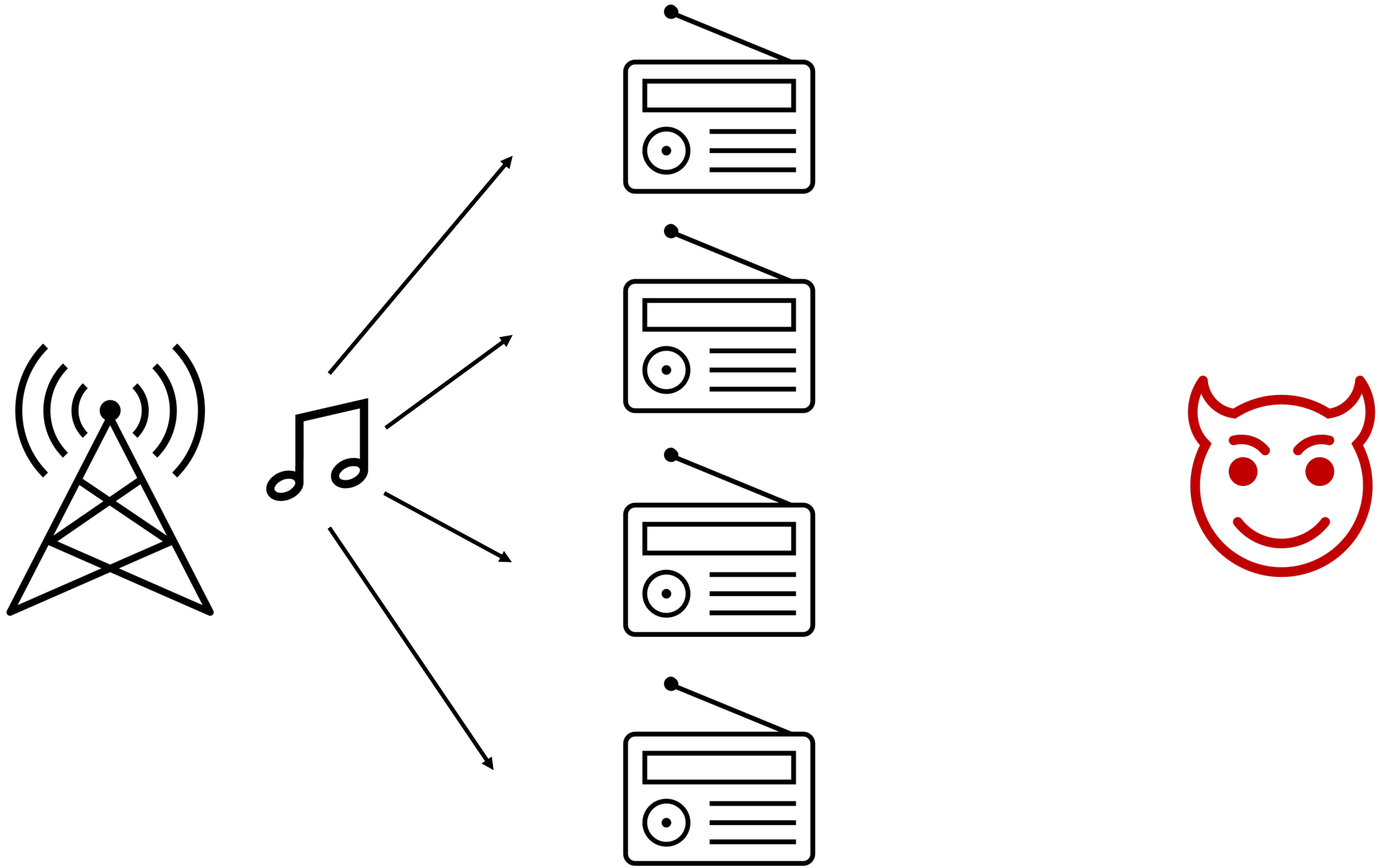
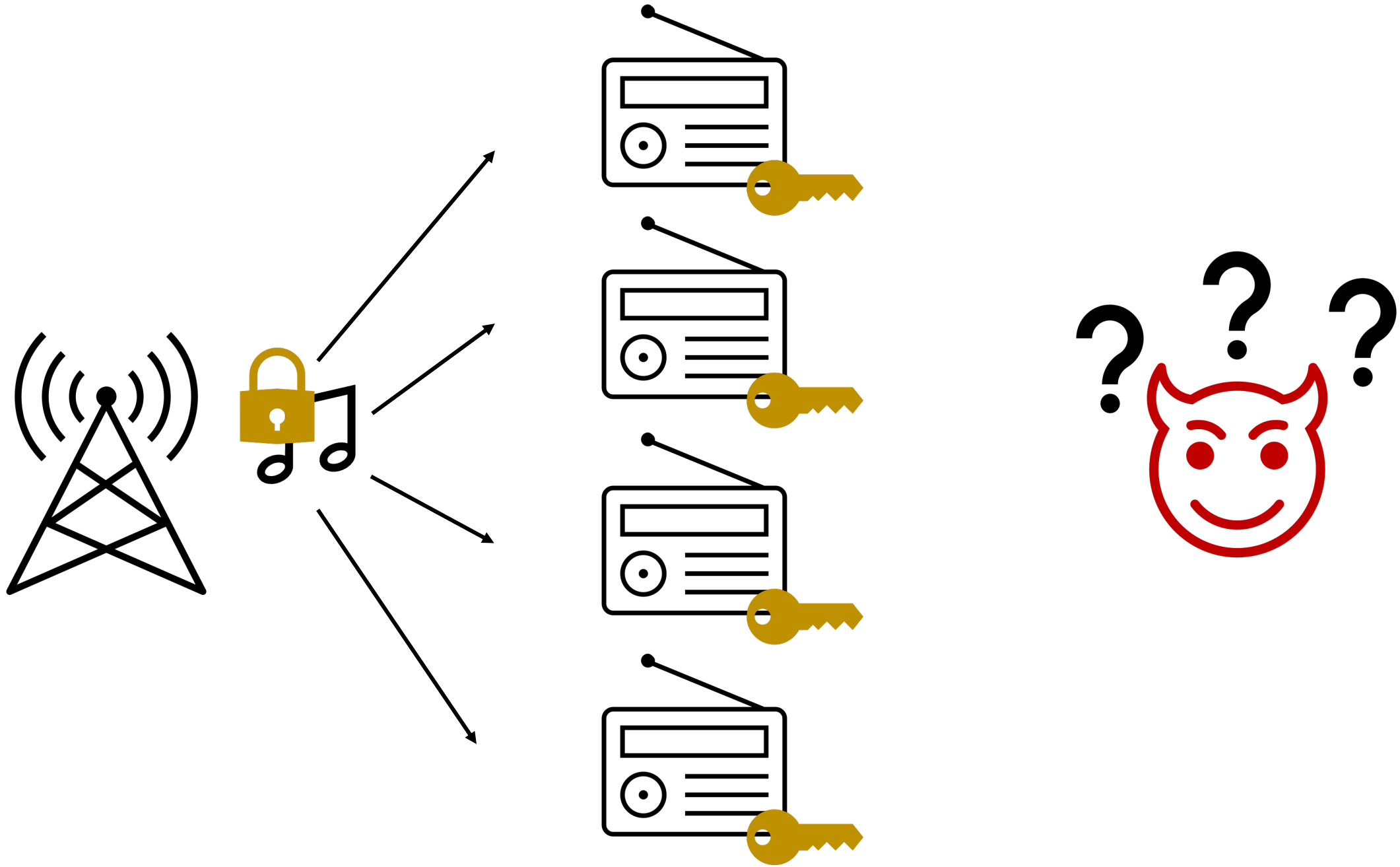
