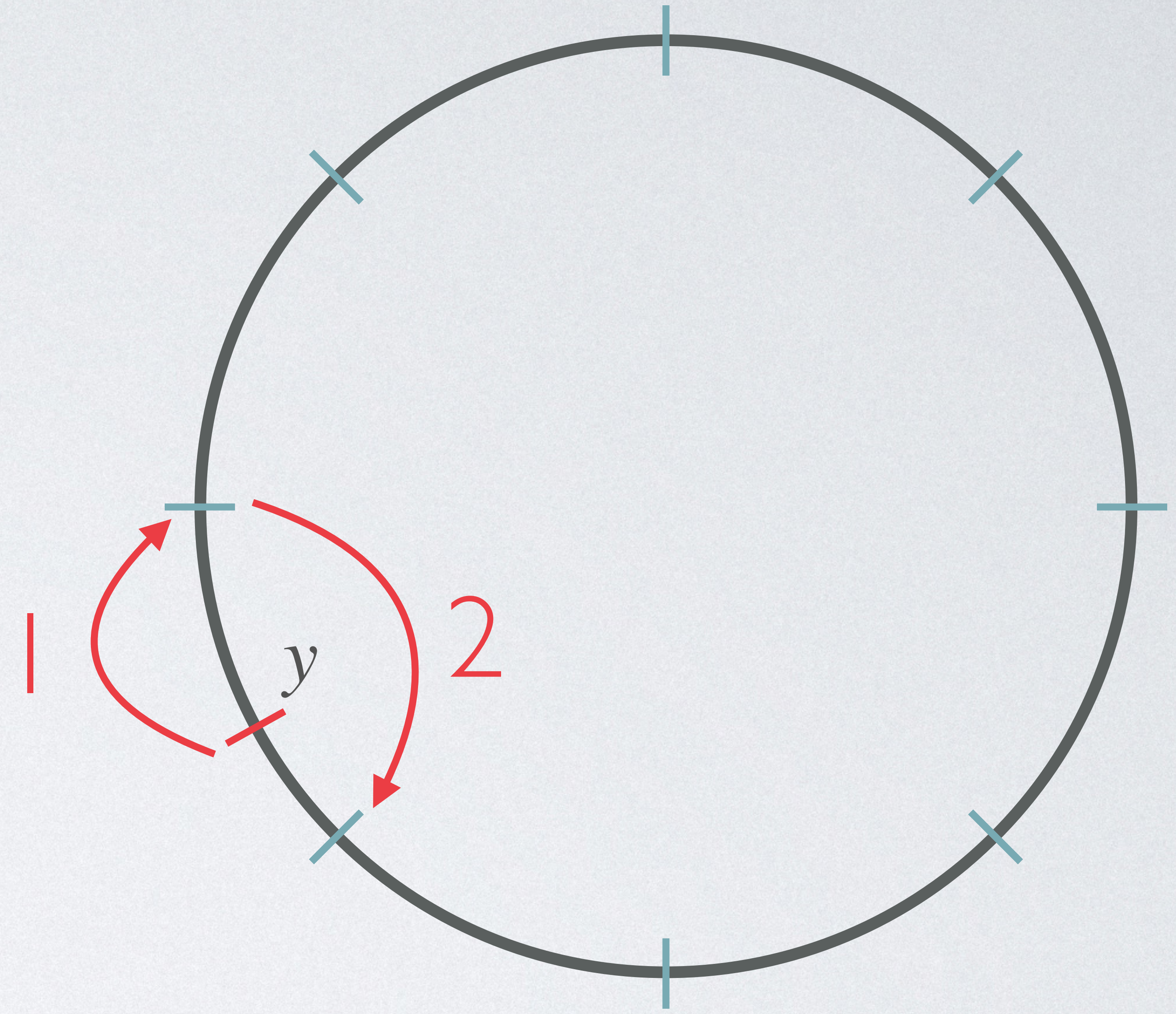
y ←



Attacking One-Way Permutations

Hellman '80

Permutation $\pi : [N] \to [N]$



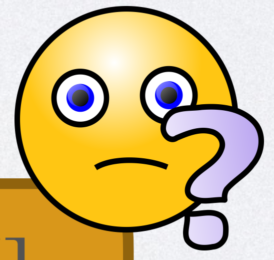
For every cycle of length at least S,
store points x_i at distance N/S

Advice: $z = (x_1, \ldots, x_S)$

Start at y and apply π until hit x_j,

start at x_{j-1} and apply π until hit y,

$\xleftarrow{\;y\;}$



$x \leftarrow [N]$
$y = \pi(x)$

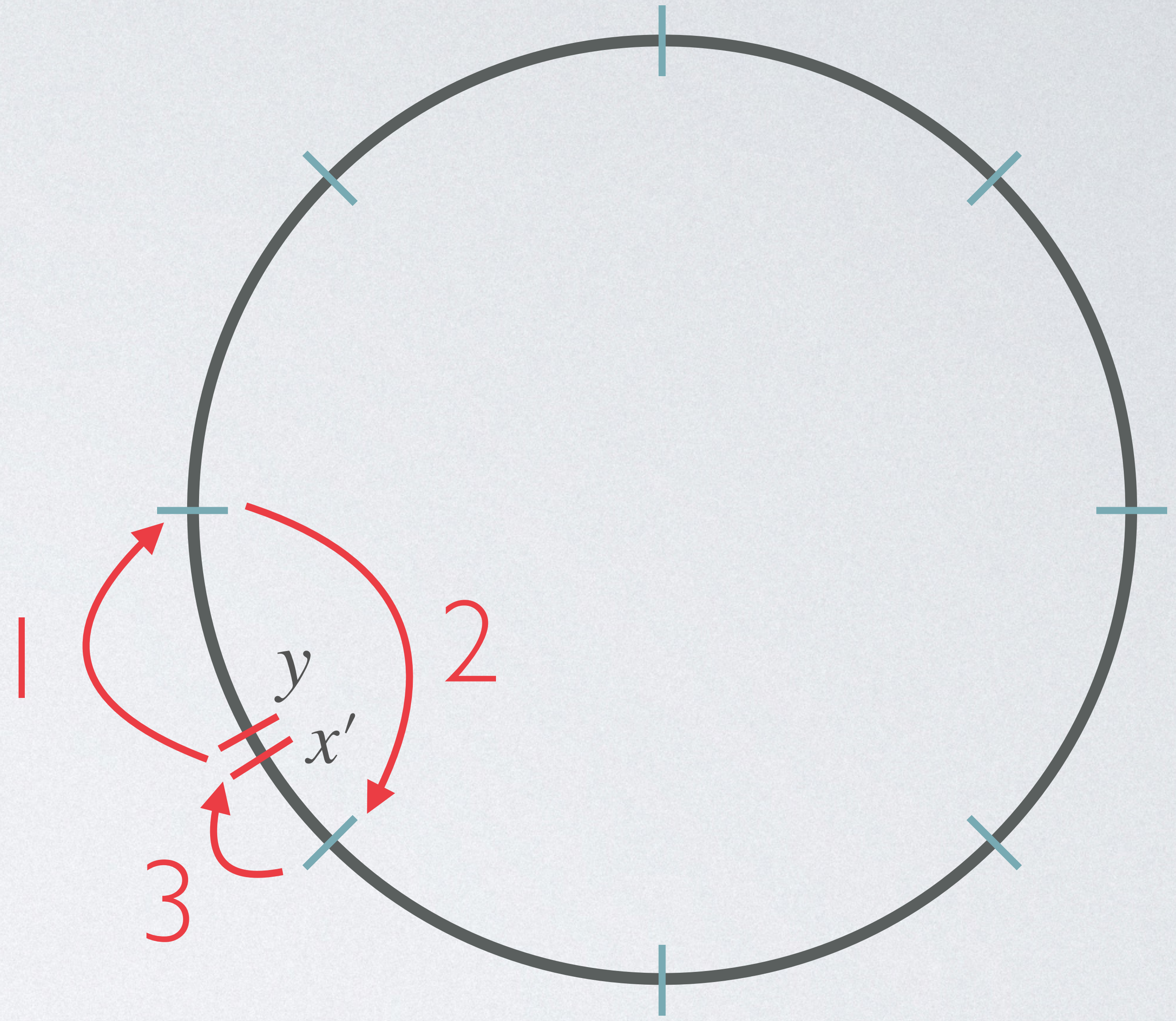
$x' \stackrel{?}{=} x$

Attacking One-Way Permutations

Hellman '80

Permutation $\pi : [N] \to [N]$

For every cycle of length at least S,
store points x_i at distance N/S

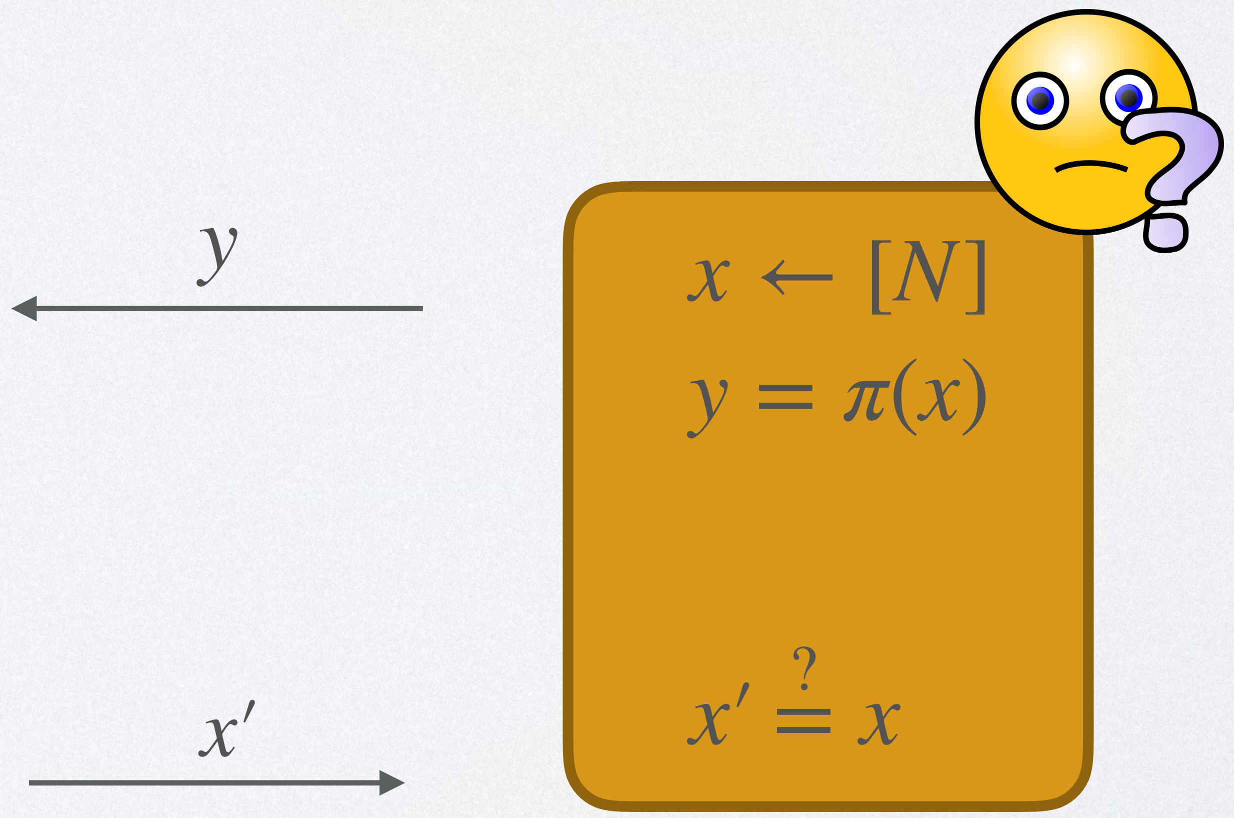


Advice: $z = (x_1, \ldots, x_S)$

Start at y and apply π until hit x_j,

start at x_{j-1} and apply π until hit y,

x': value just before y



Attacking One-Way Permutations

Hellman '80

Permutation $\pi : [N] \to [N]$

For every cycle of length at least S,

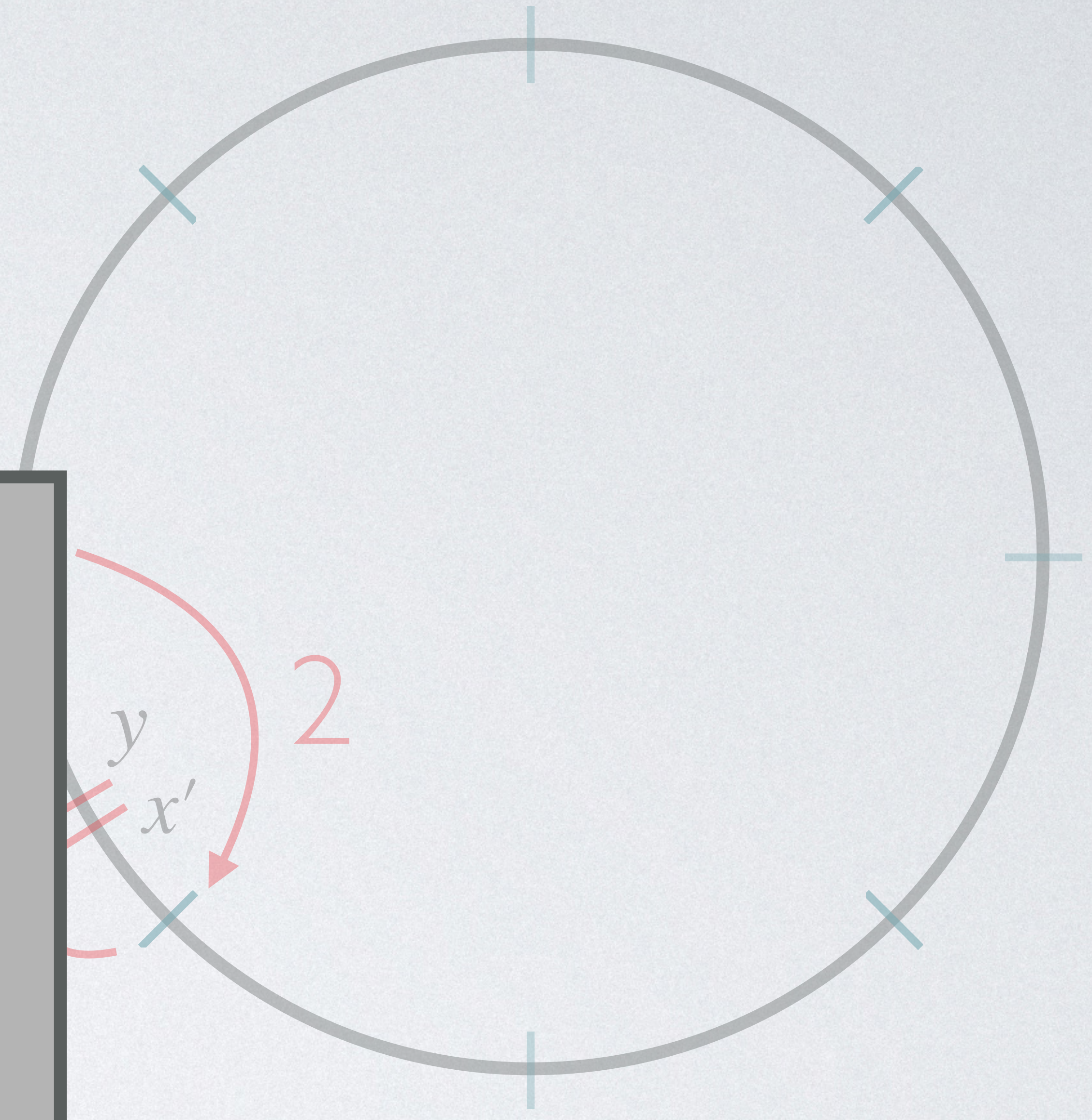
store

Advice:

Space complexity: S

Time complexity: $T = N/S$

Total complexity for $S = T = \sqrt{N}$: $\sqrt{N}$



Start at y and apply π until hit x_j,

start at x_{j-1} and apply π until hit y,

x': value just before y

y ←

x' →

$x \leftarrow [N]$

$y = \pi(x)$

$x' \stackrel{?}{=} x$

Attacking One-Way Permutations

Hellman '80

Permutation $\pi : [N] \to [N]$

For every cycle of length at least S,

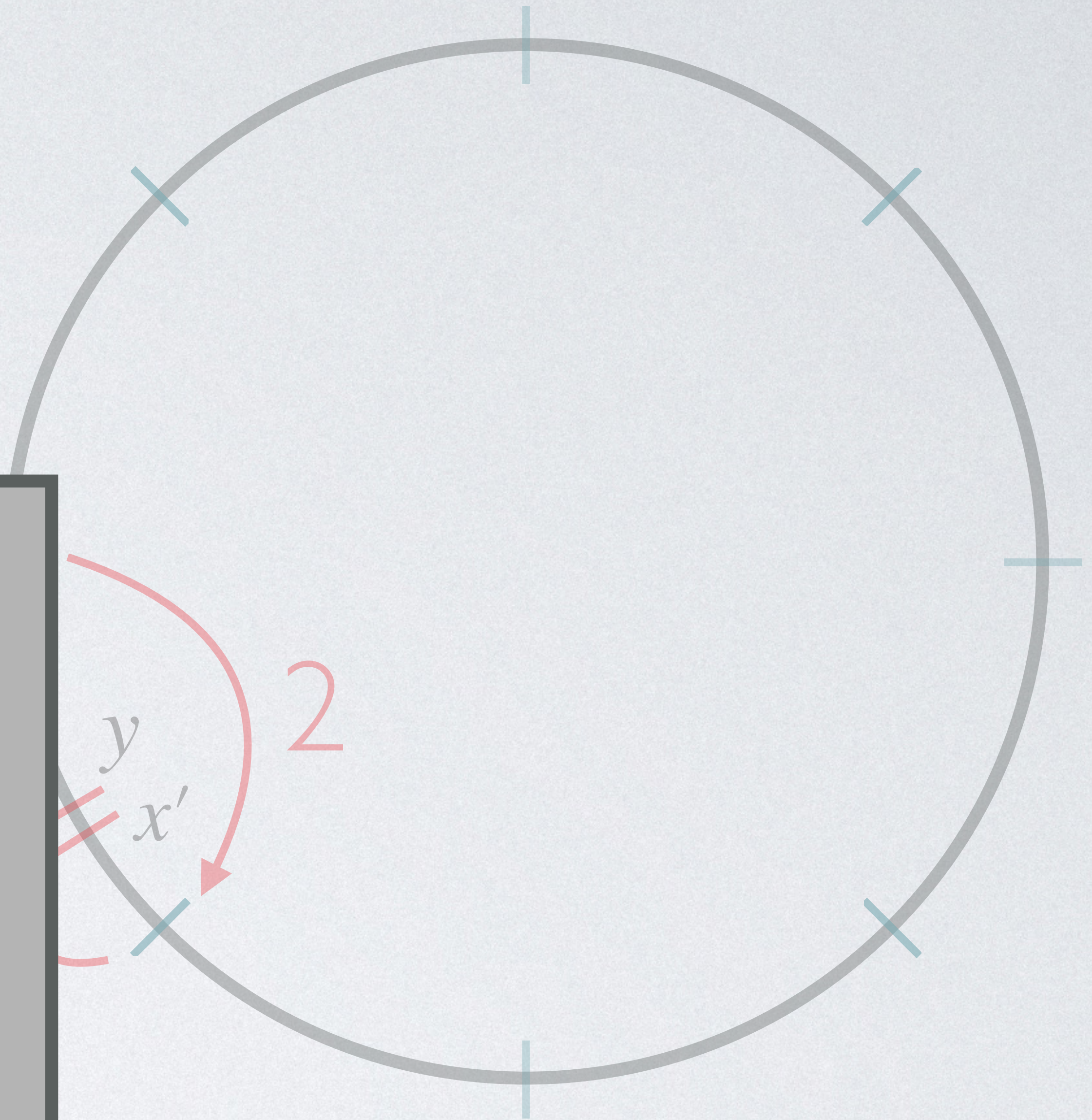
store

Advice:

Space complexity: S

Time complexity: $T = N/S$

Total complexity for $S = T = \sqrt{N}$: $\sqrt{N}$



Start at y and apply π until hit x_j,

start at x_{j-1} and apply π until hit y,

x': value just before y

Analysis in RPM:
security up to N queries

More Preprocessing Attacks

S: Space T: Time

More Preprocessing Attacks

	Bound	Preprocessing Attack	Reference

S: Space T: Time

More Preprocessing Attacks

	Bound	Preprocessing Attack	Reference
OWP	T/N	ST/N	Hellman

S: Space T: Time

More Preprocessing Attacks

	Bound	Preprocessing Attack	Reference
OWP	T/N	ST/N	Hellman
Discrete Logarithms	T^2/N	ST^2/N	Bernstein, Lange; Corrigan-Gibbs, Kogan

S: Space T: Time

More Preprocessing Attacks

	Bound	Preprocessing Attack	Reference
OWP	T/N	ST/N	Hellman
Discrete Logarithms	T^2/N	ST^2/N	Bernstein, Lange; Corrigan-Gibbs, Kogan
Even Mansour	T^2/N	ST^2/N	Fouque, Joux, Mavromati

S: Space T: Time

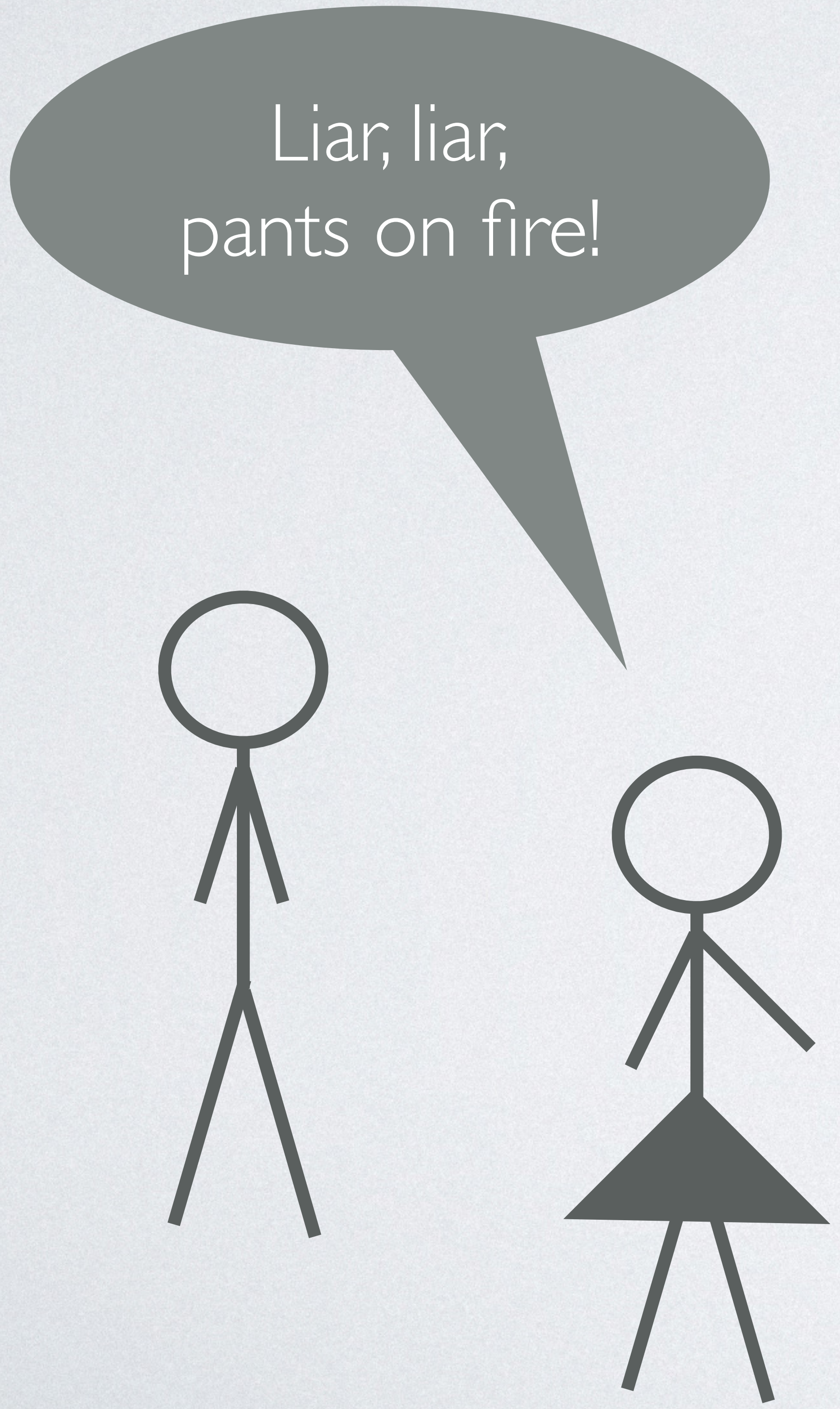
Idealized-Model Methodology

For "natural" applications:

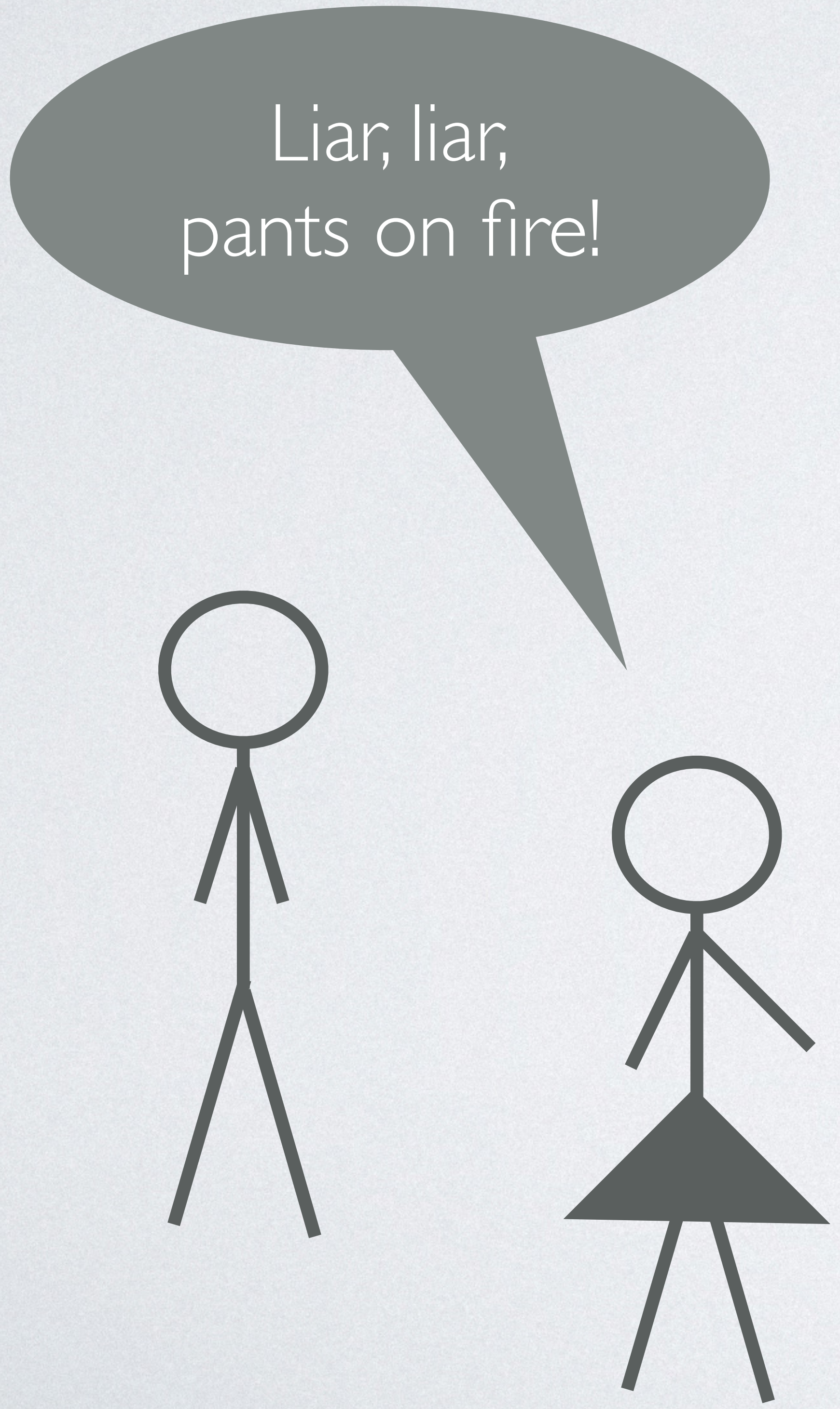
Security in idealized model

=

Security in standard model
using best possible instantiation

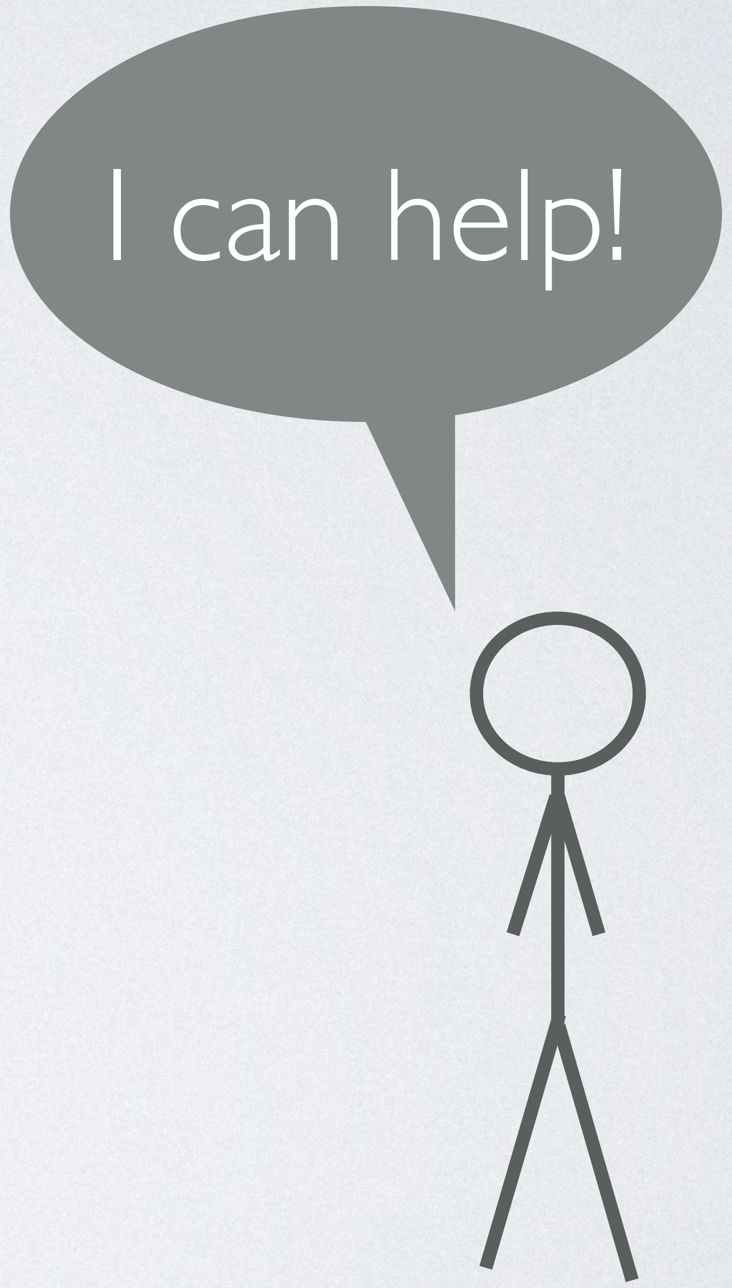


Idealized-Model Methodology



For "natural" applications:

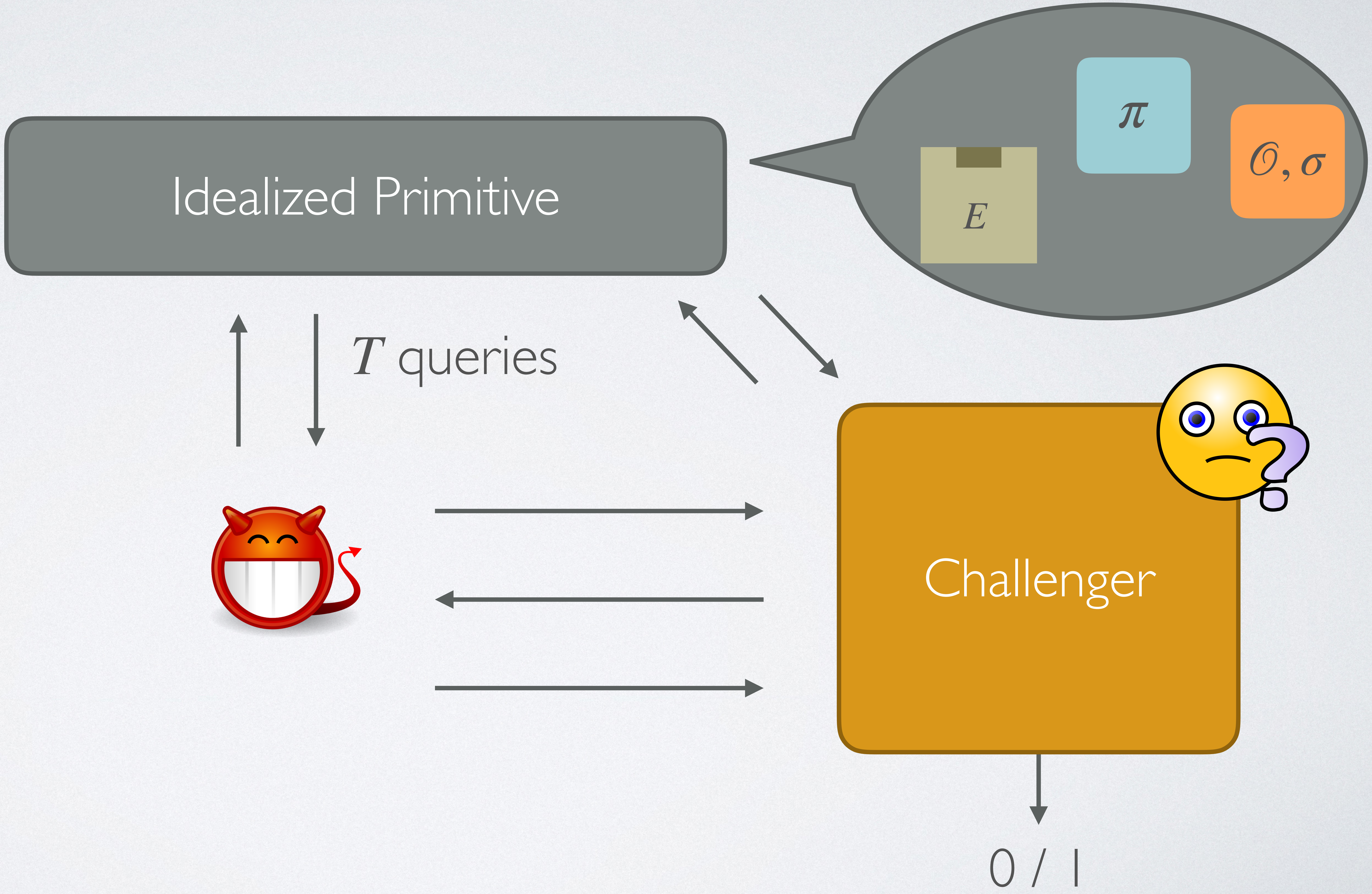
Security in idealized model
=
Security in standard model
using best possible instantiation



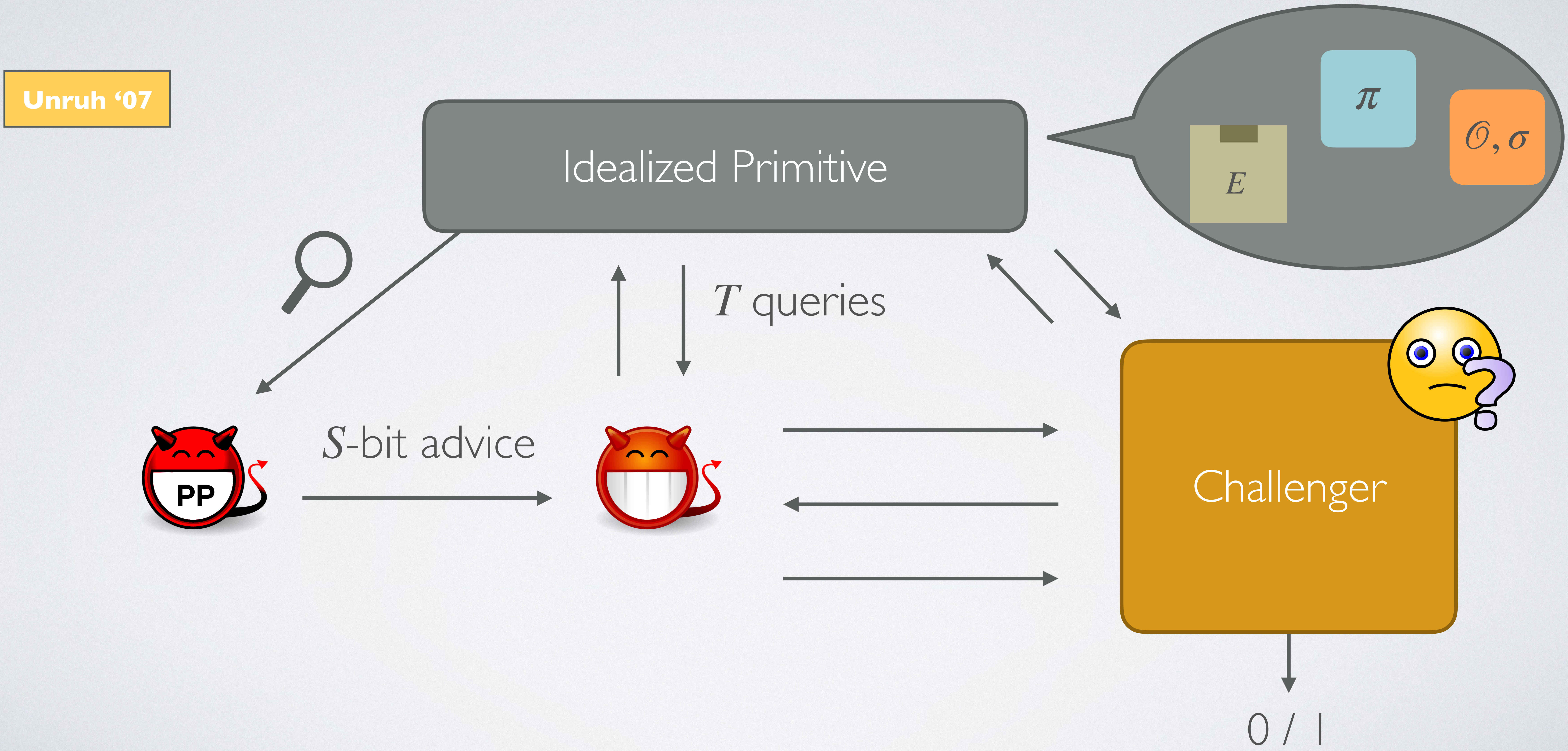
Unruh

Auxiliary-Input (AI) Model

Unruh '07



Auxiliary-Input (AI) Model



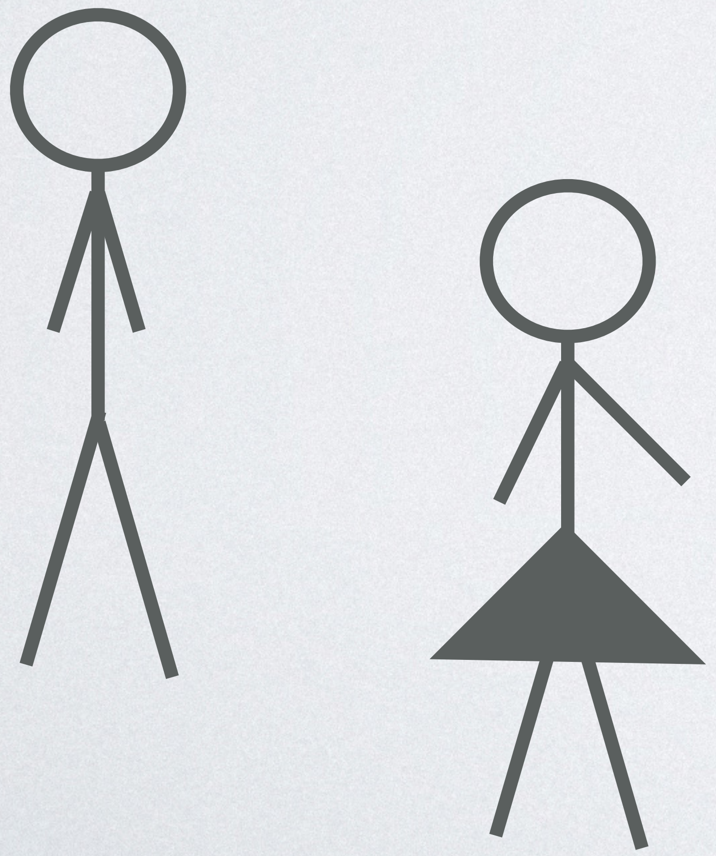
Auxiliary-Input Idealized-Model Methodology

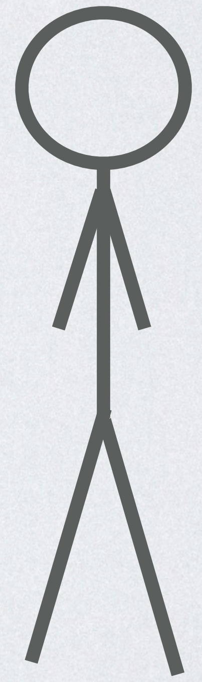
For "natural" applications:

Security in **AI** idealized model

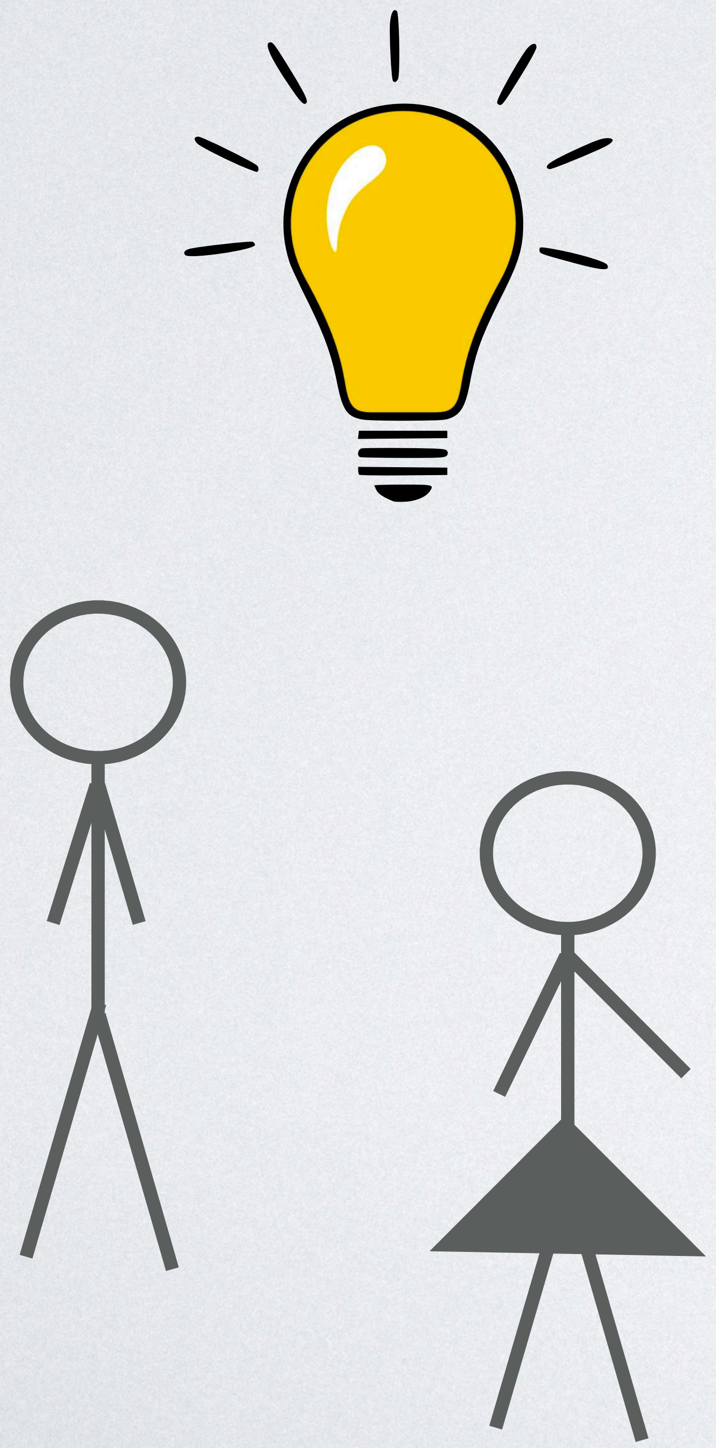
=

Security in **standard model**
against preprocessing attacks
using best possible instantiation



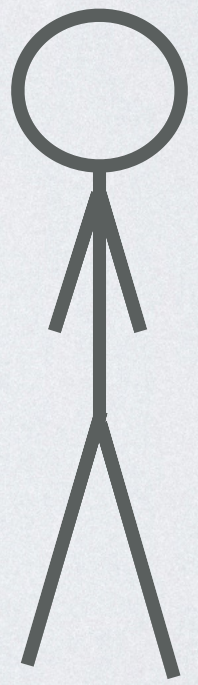


Auxiliary-Input Idealized-Model Methodology



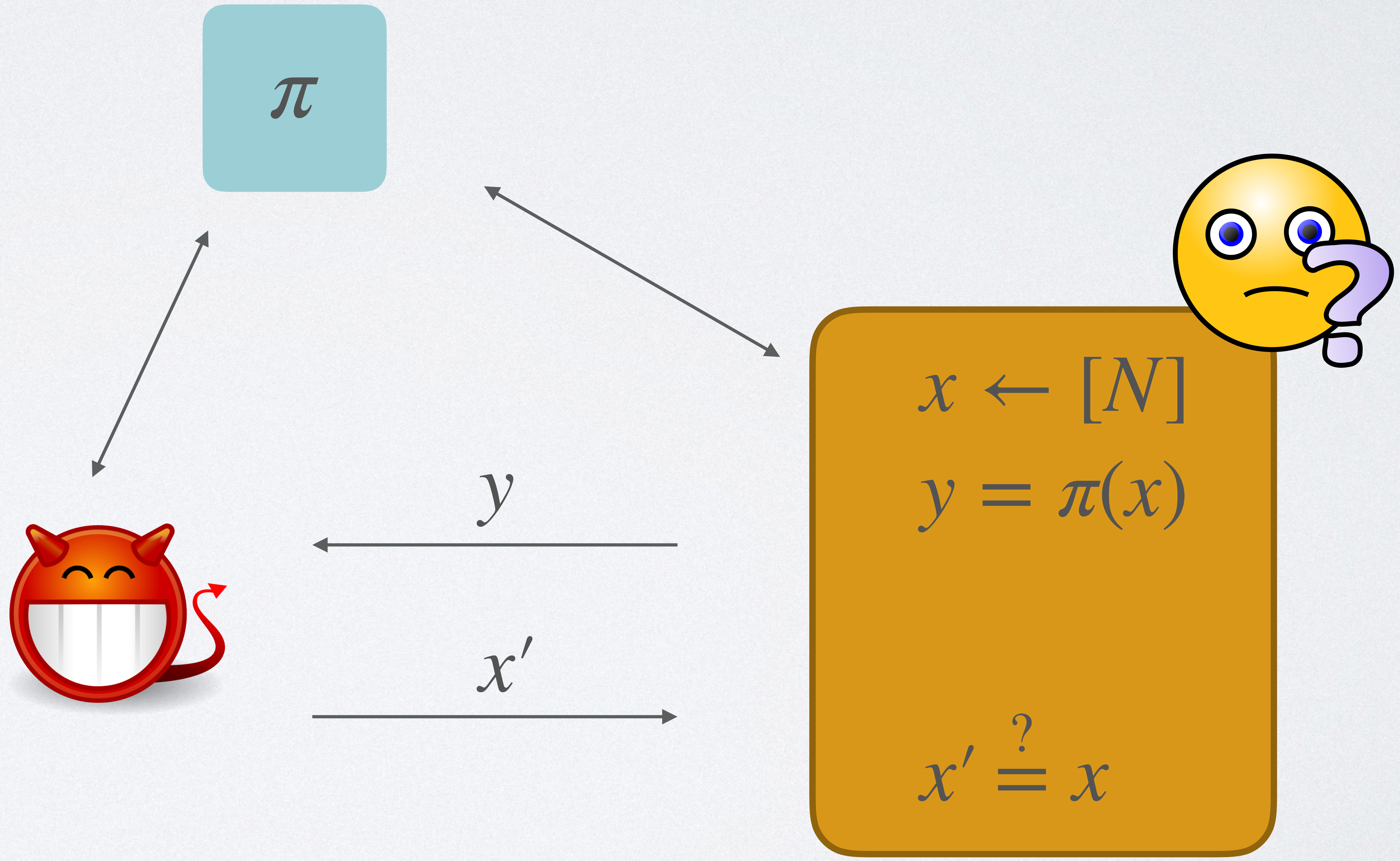
For "natural" applications:

Security in **AI** idealized model
=
Security in **standard model**
against preprocessing attacks
using best possible instantiation