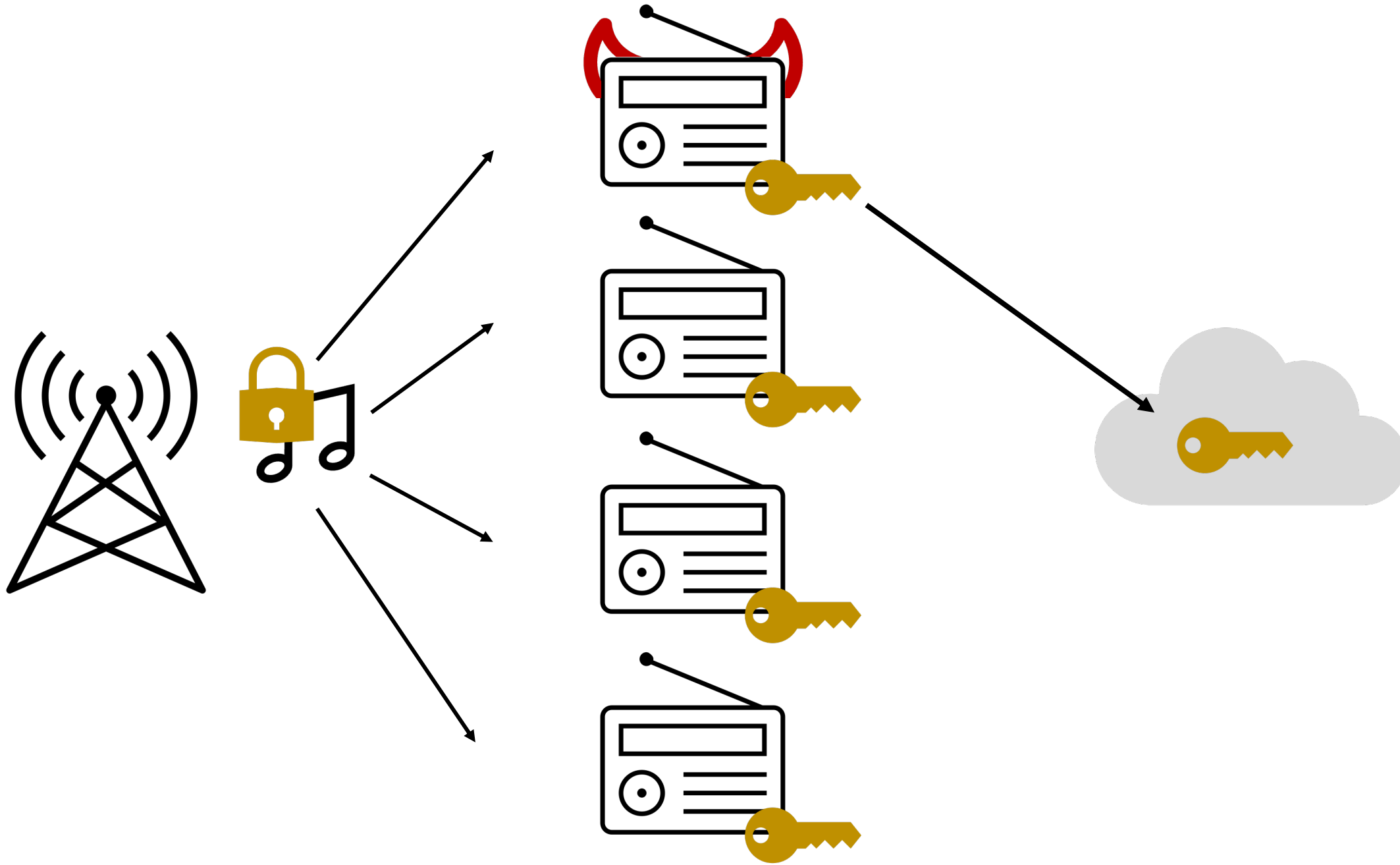