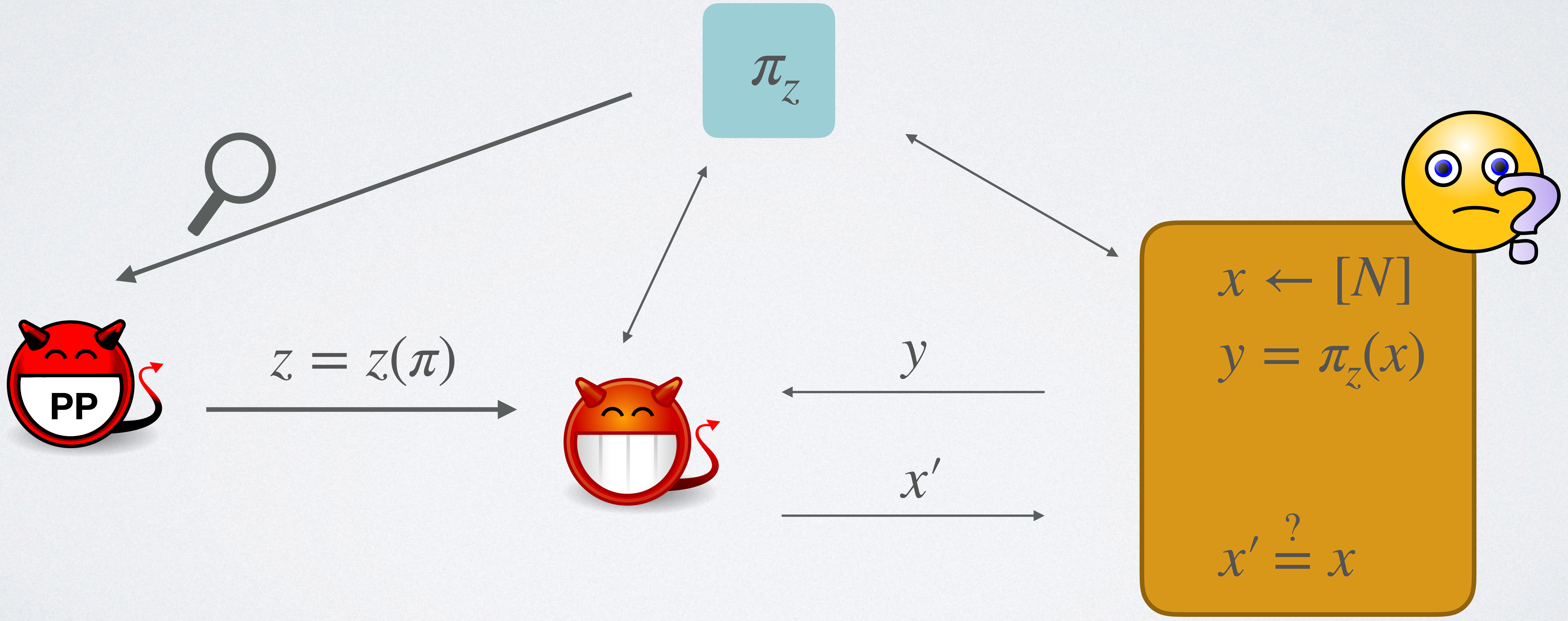
Toy Example: One-Way Permutations

Random permutation $\pi : [N] \to [N]$



Toy Example: One-Way Permutations

Random permutation $\pi : [N] \to [N]$



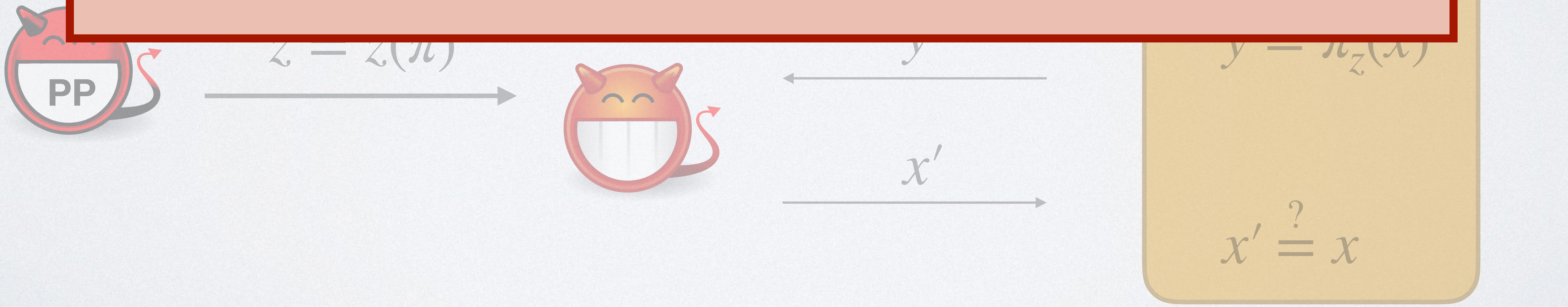
Toy Example: One-Way Permutations

Random permutation $\pi : [N] \to [N]$

Conditioned on z, distribution of π may be ugly:

– Distribution of coordinates unclear

– Dependence of coordinates unclear



Security analysis with auxiliary information seems hard…

AI and the Random-Oracle Model

AI and the Random-Oracle Model

Reference	Technique	Difficulty	Applicability	Bounds	Computational
Unruh '07	Presampling	Easy	Generic	Loose	Limited

AI and the Random-Oracle Model

Reference	Technique	Difficulty	Applicability	Bounds	Computational
Unruh '07	Presampling	Easy	Generic	Loose	Limited
Dodis, Guo, Katz '17	Compression	Hard	OWF, PRG, PRF, CRHF, MAC	Tight	No

AI and the Random-Oracle Model

Reference	Technique	Difficulty	Applicability	Bounds	Computational
Unruh '07	Presampling	Easy	Generic	Loose	Limited
Dodis, Guo, Katz '17	Compression	Hard	OWF, PRG, PRF, CRHF, MAC	Tight	No
C, Dodis, Guo, Steinberger '18	Presampling++	Easy	Generic	Tight	Yes

AI and the Random-Permutation Model

AI and the Random-Permutation Model

Reference	Technique	Difficulty	Applicability	Bounds	Computational
Tessaro '11	Presampling	Easy	Generic	Loose	Limited

AI and the Random-Permutation Model

Reference	Technique	Difficulty	Applicability	Bounds	Computational
Tessaro '11	Presampling	Easy	Generic	Loose	Limited
De, Trevisan, Tulsiani '10	Compression	Hard	OWP	Tight	No

AI and the Random-Permutation Model

Reference	Technique	Difficulty	Applicability	Bounds	Computational
Tessaro '11	Presampling	Easy	Generic	Loose	Limited
De, Trevisan, Tulsiani '10	Compression	Hard	OWP	Tight	No
This work	Presampling++	Easy	Generic	Tight	Yes

AI and the Random-Permutation Model

Reference	Technique	Difficulty	Applicability	Bounds	Computational
Tessaro '11	Presampling	Easy	Generic	Loose	Limited
De, Trevisan, Tulsiani '10	Com…	[illegible]	[illegible]	…ht	No
This work	Presampling++	Easy	Generic	Tight	Yes

Before this work:

No **non-uniform** bounds known for any symmetric primitives

AI and the Generic-Group Model

AI and the Generic-Group Model

Reference	Technique	Difficulty	Applicability	Bounds
Corrigan-Gibbs, Kogan '18	Compression	Hard	DL, CDH, DDH, …	Tight

AI and the Generic-Group Model

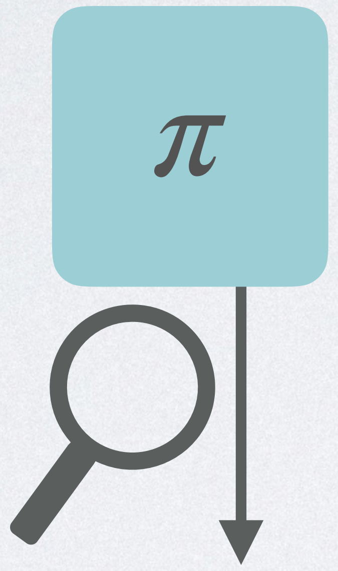
Reference	Technique	Difficulty	Applicability	Bounds
Corrigan-Gibbs, Kogan '18	Compression	Hard	DL, CDH, DDH, …	Tight
This work	Presampling++	Easy	Generic	Tight

Presampling Technique

- Analyze constructions in much simpler so-called **Bit-Fixing (BF) Model**
- Use **generic connection** between AI model and BF model to get AI model bound

Bit-Fixing: Random Permutations

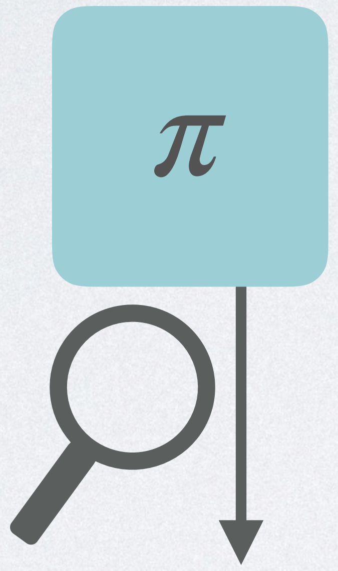
$$\pi : [N] \to [N]$$



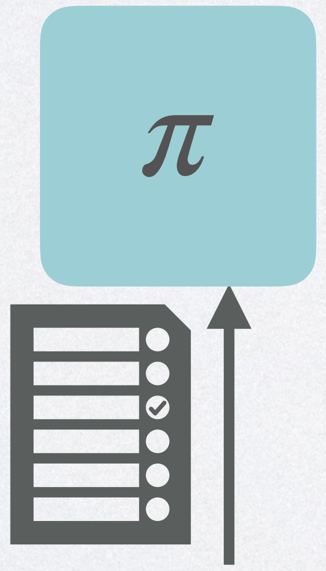
AI-RPM: Leak arbitrary S-bit advice about entire function table

Bit-Fixing: Random Permutations

$$\pi : [N] \to [N]$$



AI-RPM: Leak arbitrary S-bit advice about entire function table



BF-RPM: Prefix arbitrary P coordinates (no collisions)

Bit-Fixing: Ideal Ciphers

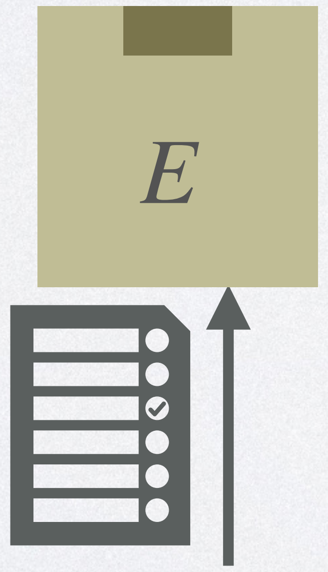
$$E : [K] \times [N] \rightarrow [N]$$

AI-ICM: Leak arbitrary S-bit advice about entire function table

Bit-Fixing: Ideal Ciphers

$$E : [K] \times [N] \to [N]$$

AI-ICM: Leak arbitrary S-bit advice about entire function table



BF-ICM: Prefix arbitrary P coordinates (no collisions for each key)

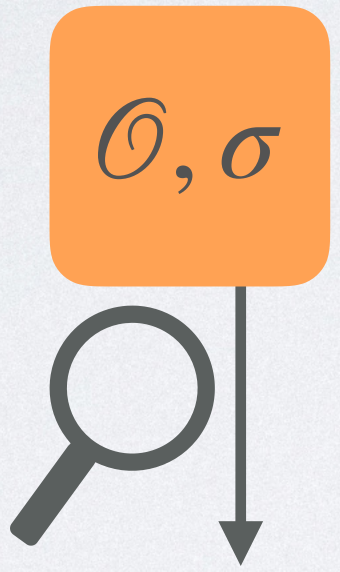
Bit-Fixing: Generic Groups

$$\sigma : [N] \to [M]$$

AI-GGM: Leak arbitrary S-bit advice about entire function table of σ

Bit-Fixing: Generic Groups

$$\sigma : [N] \to [M]$$



AI-GGM: Leak arbitrary S-bit advice about entire function table of σ

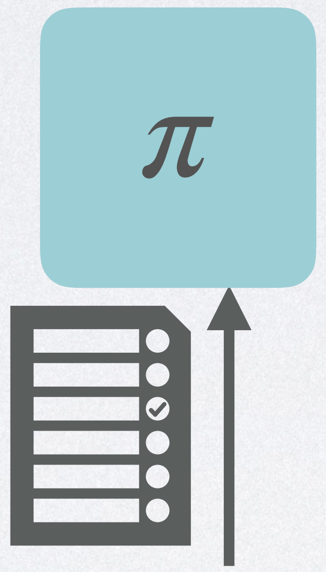
BF-GGM: Prefix arbitrary P coordinates of σ (no collisions)

Bit-Fixing to Auxiliary Input

Theorem:

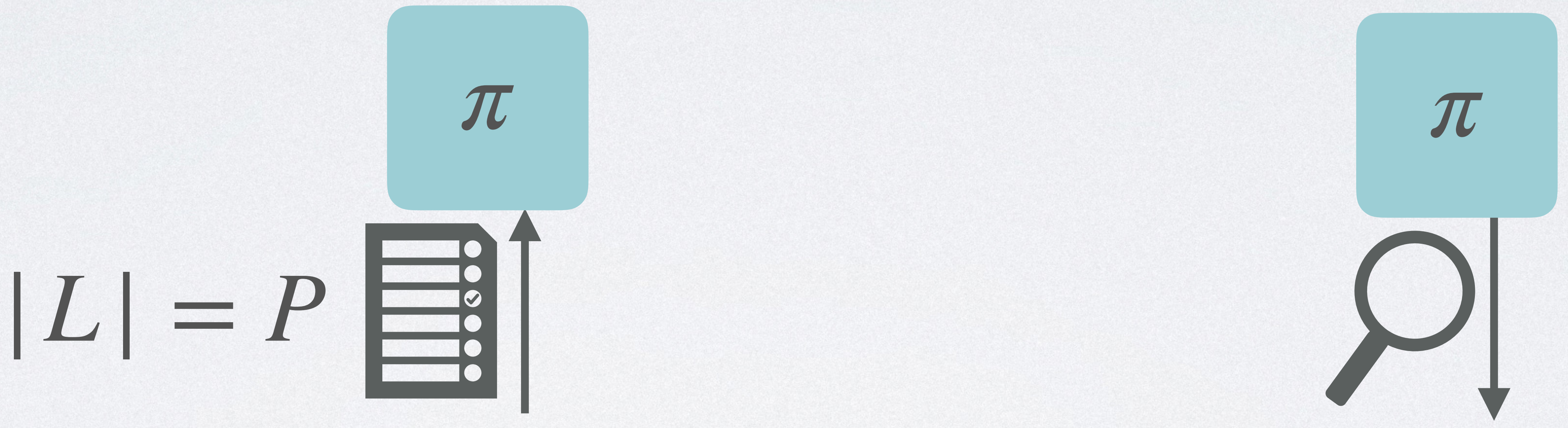
Bit-Fixing to Auxiliary Input

$|L| = P$



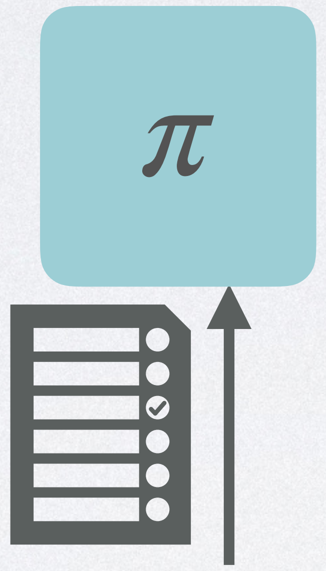
Theorem: (S, T, ε)-secure

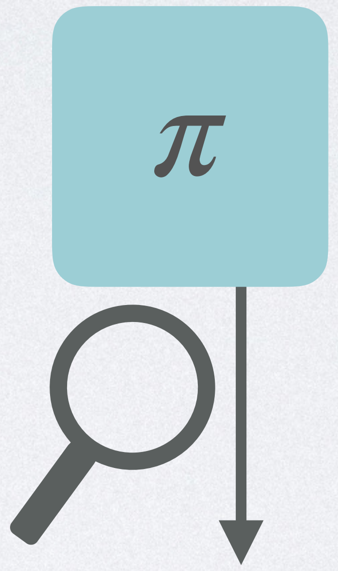
Bit-Fixing to Auxiliary Input



Theorem: (S, T, ε)-secure $\implies$ (S, T, ε')-secure

Bit-Fixing to Auxiliary Input

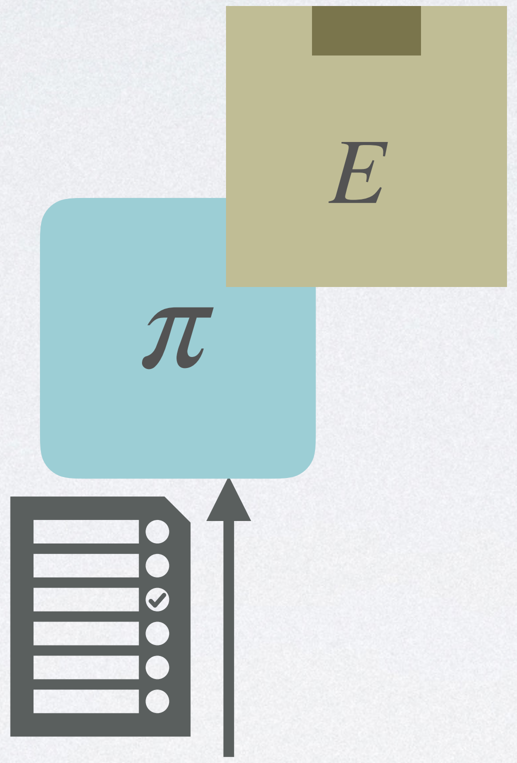
$|L| = P$ 

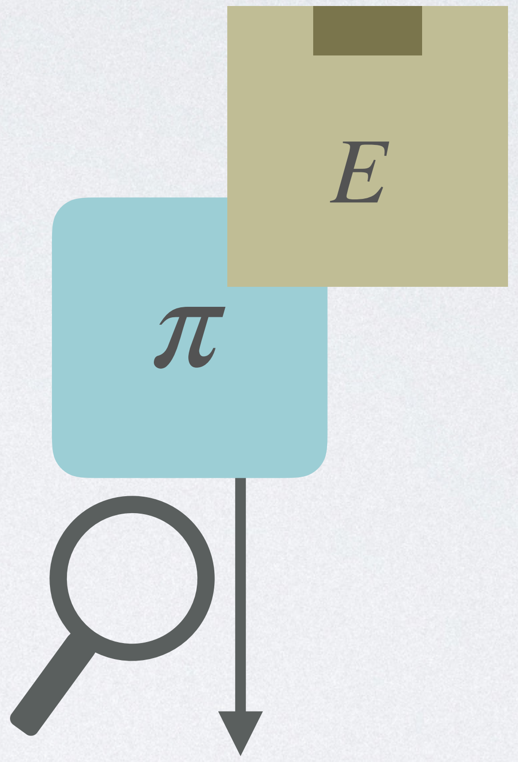


Theorem: (S, T, ε)-secure $\Longrightarrow$ (S, T, ε')-secure

where $\varepsilon' \leq \varepsilon + \frac{ST}{P}$

Bit-Fixing to Auxiliary Input

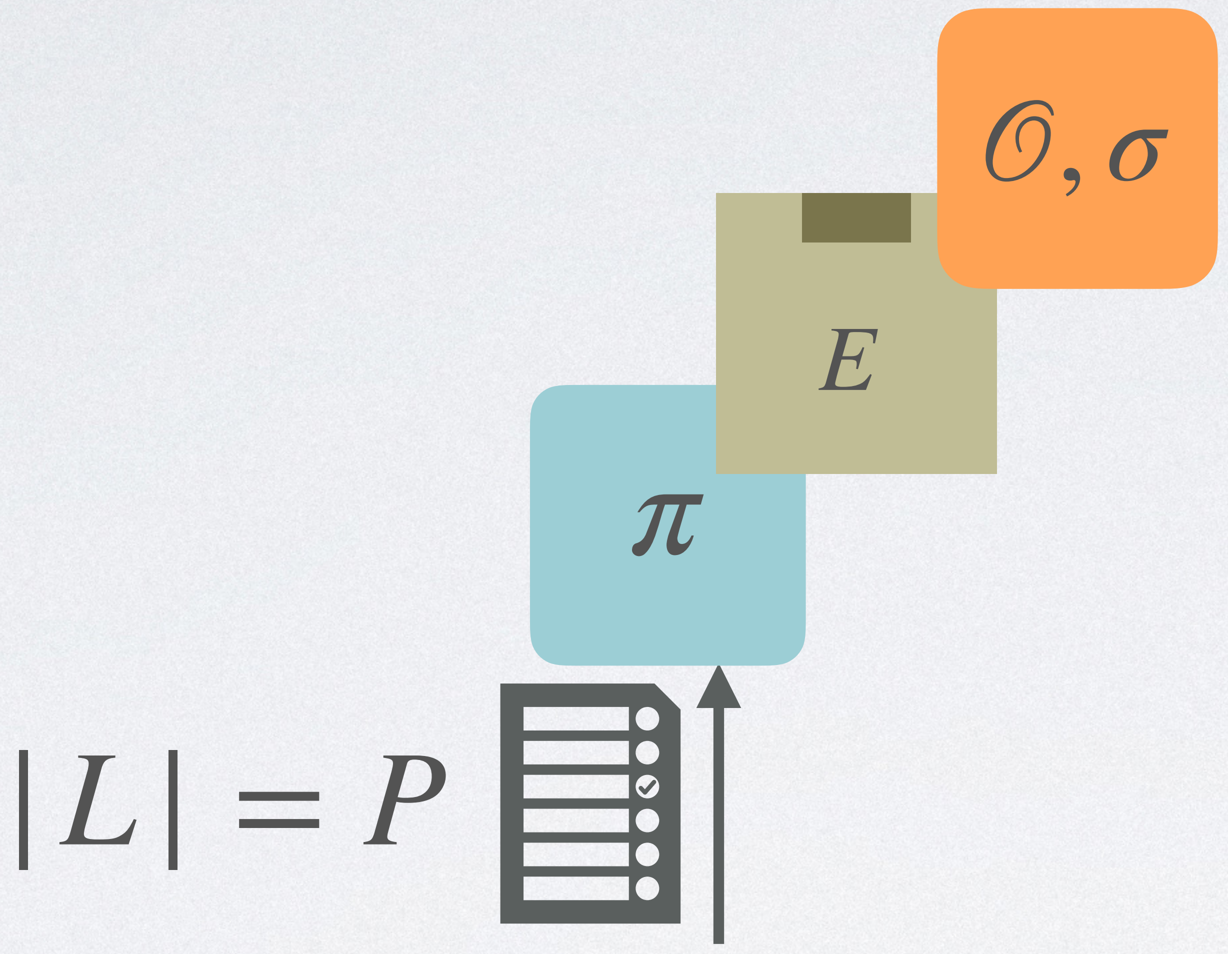
$|L| = P$ 

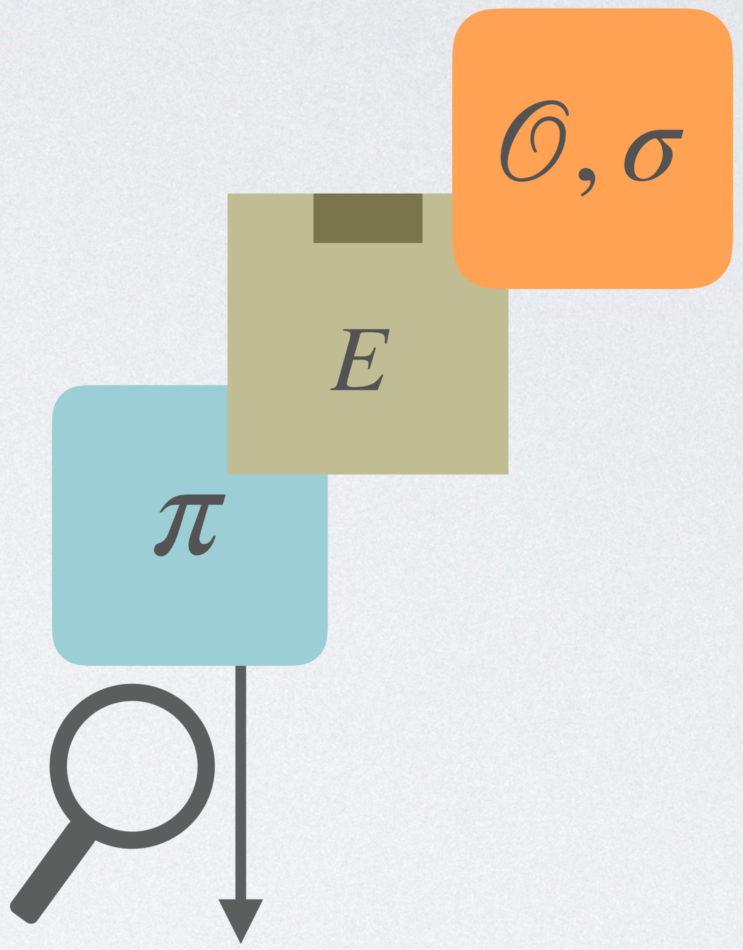


Theorem: (S, T, ε)-secure $\implies$ (S, T, ε')-secure

where $\varepsilon' \leq \varepsilon + \dfrac{ST}{P}$

Bit-Fixing to Auxiliary Input

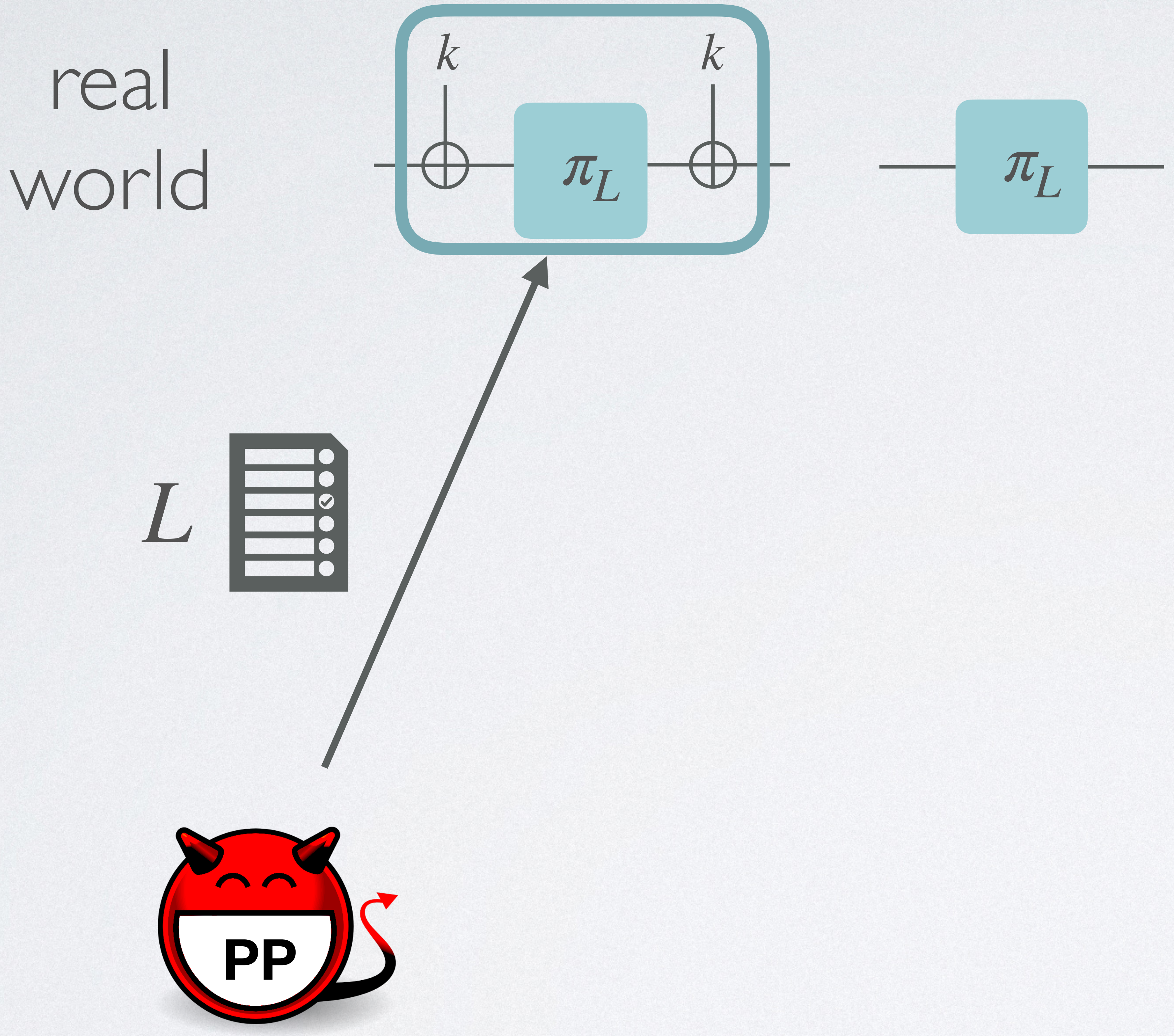




Theorem: (S, T, ε)-secure $\implies$ (S, T, ε')-secure

where $\varepsilon' \leq \varepsilon + \dfrac{ST}{P}$

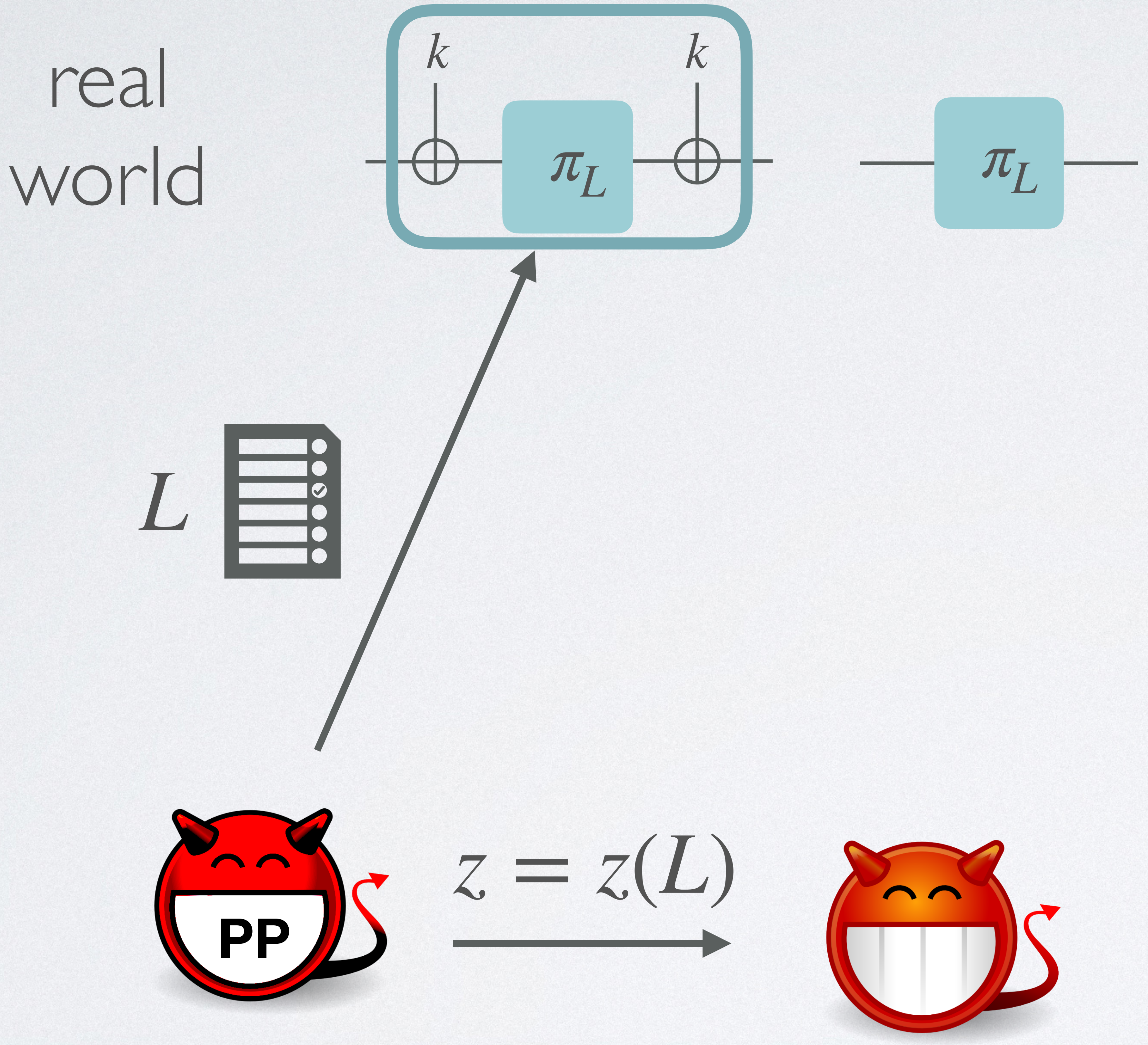
EM Cipher in BF Model



P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$

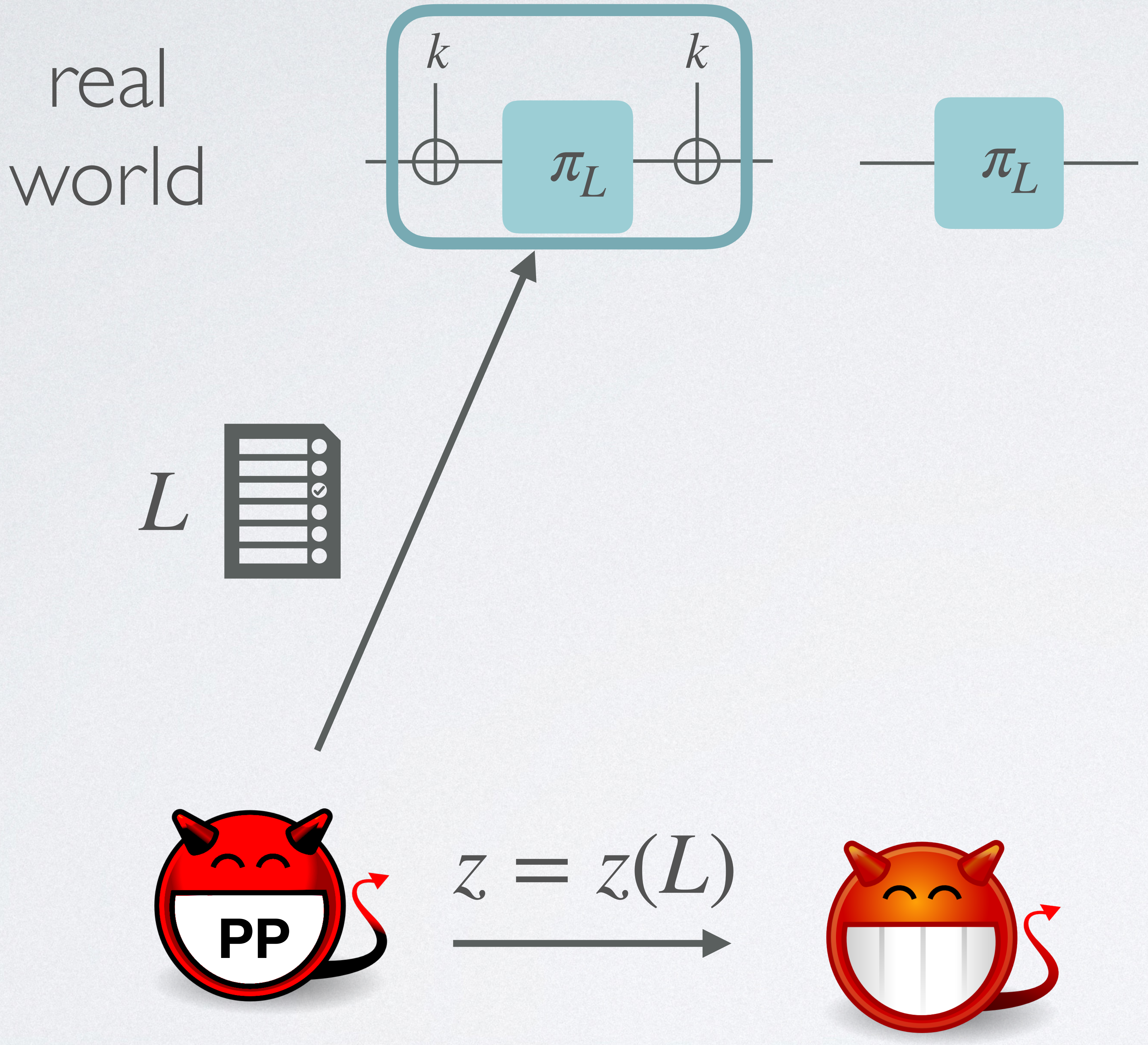
EM Cipher in BF Model



P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$

EM Cipher in BF Model



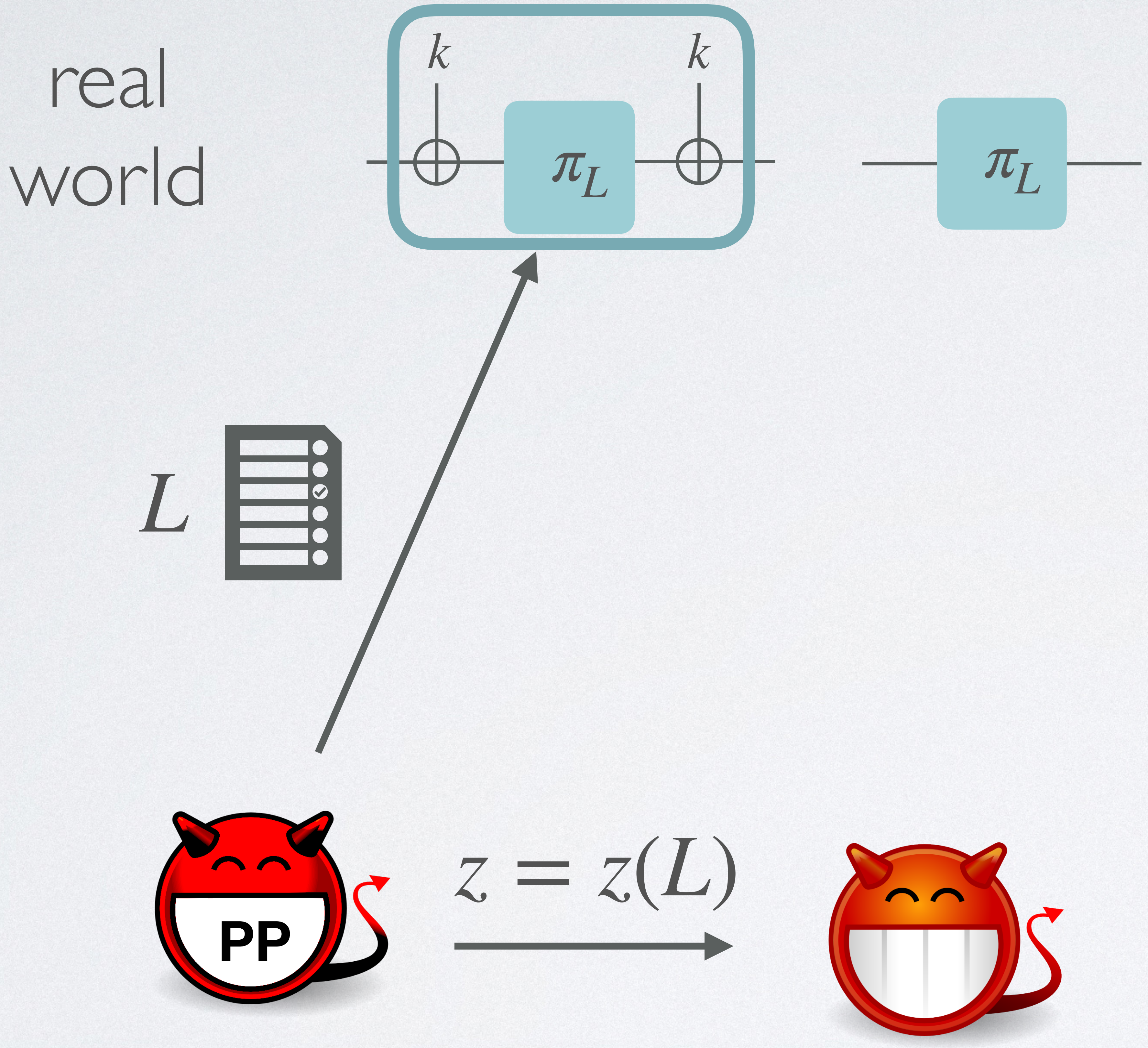
P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$

q construction queries

$\{u_i, v_i\}_{i=1}^{q}$

EM Cipher in BF Model



P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$

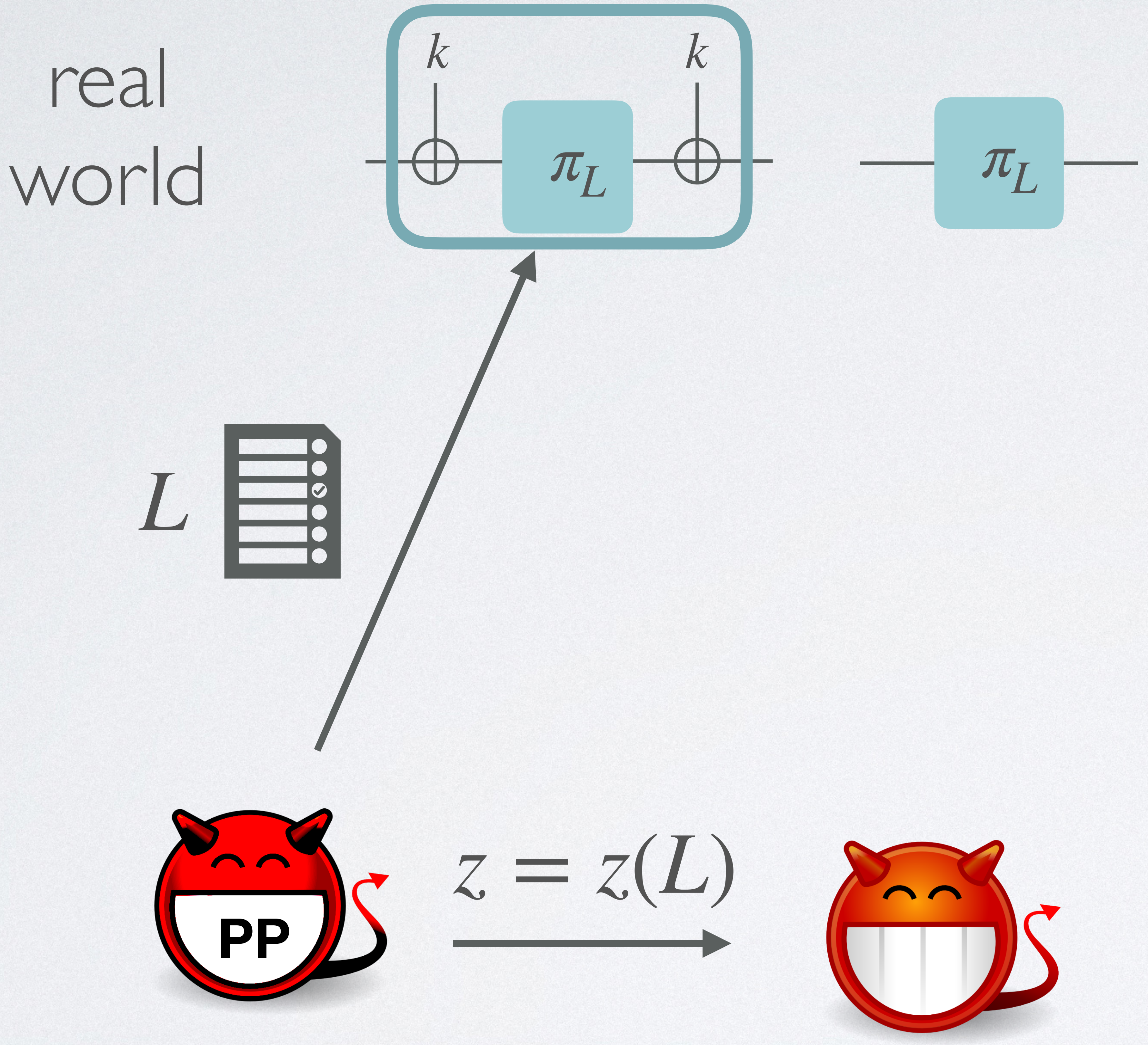
q construction queries

$\{u_i, v_i\}_{i=1}^{q}$

T primitive queries

$\{x_j, y_j\}_{j=1}^{T}$

EM Cipher in BF Model



P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$

q construction queries

$\{u_i, v_i\}_{i=1}^{q}$

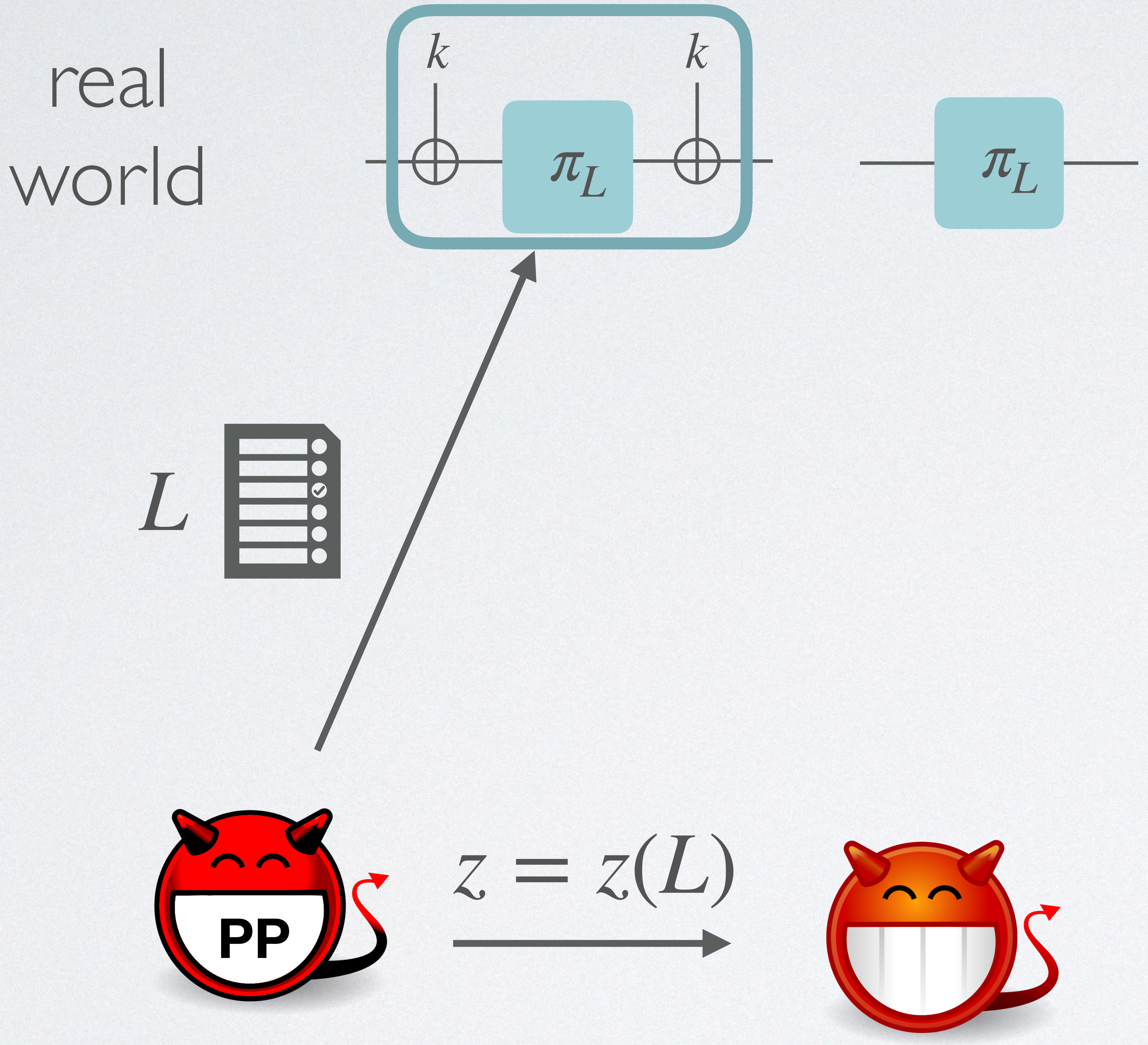
T primitive queries

$\{x_j, y_j\}_{j=1}^{T}$

Event **BAD**:

$\exists i, j : \; u_i \oplus k = x_j \; \vee \; v_i \oplus k = y_j$

EM Cipher in BF Model



P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$

q construction queries

$\{u_i, v_i\}_{i=1}^{q}$

T primitive queries

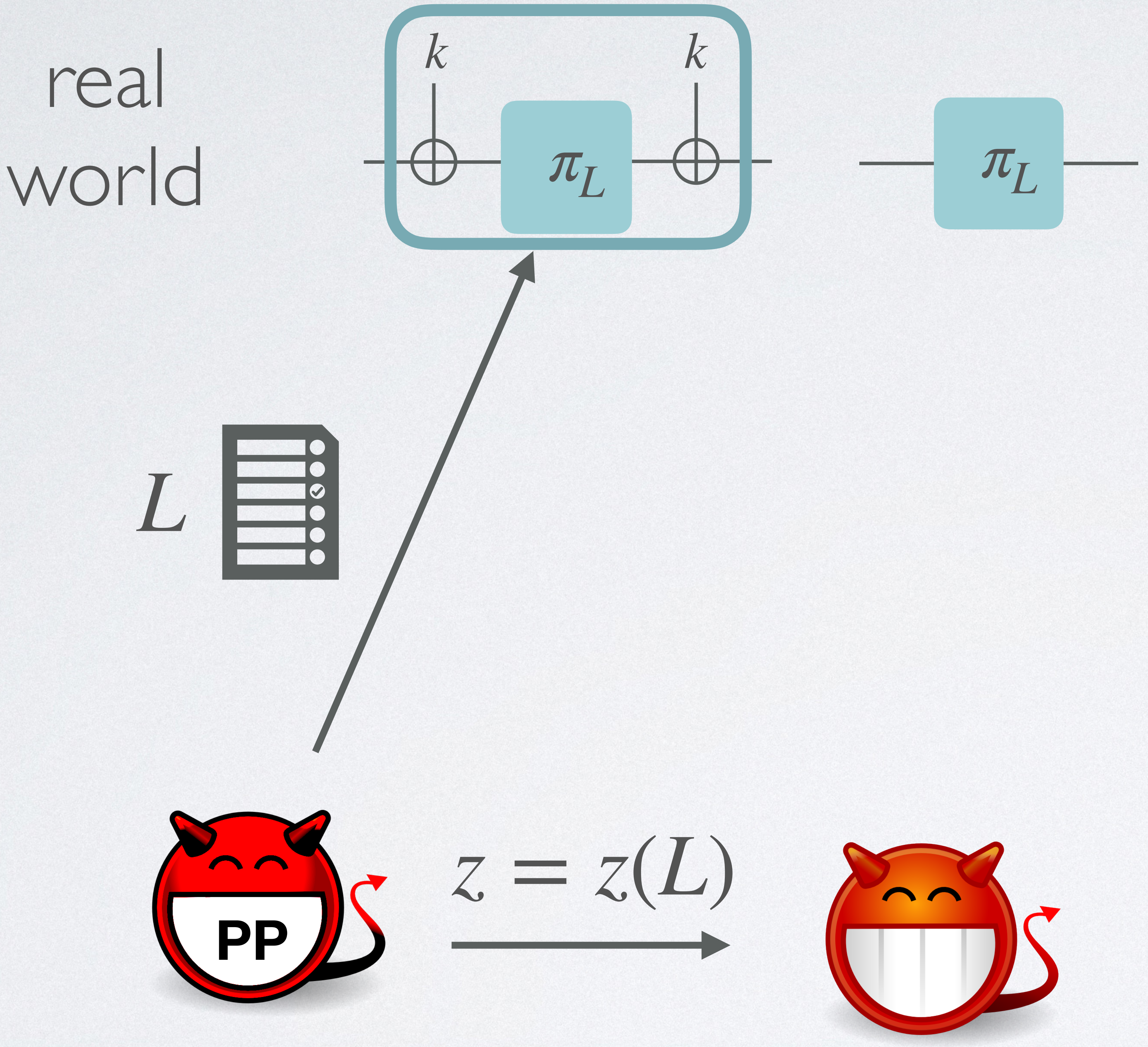
$\{x_j, y_j\}_{j=1}^{T}$

Event **BAD**:

$$\exists i, j : \; u_i \oplus k = x_j \;\vee\; v_i \oplus k = y_j$$

$$\exists i, j : \; u_i \oplus k = \tilde{x}_j \;\vee\; v_i \oplus k = \tilde{y}_j$$

EM Cipher in BF Model



P prefixed coordinates

$$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$$

q construction queries

$$\{u_i, v_i\}_{i=1}^{q}$$

T primitive queries

$$\{x_j, y_j\}_{j=1}^{T}$$

Event **BAD**:

$$\exists i, j: \; u_i \oplus k = x_j \; \vee \; v_i \oplus k = y_j$$

$$\exists i, j: \; u_i \oplus k = \tilde{x}_j \; \vee \; v_i \oplus k = \tilde{y}_j$$

$$\text{P[BAD]} \leq \frac{qT}{2^n} + \frac{qP}{2^n}$$

EM Cipher in BF Model

Bound in BF-RPM: $$\frac{qT}{2^n} + \frac{qP}{2^n}$$

EM Cipher in BF Model

Bound in BF-RPM: $$\frac{qT}{2^n} + \frac{qP}{2^n}$$

Bound in AI-RPM: $$\frac{qT}{2^n} + \frac{qP}{2^n} + \frac{ST}{P}$$

EM Cipher in BF Model

Bound in BF-RPM: $$\frac{qT}{2^n} + \frac{qP}{2^n}$$

Bound in AI-RPM: $$\frac{qT}{2^n} + \boxed{\frac{qP}{2^n} + \frac{ST}{P}}$$

EM Cipher in BF Model

Bound in BF-RPM: $\frac{qT}{2^n} + \frac{qP}{2^n}$

Bound in AI-RPM: $\frac{qT}{2^n} + \boxed{\frac{qP}{2^n} + \frac{ST}{P}} \longrightarrow P \approx \sqrt{STN/q}$

EM Cipher in BF Model

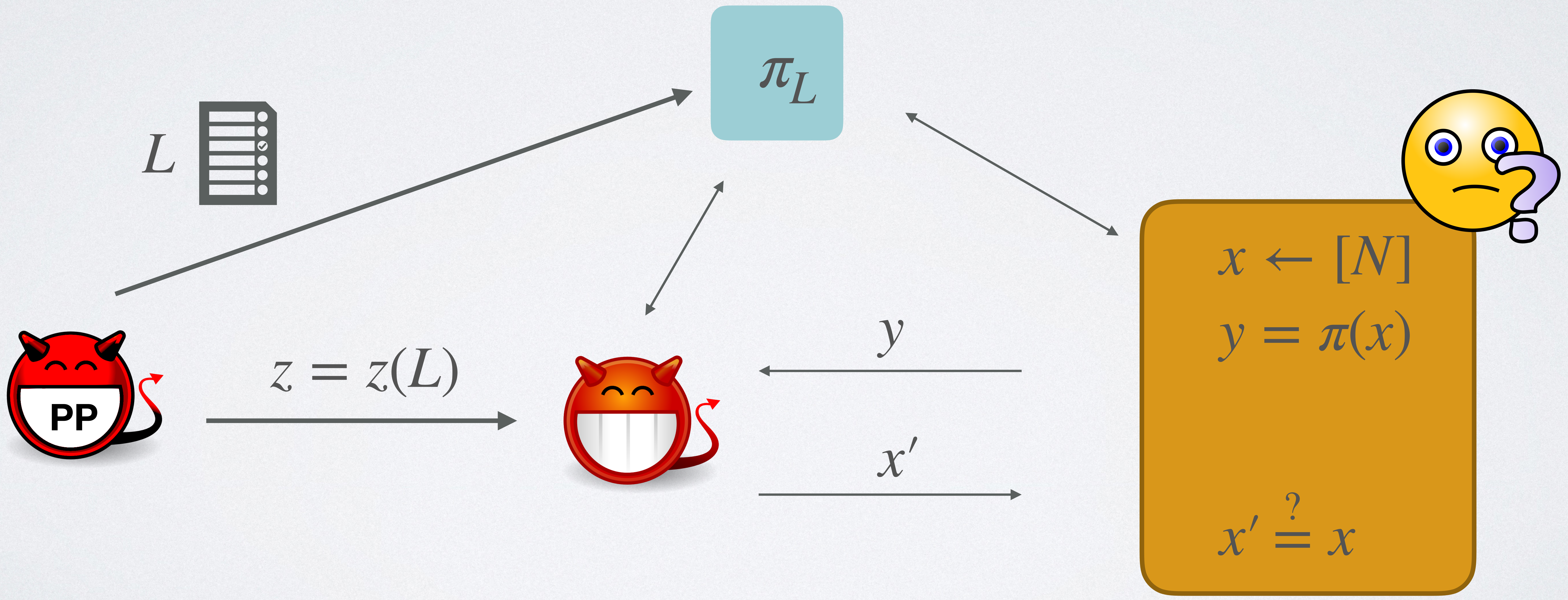
Bound in BF-RPM: $\frac{qT}{2^n} + \frac{qP}{2^n}$

Bound in AI-RPM: $\frac{qT}{2^n} + \boxed{\frac{qP}{2^n} + \frac{ST}{P}} \longrightarrow P \approx \sqrt{STN/q}$

$$= \frac{qT}{2^n} + \sqrt{\frac{STq}{N}}$$

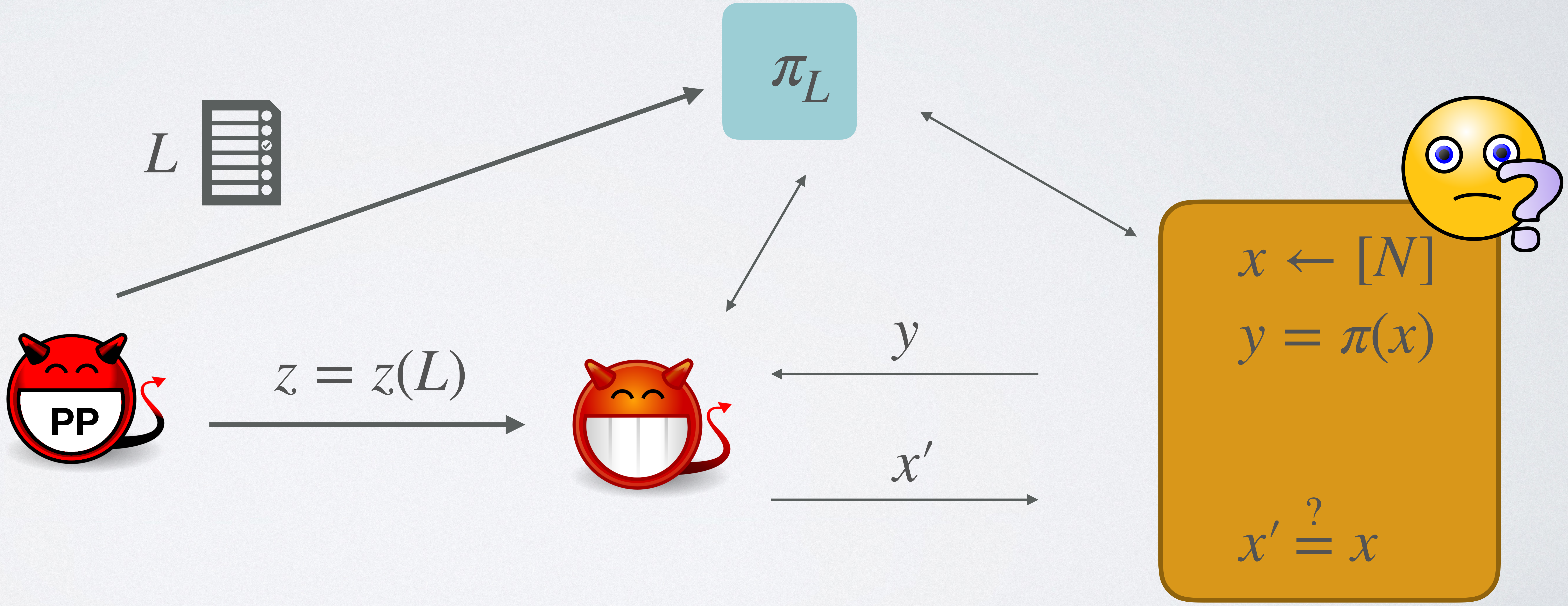
OWP in BF Model

Random permutation $\pi : [N] \to [N]$



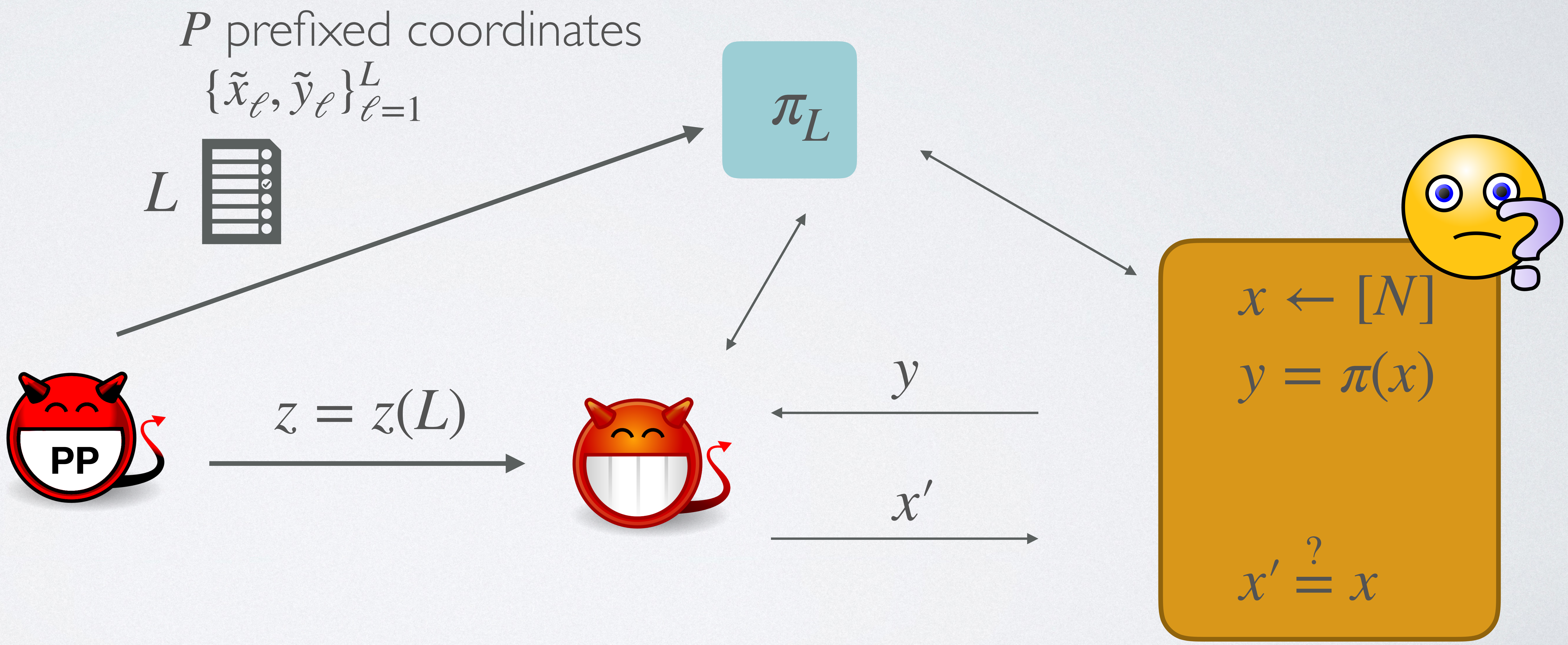
OWP in BF Model

Prefixed random permutation $\pi_L : [N] \to [N]$



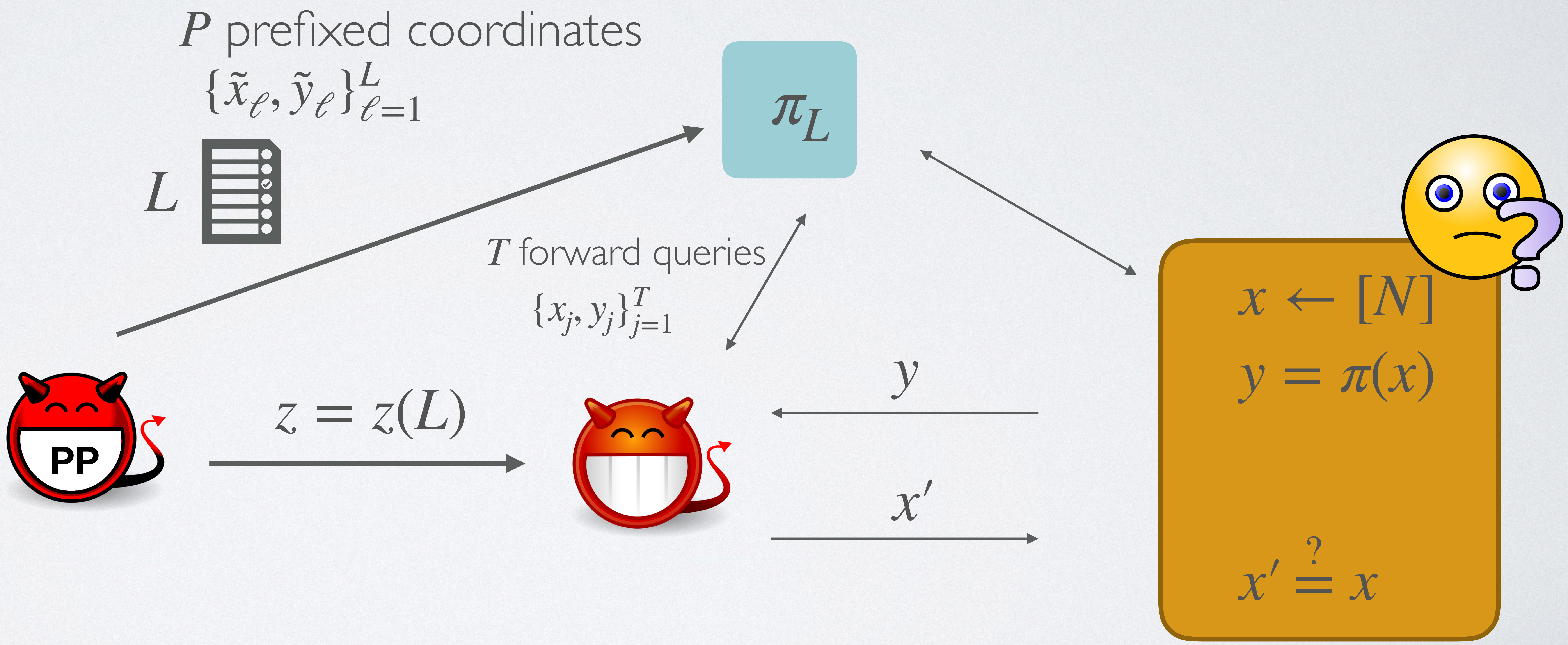
OWP in BF Model

Prefixed random permutation $\pi_L : [N] \to [N]$



OWP in BF Model

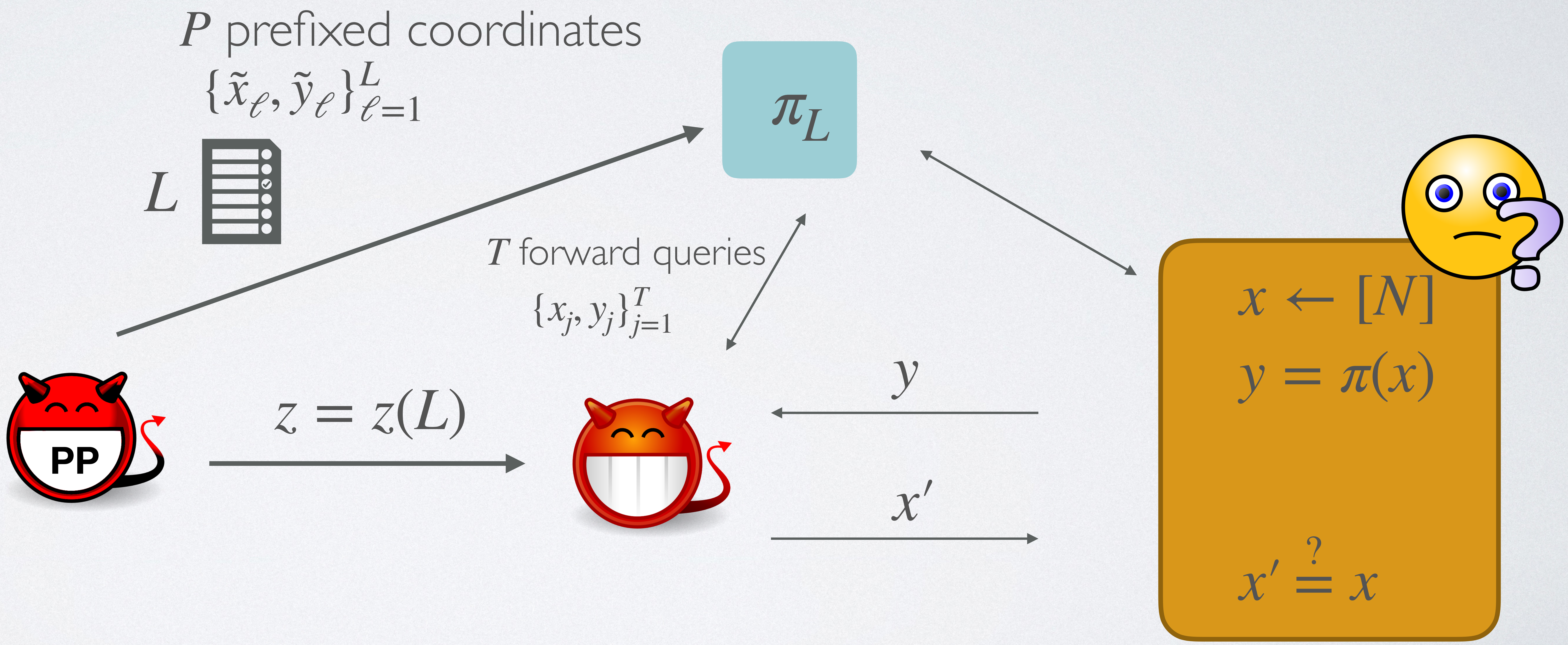
Prefixed random permutation $\pi_L : [N] \to [N]$



OWP in BF Model

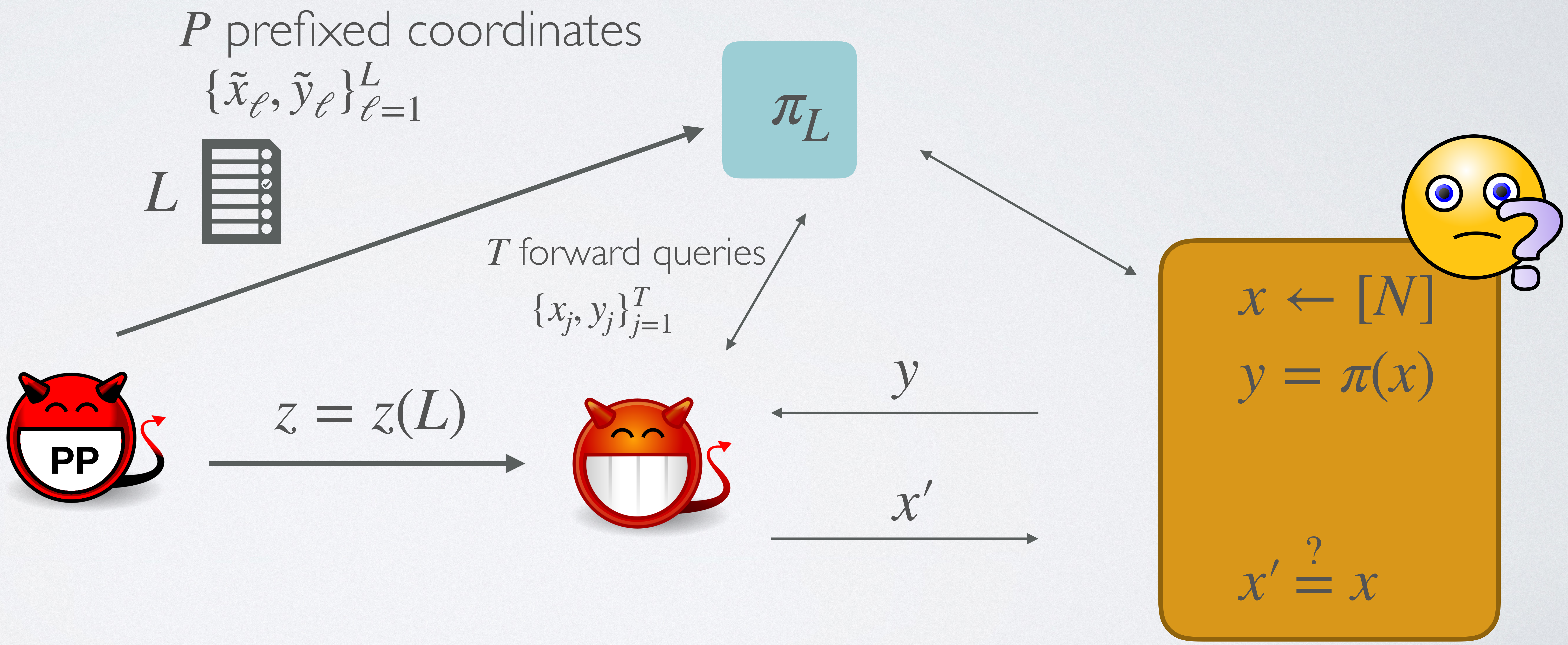
Prefixed random permutation $\pi_L : [N] \to [N]$

Event BAD:



OWP in BF Model

Prefixed random permutation $\pi_L : [N] \to [N]$

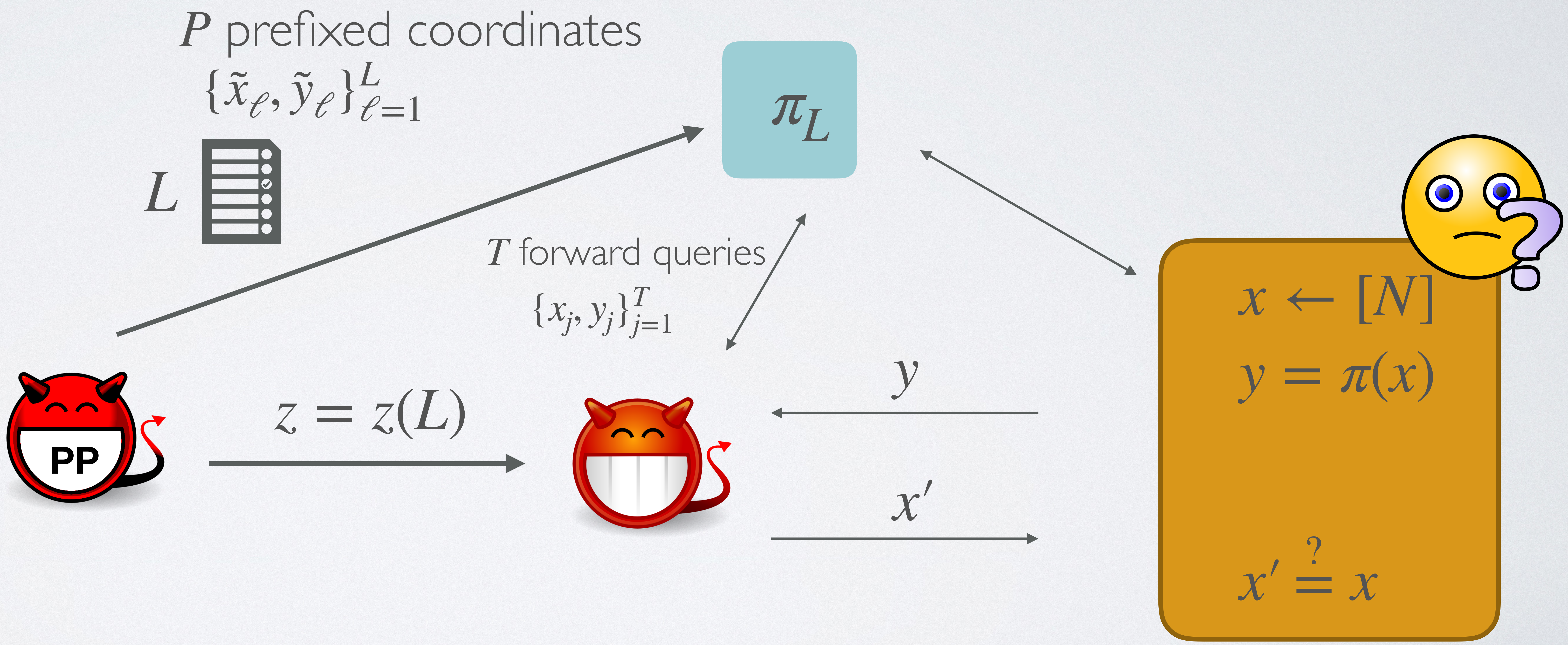


Event **BAD**:

$\exists j : \quad x_j = x$

OWP in BF Model

Prefixed random permutation $\pi_L : [N] \to [N]$



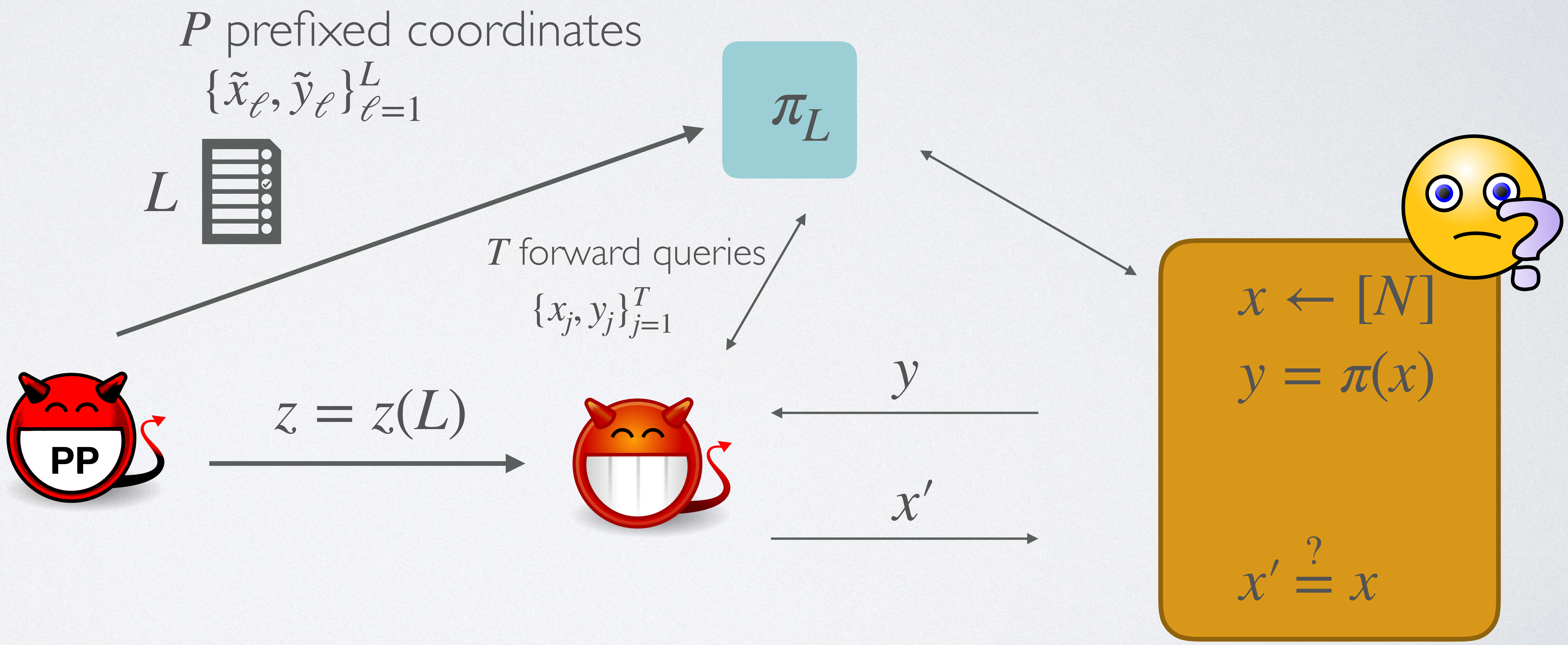
Event **BAD**:

$\exists j : \quad x_j = x$

$\exists \ell : \quad \tilde{x}_\ell = x$

OWP in BF Model

Prefixed random permutation $\pi_L : [N] \to [N]$



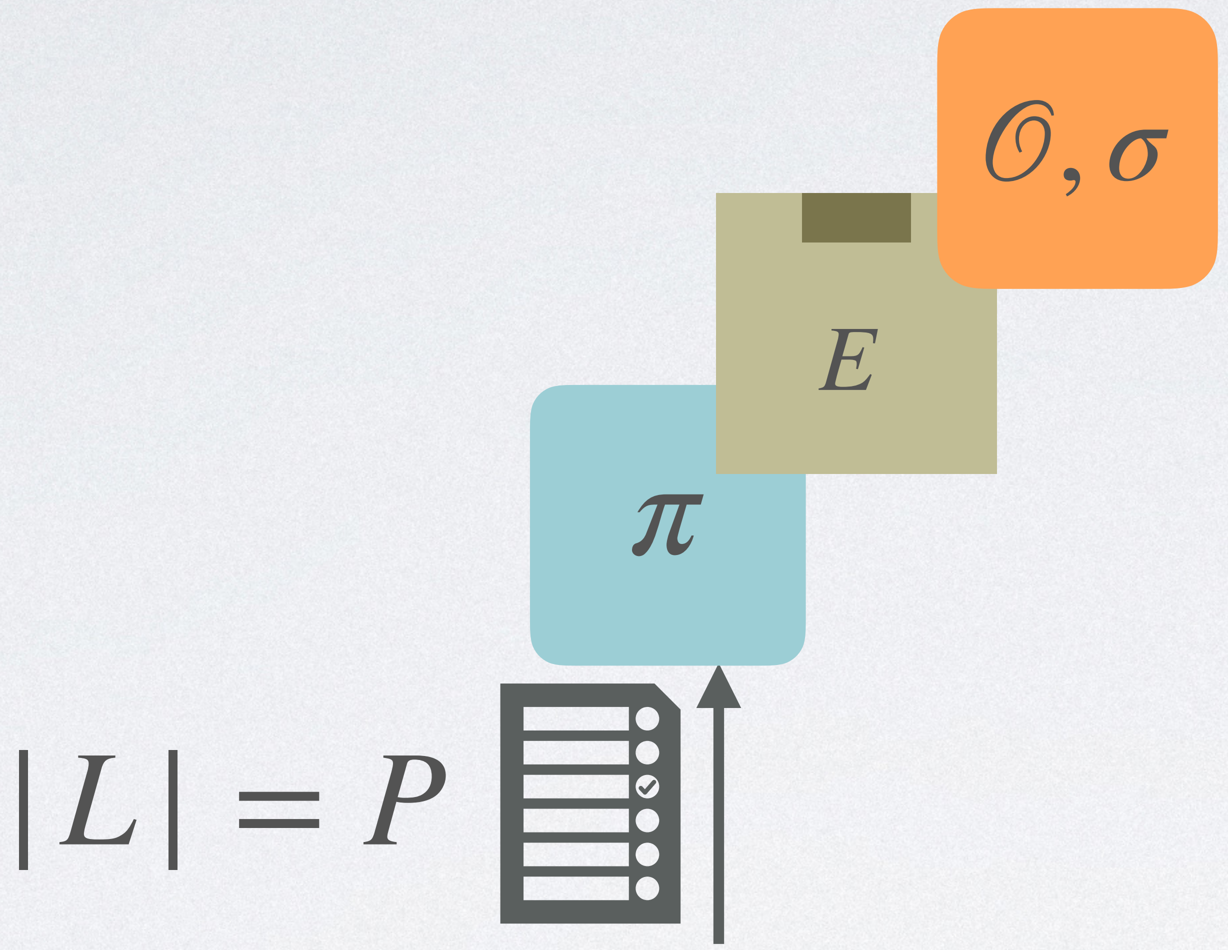
Event BAD:

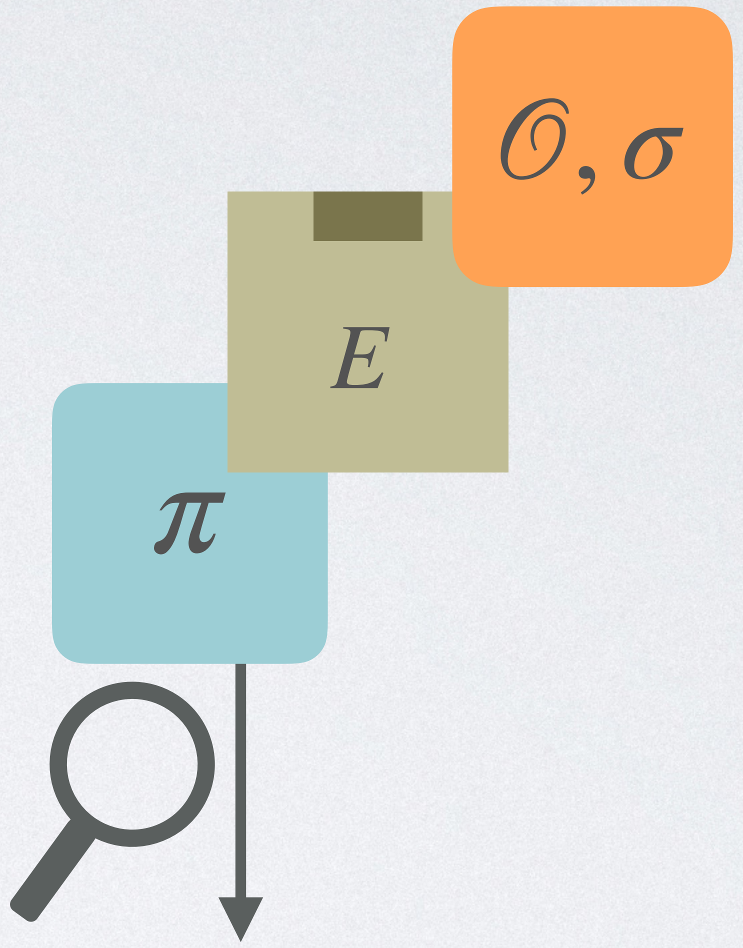
$\exists j : \quad x_j = x$

$\exists \ell : \quad \tilde{x}_\ell = x$

$$\text{P[BAD]} \leq \frac{T}{N} + \frac{P}{N}$$

Bit-Fixing to Auxiliary Input





Theorem: (S, T, ε)-secure $\Longrightarrow$ (S, T, ε')-secure

where $\varepsilon' \leq 2\varepsilon$
and $P \approx ST$

For **unpredictability** applications

OWP in BF Model

Bound in BF-GGM: $\frac{T}{N} + \frac{P}{N} \longrightarrow P \approx ST$

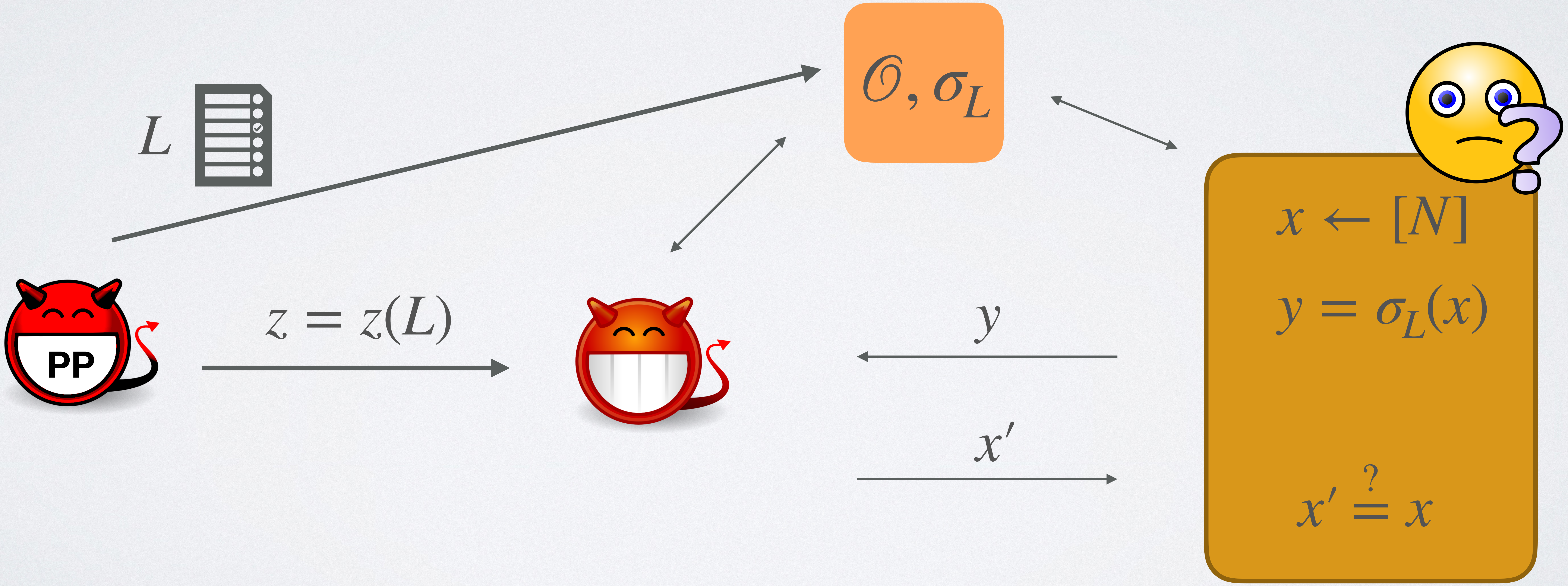
Bound in AI-GGM: $\frac{T}{N} + \frac{ST}{N}$

Bound in AI-GGM (compression proof): $\frac{ST}{N}$

De, Trevisan, Tulsiani '10

Discrete Logarithms in BF Model

Random injection $\sigma : [N] \to [M]$



Discrete Logarithms in BF Model

Prefixed random injection $\sigma_L : [N] \to [M]$

P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$

By making queries to $\mathcal{O}$:

$\mathcal{A}$ "generates" degree-1 polynomials in X

Event BAD: two polynomials collide at $X = x$

Discrete Logarithms in BF Model

Prefixed random injection $\sigma_L : [N] \to [M]$

P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$

By making queries to $\mathcal{O}$:

$\mathcal{A}$ "generates" degree-1 polynomials in X

Event BAD: two polynomials collide at $X = x$

$\exists i, j$: some polynomial evaluates to $\tilde{x}_j$ at $X = x$

Discrete Logarithms in BF Model

Prefixed random injection $\sigma_L : [N] \to [M]$

P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$

By making queries to $\mathcal{O}$:

$\mathcal{A}$ "generates" degree-1 polynomials in X

Event BAD: two polynomials collide at $X = x$

$\exists i, j$: some polynomial evaluates to $\tilde{x}_j$ at $X = x$

$$\text{P[BAD]} \leq \frac{T^2}{N} + \frac{PT}{N}$$

Discrete Logarithms in BF Model

Bound in BF-GGM: $\frac{T^2}{N} + \frac{PT}{N} \longrightarrow P \approx ST$

Bound in AI-GGM: $\frac{T^2}{N} + \frac{ST^2}{N}$

Discrete Logarithms in BF Model

Bound in BF-GGM: $\frac{T^2}{N} + \frac{PT}{N} \longrightarrow P \approx ST$

Bound in AI-GGM: $\frac{T^2}{N} + \frac{ST^2}{N}$

Bound in AI-GGM (compression proof): $\frac{ST^2}{N}$

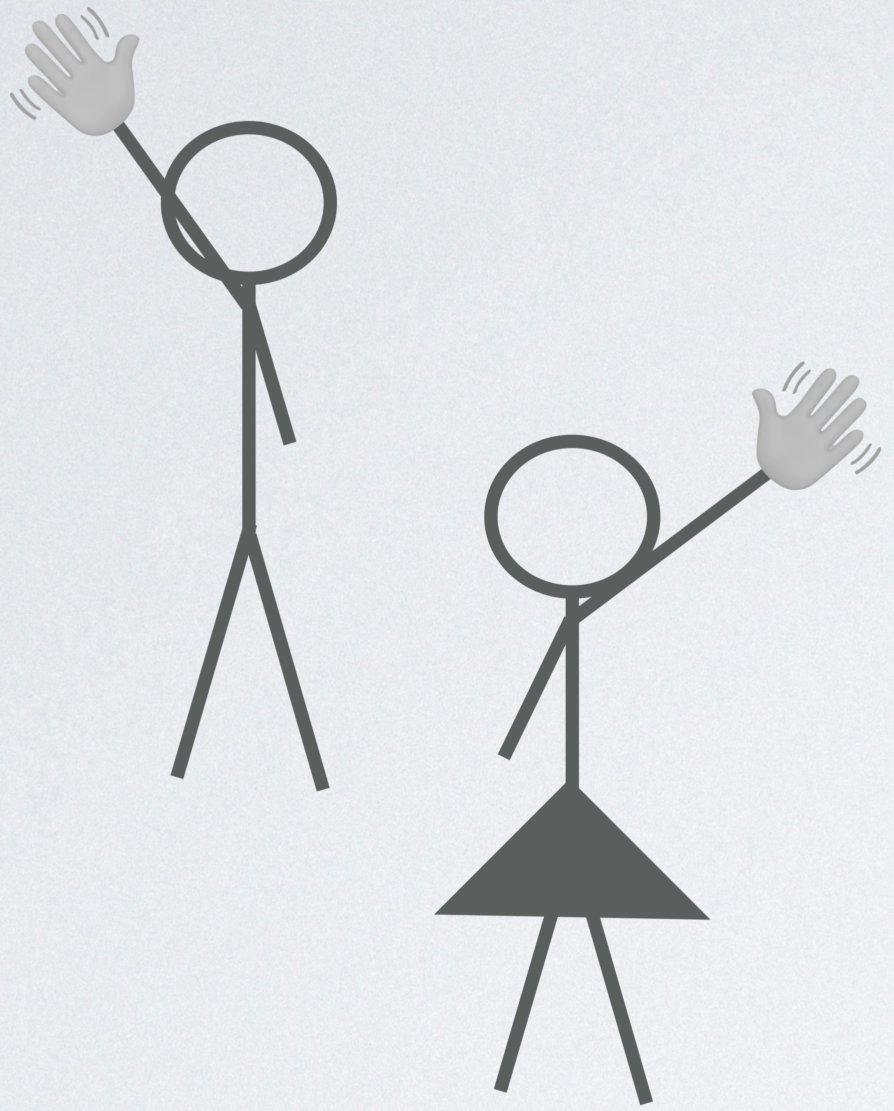
Corrigan-Gibbs, Kogan '18

Summary of Bounds

- **Basic:** OWP, Even-Mansour, ideal cipher as block cipher, Davies-Meyer as a PRF, CRHF based on Davies-Meyer
- **Symmetric:** Merkle-Damgard with Davies-Meyer and sponges (as CRHF, PRF, MAC), …
- **Generic group model:** DL, CDH, DDH, OM-DL, KEA, …
- **Computational:** Full-domain permutation encryption

Some Future Work

- Close gaps:
  - EM: bound: $\sqrt{\frac{ST^2}{N}}$ attack: $\frac{ST^2}{N}$
  - DDH: bound: $\sqrt{\frac{ST^2}{N}}$ attack: $\frac{ST^2}{N}$, $\sqrt{\frac{S}{N}}$
- Tight bounds for other primitives (e.g., KAC)



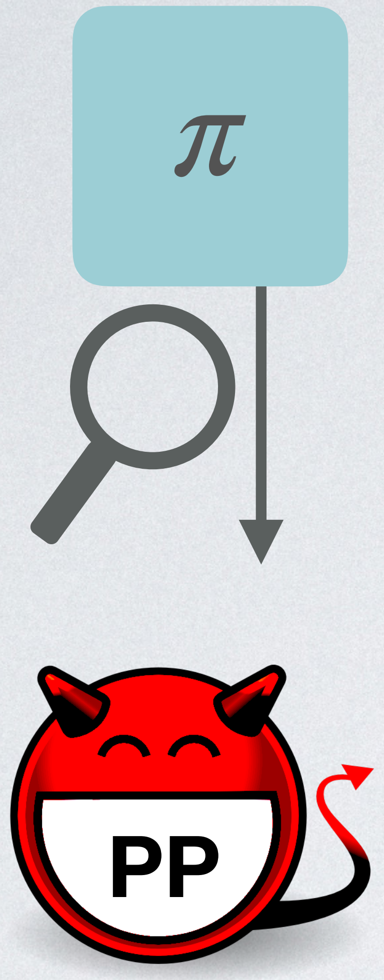
Thank you!

eprint.iacr.org/2018/226

Proof of Presampling

Göös, Lovett, Meka '16

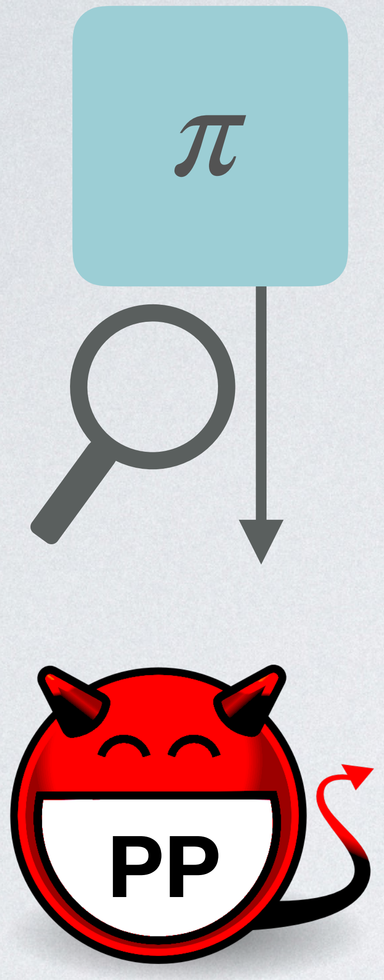
C, Dodis, Guo, Steinberger '18



Proof of Presampling

Göös, Lovett, Meka '16

C, Dodis, Guo, Steinberger '18

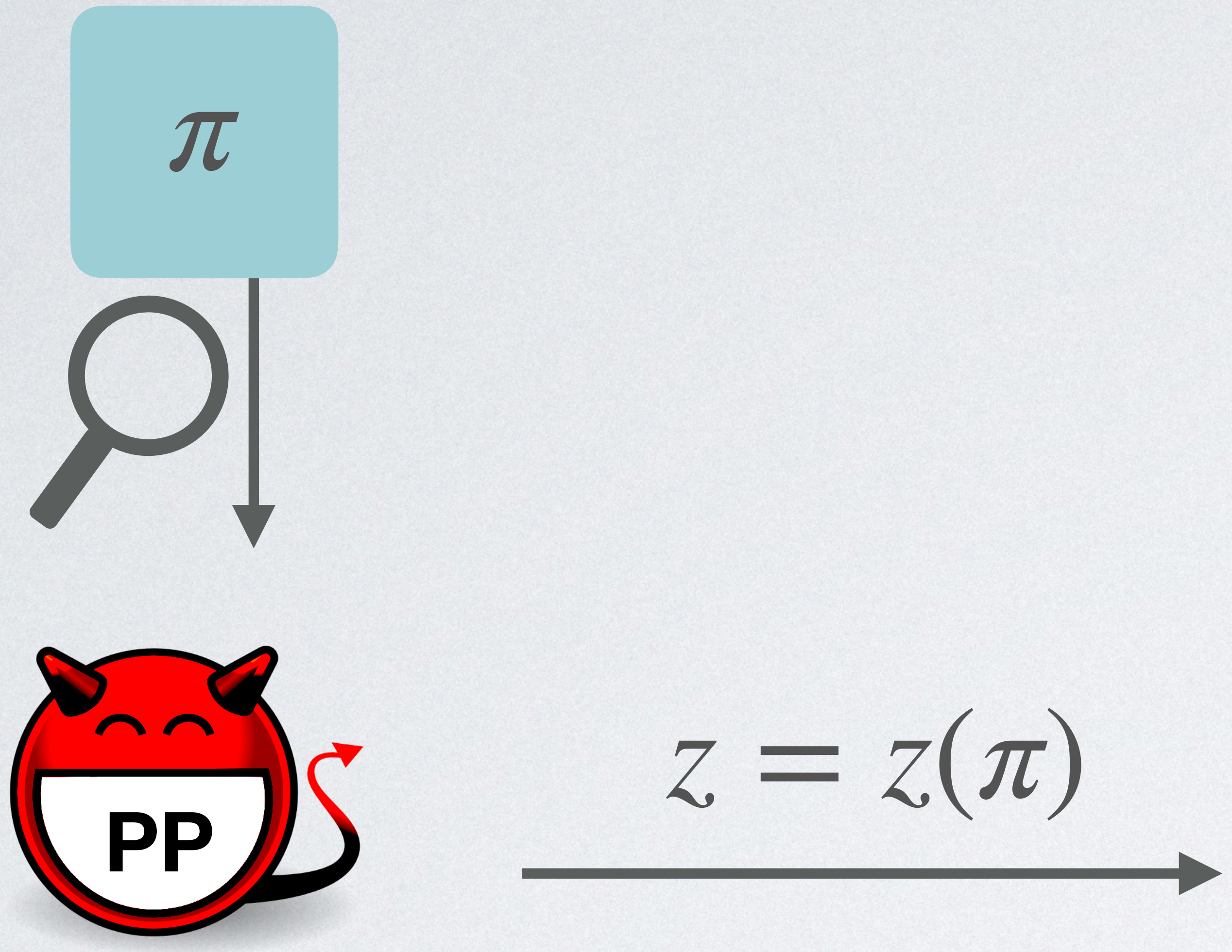


Before leakage: $\mathrm{H}_\infty(\pi) = \log N!$

Proof of Presampling

Göös, Lovett, Meka '16

C, Dodis, Guo, Steinberger '18



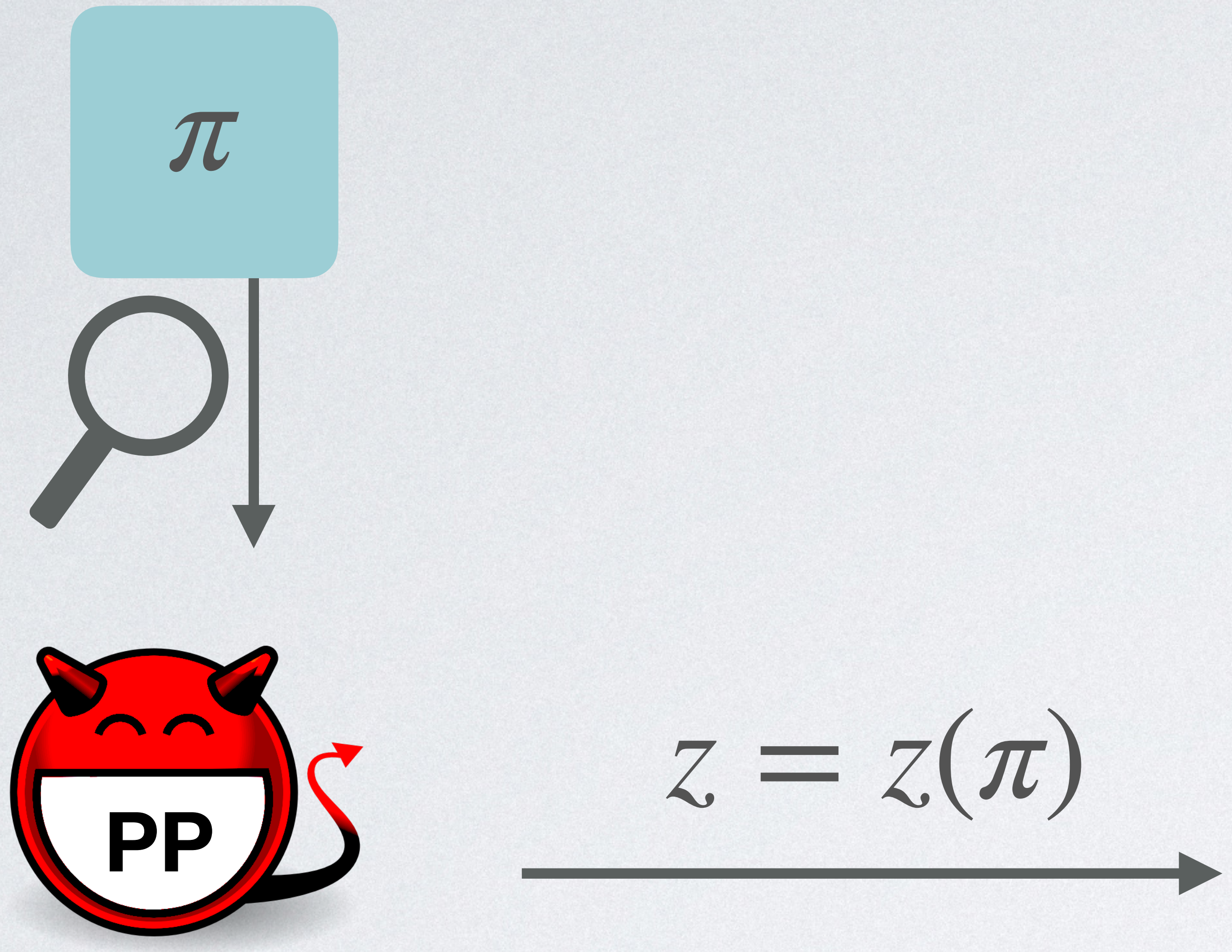
Before leakage: $\mathrm{H}_\infty(\pi) = \log N!$

After leakage: $\mathrm{H}_\infty(\pi \mid z) = \log N! - S$

Proof of Presampling

Göös, Lovett, Meka ‘16

C, Dodis, Guo, Steinberger ‘18



Before leakage: $\mathrm{H}_\infty(\pi) = \log N!$

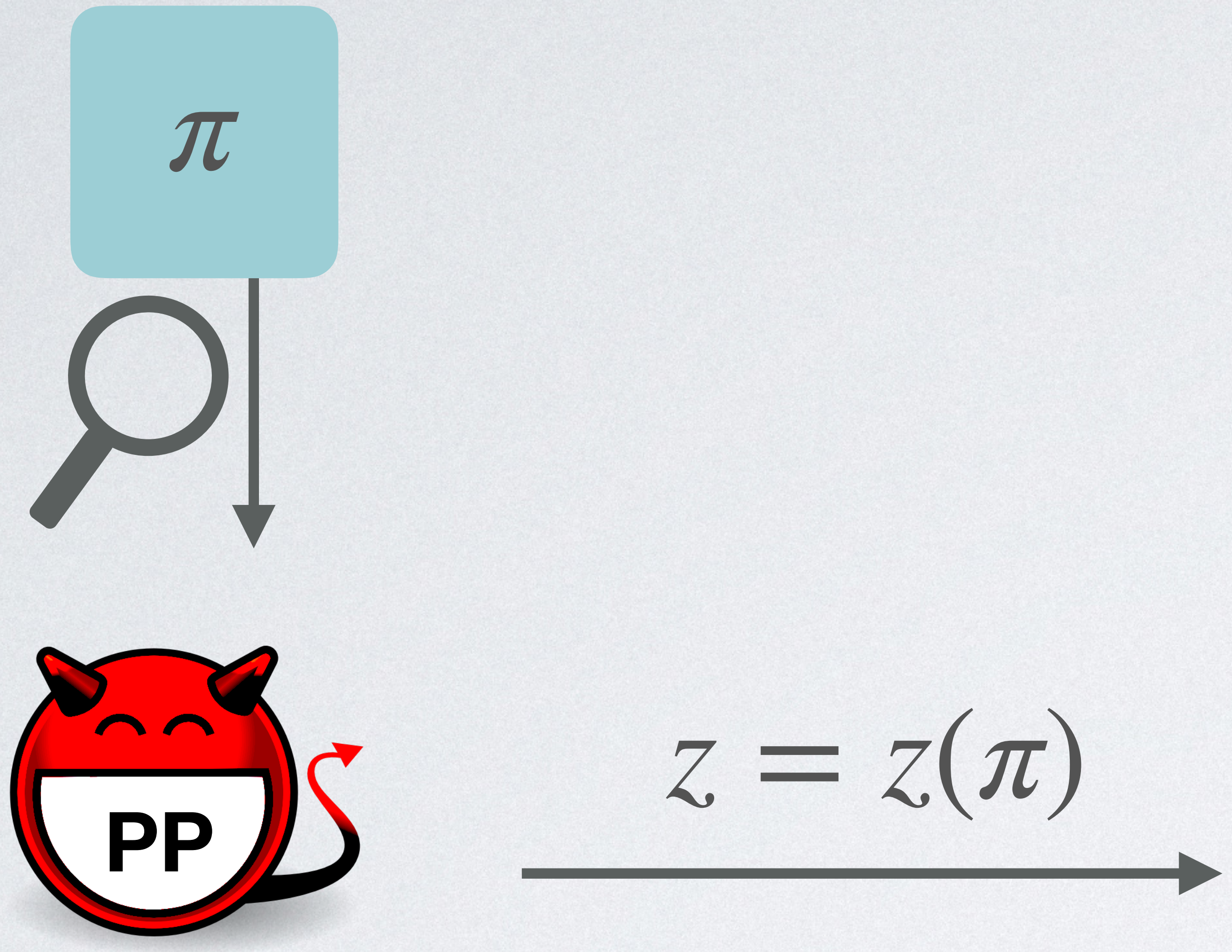
After leakage: $\mathrm{H}_\infty(\pi \,|\, z) = \log N! - S$

1. Decompose $\pi' := \pi \,|\, z$ into dense sources

Göös, Lovett, Meka ‘16

C, Dodis, Guo, Steinberger ‘18

Proof of Presampling



Before leakage: $H_\infty(\pi) = \log N!$

After leakage: $H_\infty(\pi \,|\, z) = \log N! - S$

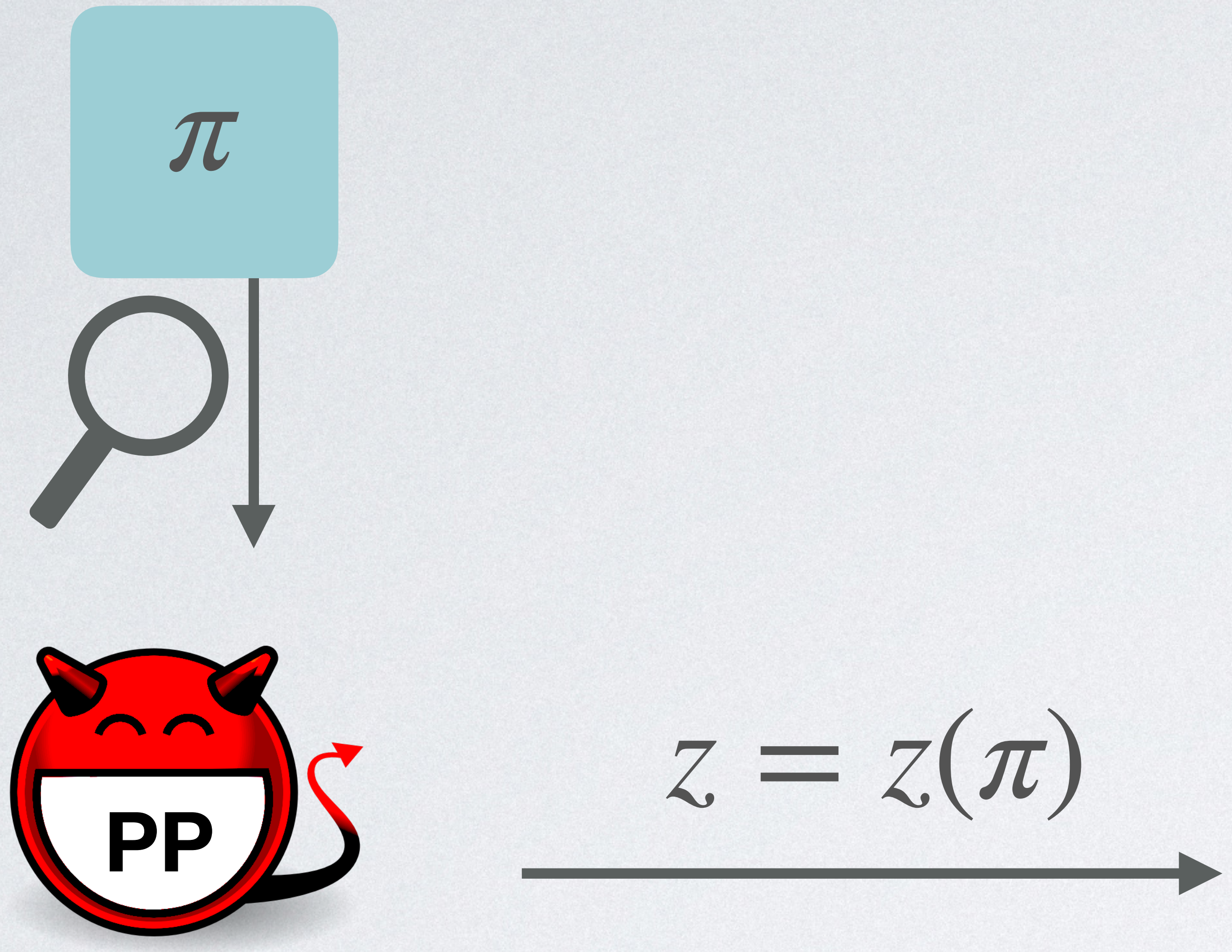
1. Decompose $\pi' := \pi \,|\, z$ into dense sources

   (a) Fixed on P coordinates $L \subseteq [N]$

Göös, Lovett, Meka '16

C, Dodis, Guo, Steinberger '18

Proof of Presampling



Before leakage: $\mathrm{H}_\infty(\pi) = \log N!$

After leakage: $\mathrm{H}_\infty(\pi \,|\, z) = \log N! - S$

1. Decompose $\pi' := \pi \,|\, z$ into dense sources

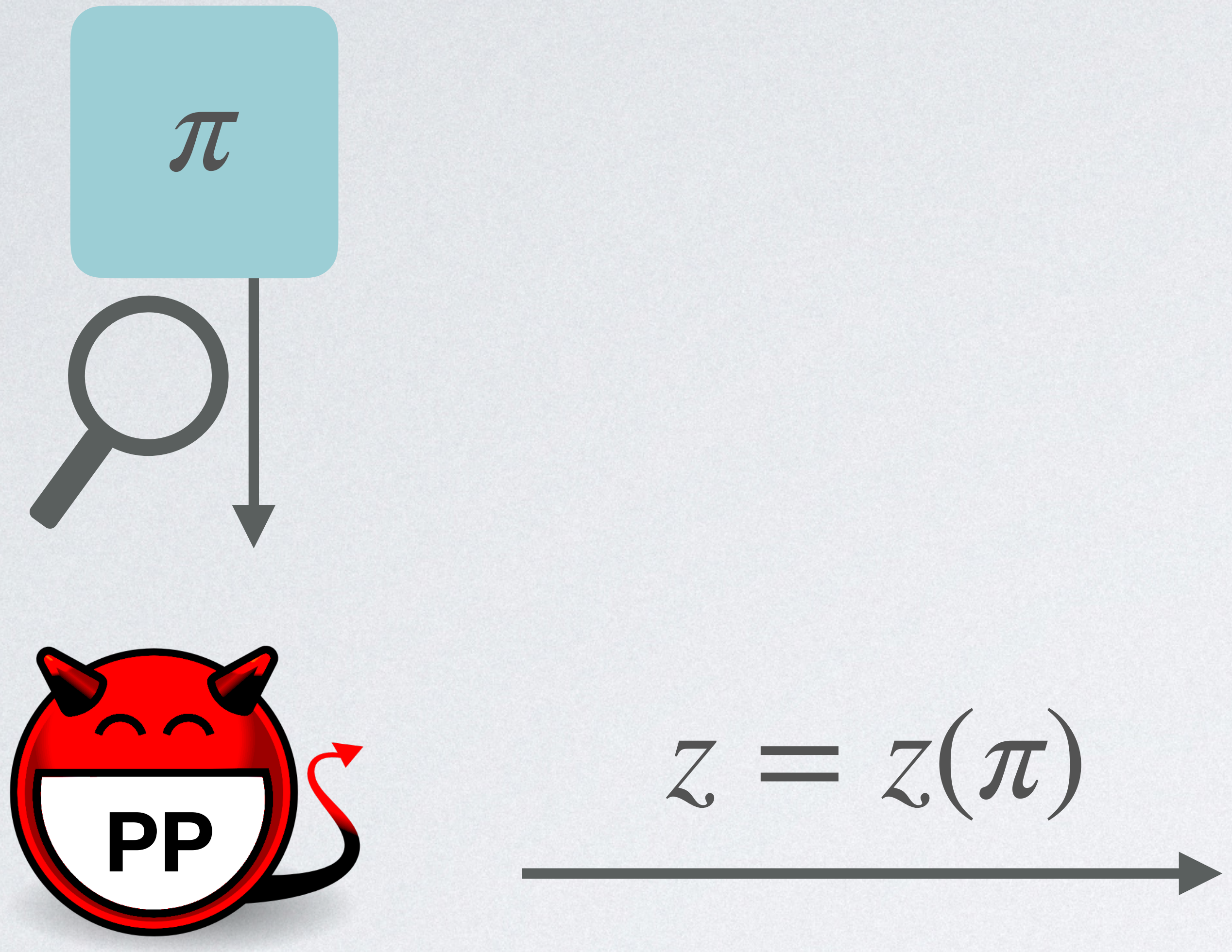
(a) Fixed on P coordinates $L \subseteq [N]$

(b) $\forall\ Q \subseteq [N] \backslash L$:

Proof of Presampling

Göös, Lovett, Meka '16

C, Dodis, Guo, Steinberger '18



Before leakage: $\mathrm{H}_\infty(\pi) = \log N!$

After leakage: $\mathrm{H}_\infty(\pi \mid z) = \log N! - S$

1. Decompose $\pi' := \pi \mid z$ into dense sources

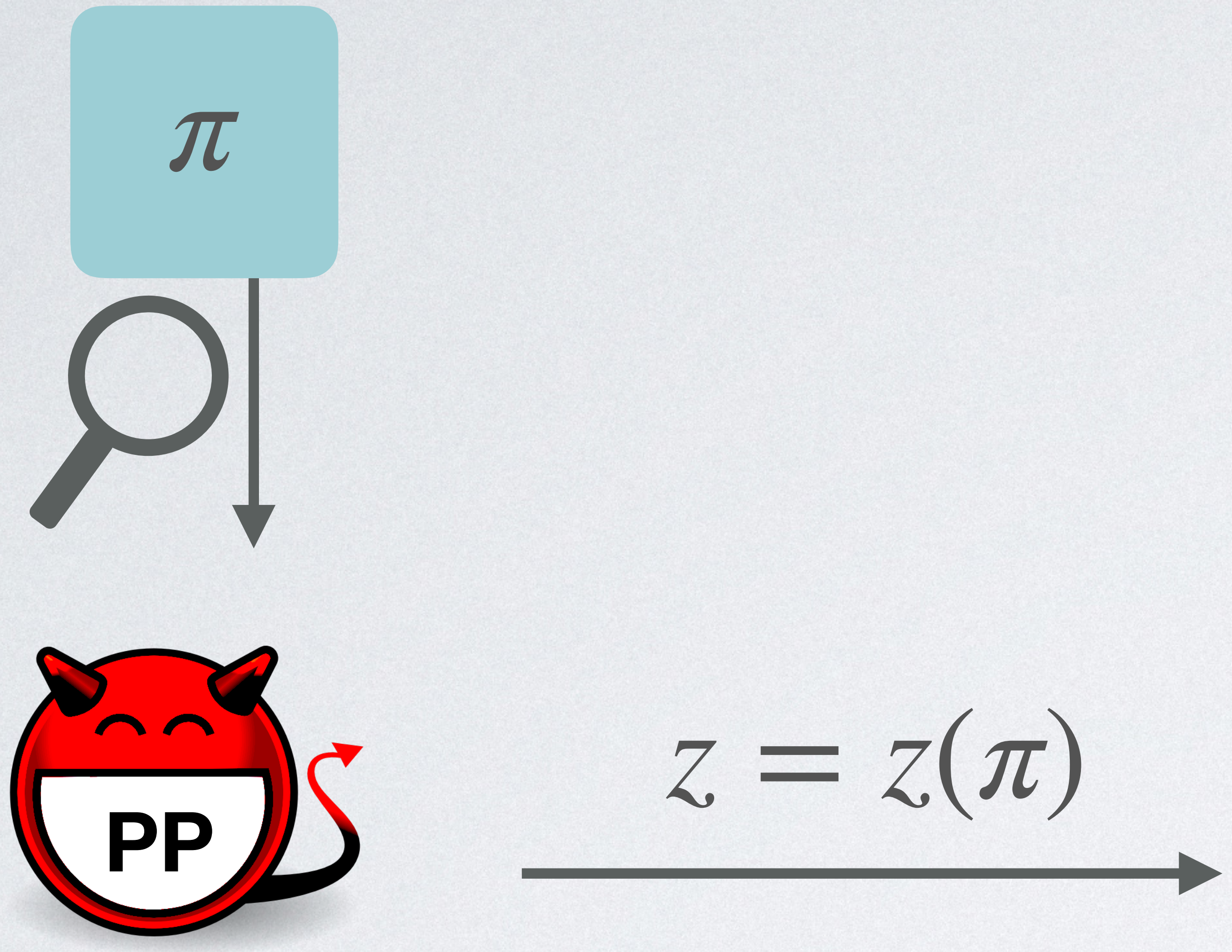
   (a) Fixed on P coordinates $L \subseteq [N]$

   (b) $\forall\ Q \subseteq [N] \setminus L : \quad \mathrm{H}_\infty(\pi'_Q)$

Proof of Presampling

Göös, Lovett, Meka '16

C, Dodis, Guo, Steinberger '18



Before leakage: $\mathrm{H}_\infty(\pi) = \log N!$

After leakage: $\mathrm{H}_\infty(\pi \,|\, z) = \log N! - S$

1. Decompose $\pi' := \pi \,|\, z$ into dense sources

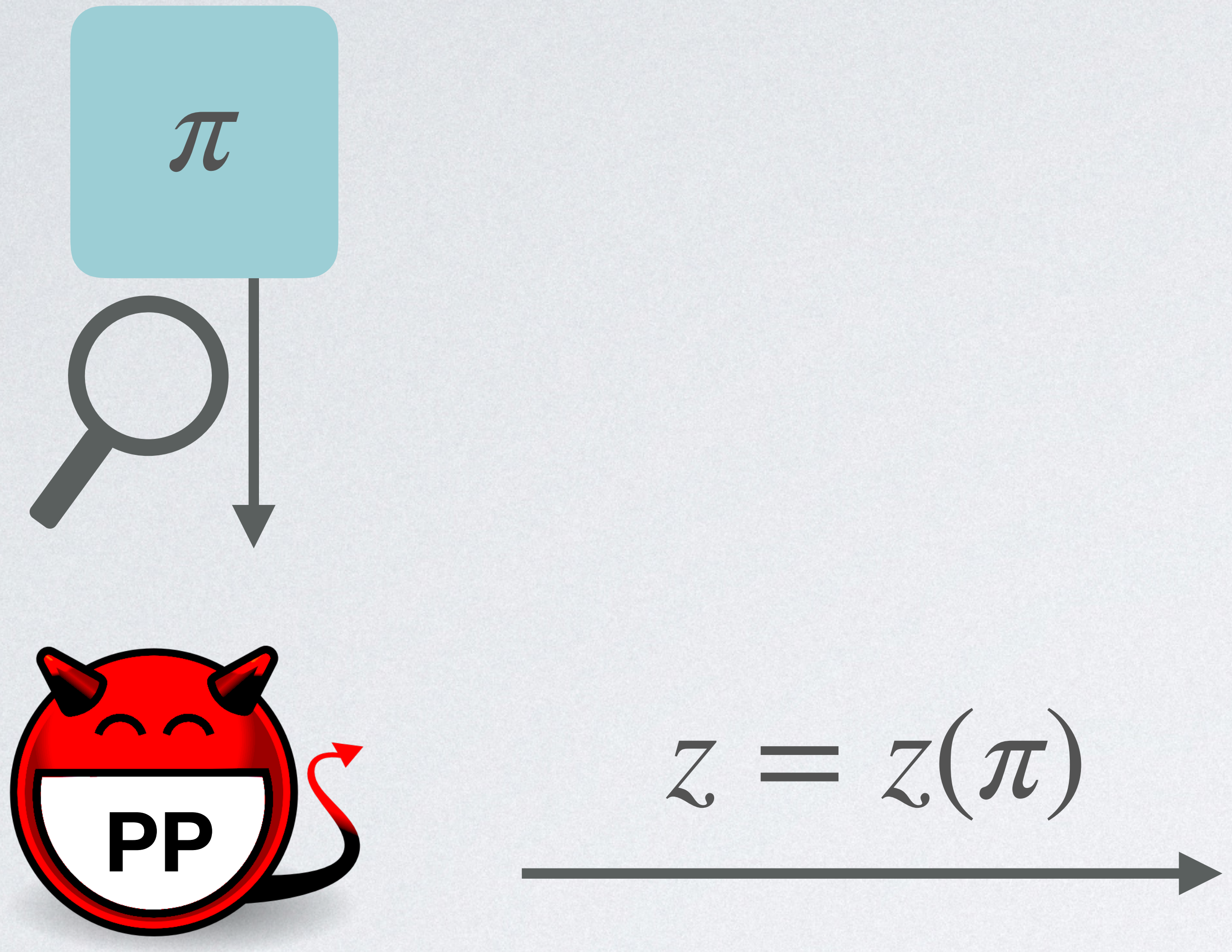
   (a) Fixed on P coordinates $L \subseteq [N]$

   (b) $\forall\ Q \subseteq [N] \backslash L : \quad \mathrm{H}_\infty(\pi'_Q) \ \geq\ (1-\delta) \cdot$

Proof of Presampling

Göös, Lovett, Meka '16

C, Dodis, Guo, Steinberger '18



Before leakage: $\mathrm{H}_\infty(\pi) = \log N!$

After leakage: $\mathrm{H}_\infty(\pi \,|\, z) = \log N! - S$

1. Decompose $\pi' := \pi \,|\, z$ into dense sources

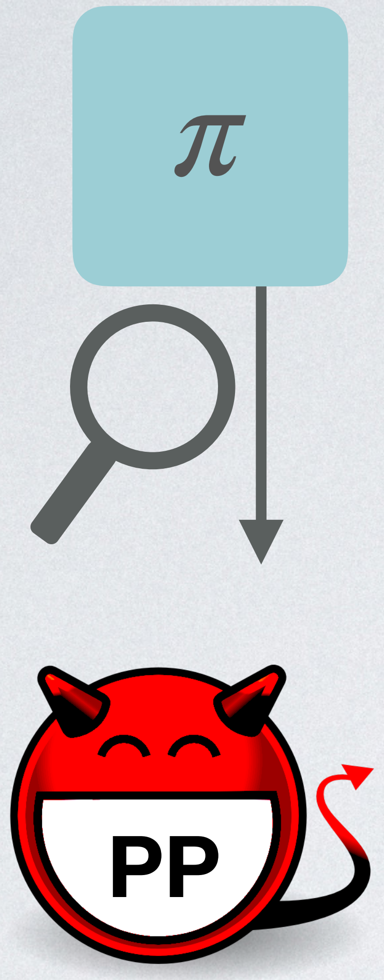
(a) Fixed on P coordinates $L \subseteq [N]$

(b) $\forall\, Q \subseteq [N] \backslash L: \quad \mathrm{H}_\infty(\pi'_Q) \;\geq\; (1-\delta) \cdot \log \dfrac{(N-P)!}{(N-P-|Q|)!}$

Göös, Lovett, Meka '16

C, Dodis, Guo, Steinberger '18

Proof of Presampling



$z = z(\pi)$

Before leakage: $H_\infty(\pi) = \log N!$

After leakage: $H_\infty(\pi \mid z) = \log N! - S$

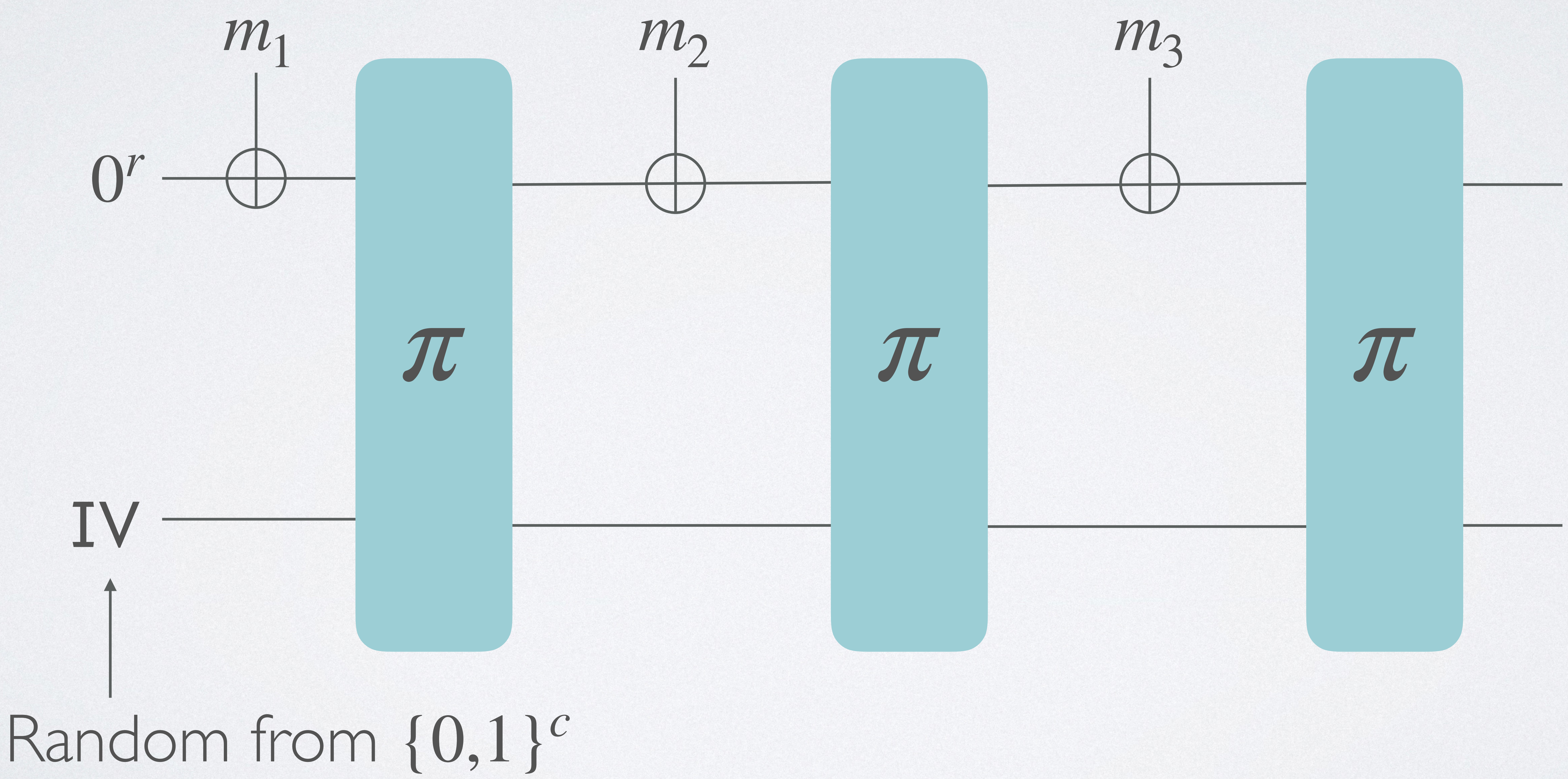
1. Decompose $\pi' := \pi \mid z$ into dense sources

   (a) Fixed on P coordinates $L \subseteq [N]$

   (b) $\forall\, Q \subseteq [N] \backslash L: \quad H_\infty(\pi'_Q) \;\geq\; (1-\delta) \cdot \log \dfrac{(N-P)!}{(N-P-|Q|)!}$

2. Show dense is indistinguishable from uniform

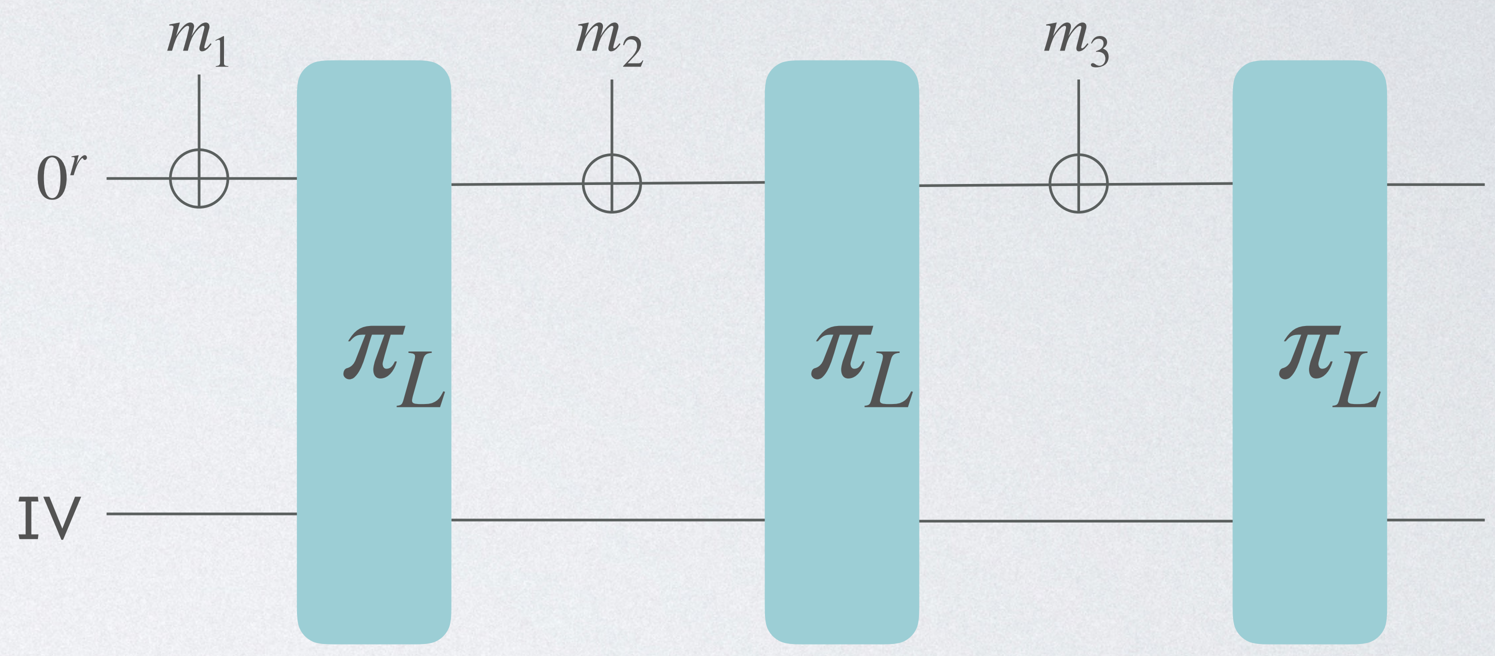
SHA-3 Collision Resistance in the AI-RPM



SHA-3 Collision Resistance in the BF-RPM

P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$



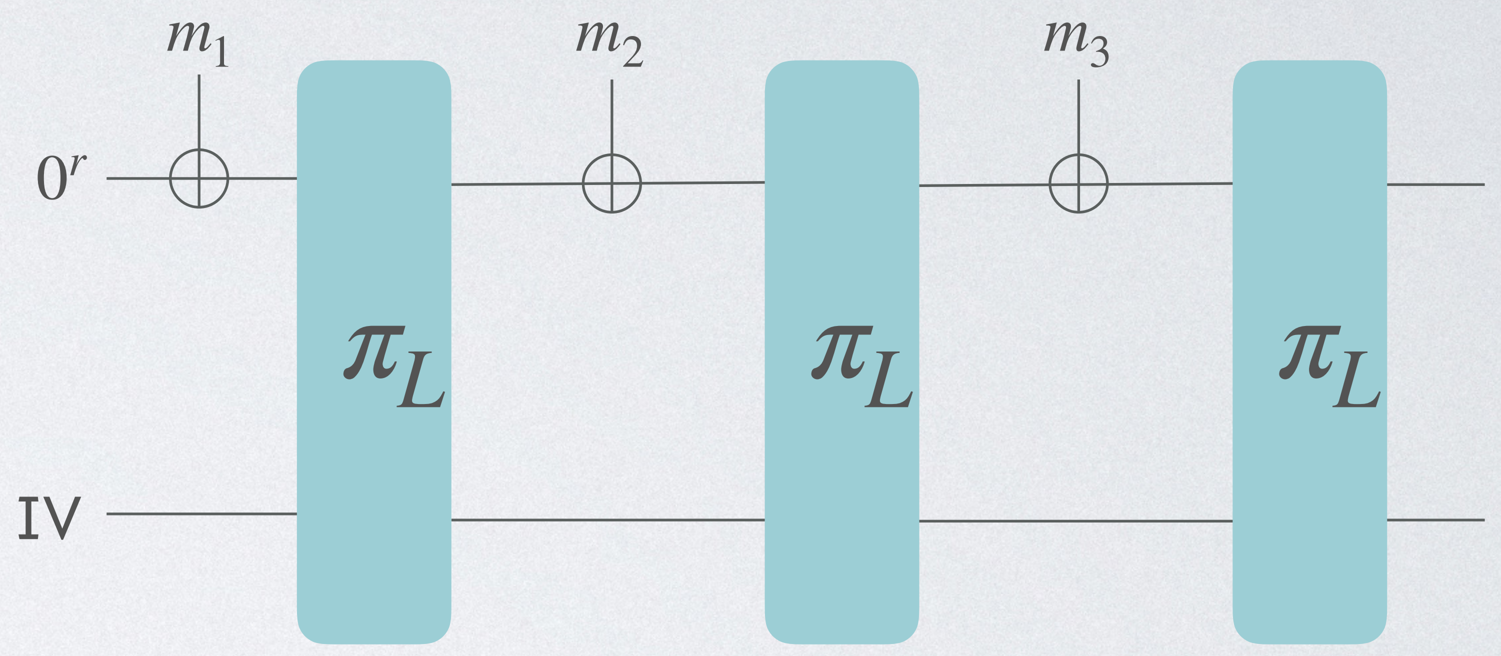
SHA-3 Collision Resistance in the BF-RPM

T queries

$\{x_j, y_j\}_{j=1}^{T}$

P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$



SHA-3 Collision Resistance in the BF-RPM

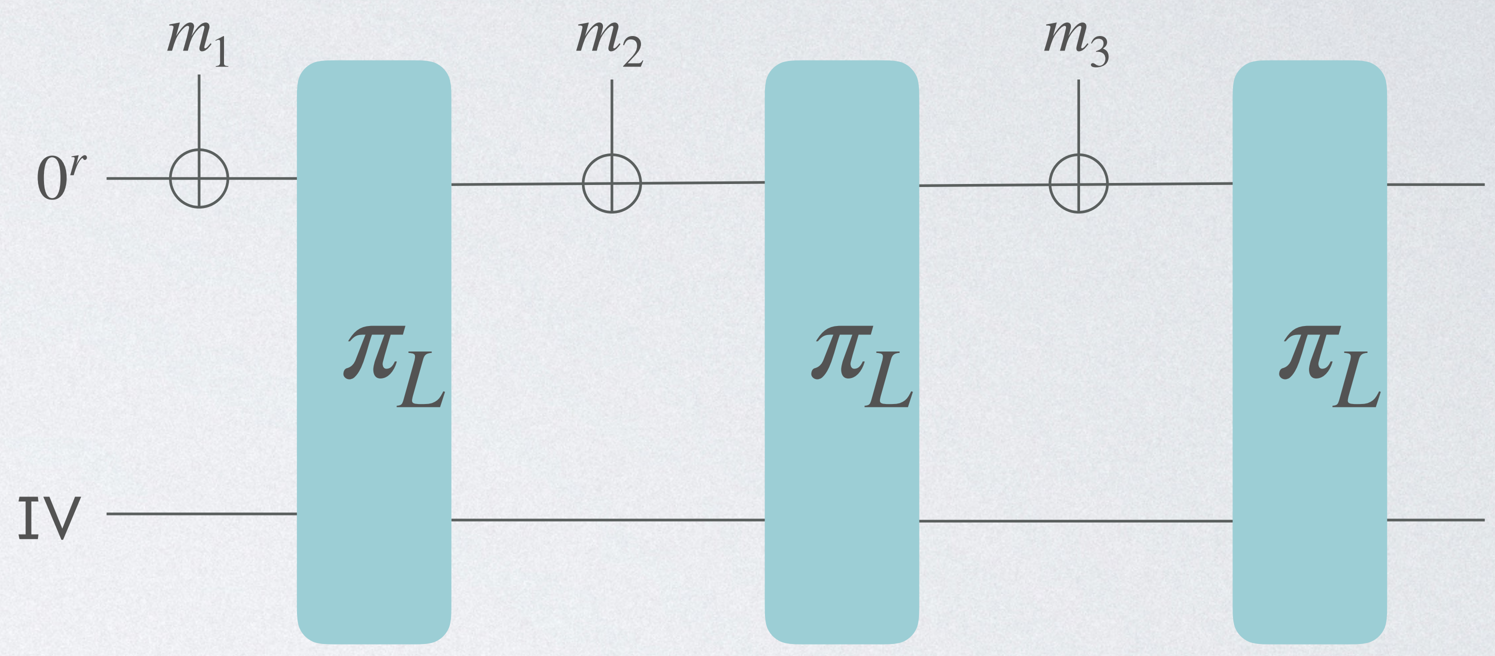
T queries

$\{x_j, y_j\}_{j=1}^{T}$

P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$

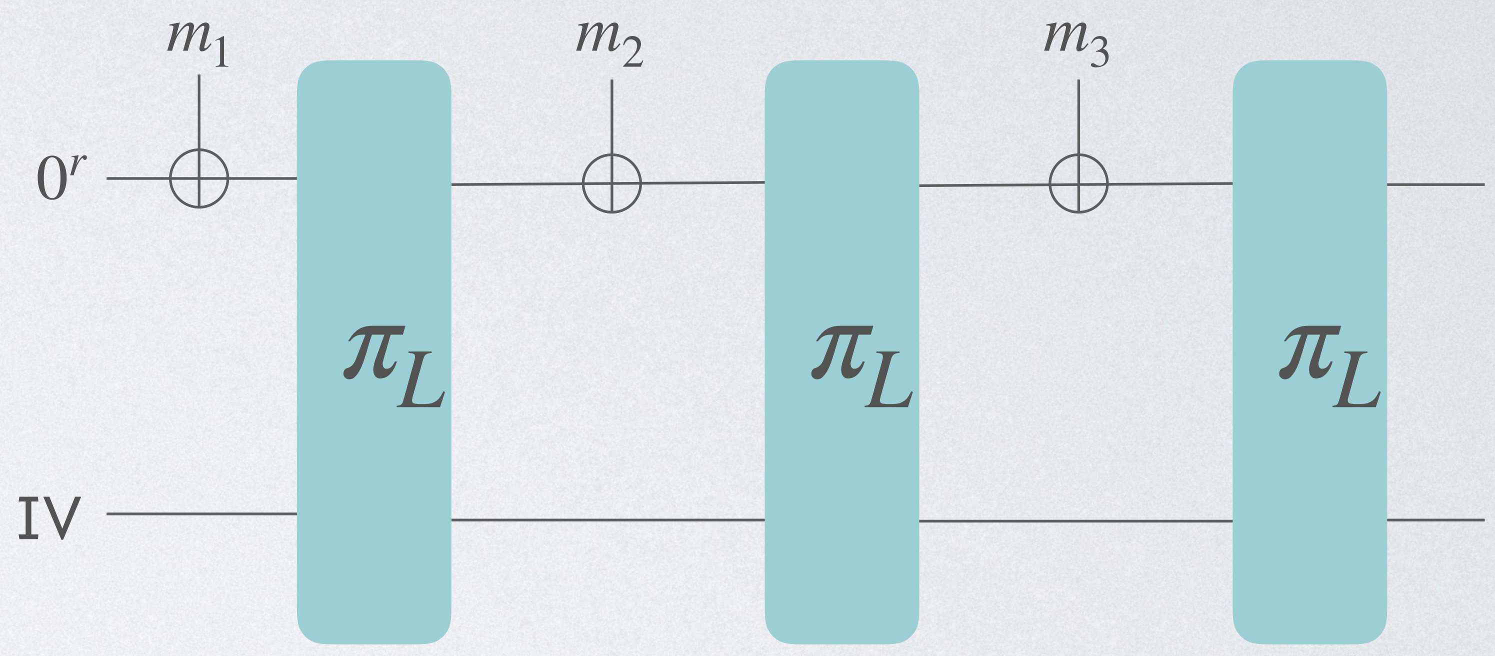
Queries: graph starting at $(0^r, IV)$



SHA-3 Collision Resistance in the BF-RPM

T queries $\{x_j, y_j\}_{j=1}^{T}$

P prefixed coordinates $\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$



Queries: graph starting at $(0^r, IV)$

Event **BAD**: capacity part not fresh

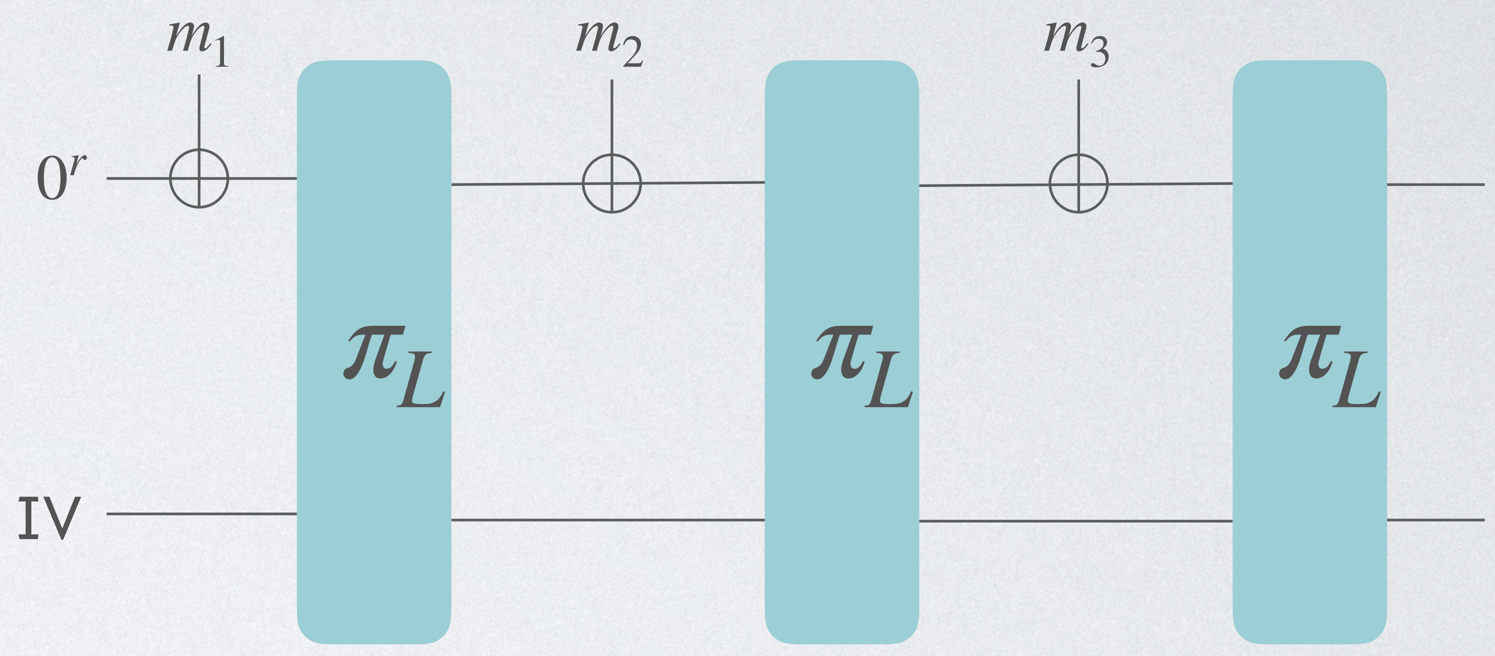
SHA-3 Collision Resistance in the BF-RPM

T queries

$\{x_j, y_j\}_{j=1}^T$

P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^L$



Queries: graph starting at $(0^r, IV)$

Event BAD: capacity part not fresh

$$\text{P[BAD]} \leq \frac{T^2}{2^c} + \frac{TP}{2^c}$$

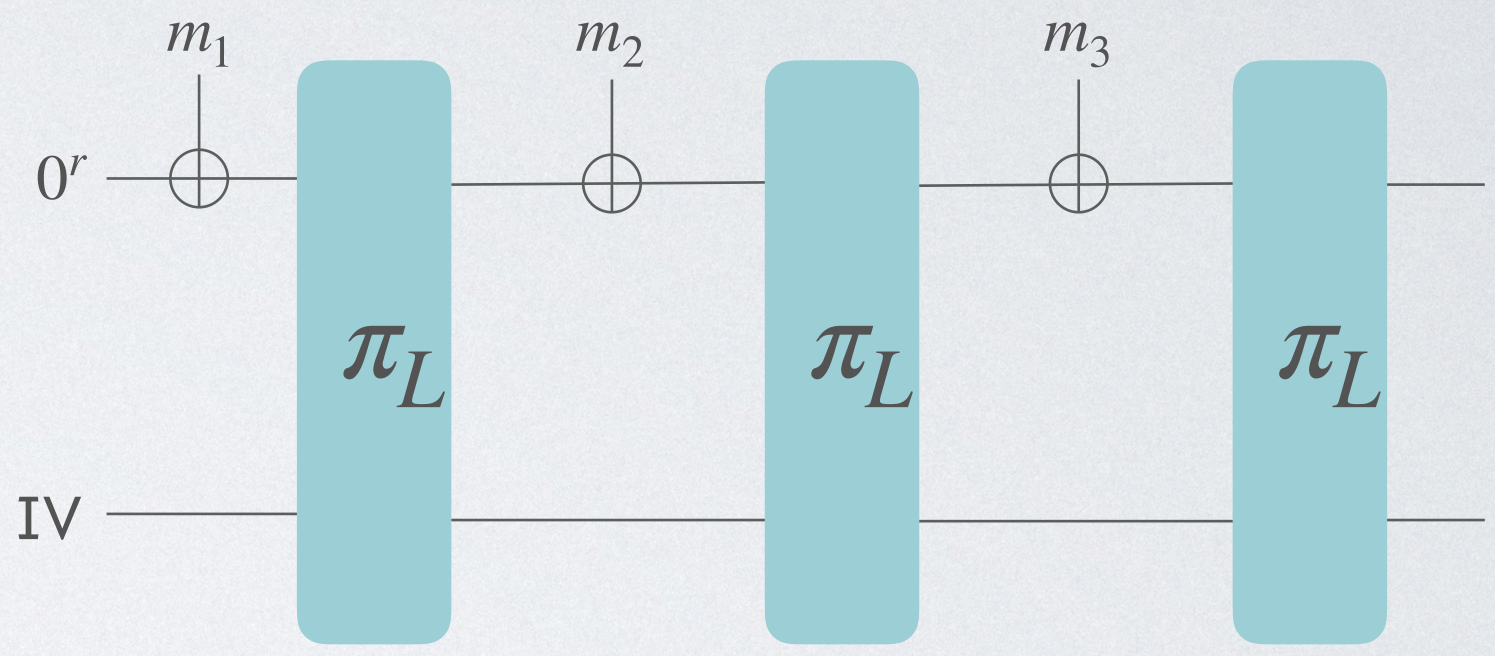
SHA-3 Collision Resistance in the BF-RPM

T queries

$\{x_j, y_j\}_{j=1}^{T}$

P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$



Queries: graph starting at $(0^r, IV)$

Event **BAD**: capacity part not fresh

$$\text{P[BAD]} \leq \frac{T^2}{2^c} + \frac{TP}{2^c}$$

$$\text{P[COLL | BAD]} \leq \frac{T^2}{2^r}$$

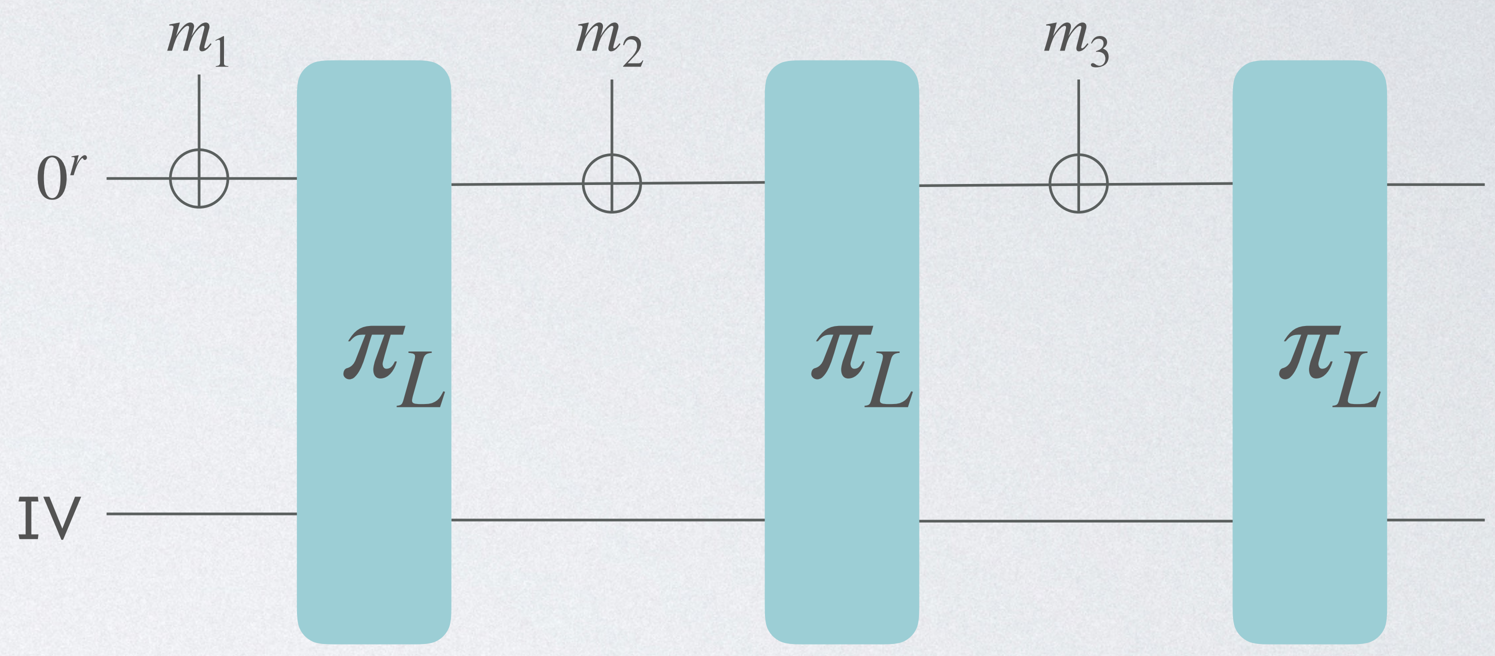
SHA-3 Collision Resistance in the BF-RPM

T queries

$\{x_j, y_j\}_{j=1}^{T}$

P prefixed coordinates

$\{\tilde{x}_\ell, \tilde{y}_\ell\}_{\ell=1}^{L}$



Queries: graph starting at $(0^r, IV)$

Event BAD: capacity part not fresh

$$\text{P[BAD]} \leq \frac{T^2}{2^c} + \frac{TP}{2^c}$$

$$\text{P[COLL | BAD]} \leq \frac{T^2}{2^r}$$

$$P \approx ST \longrightarrow$$

AI-ROM: $\dfrac{T^2}{2^r} + \dfrac{ST^2}{2^c}$