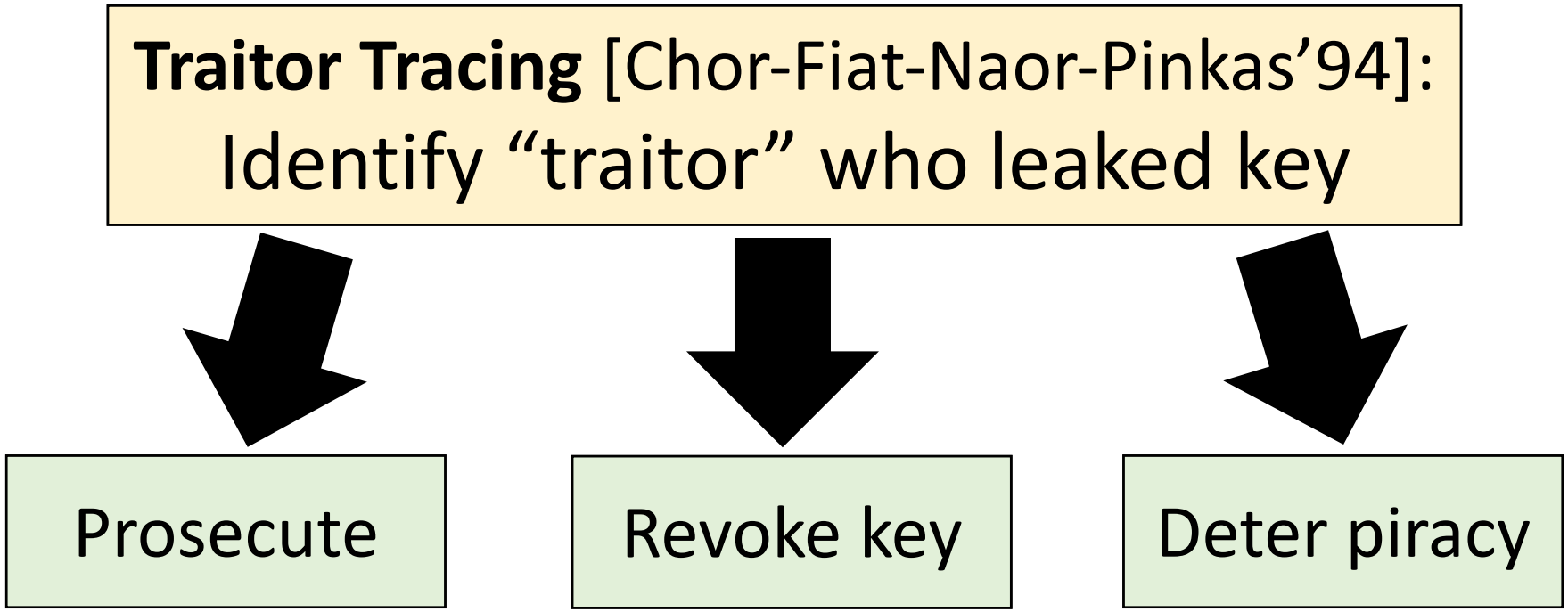
Optimal Traitor Tracing from Pairings

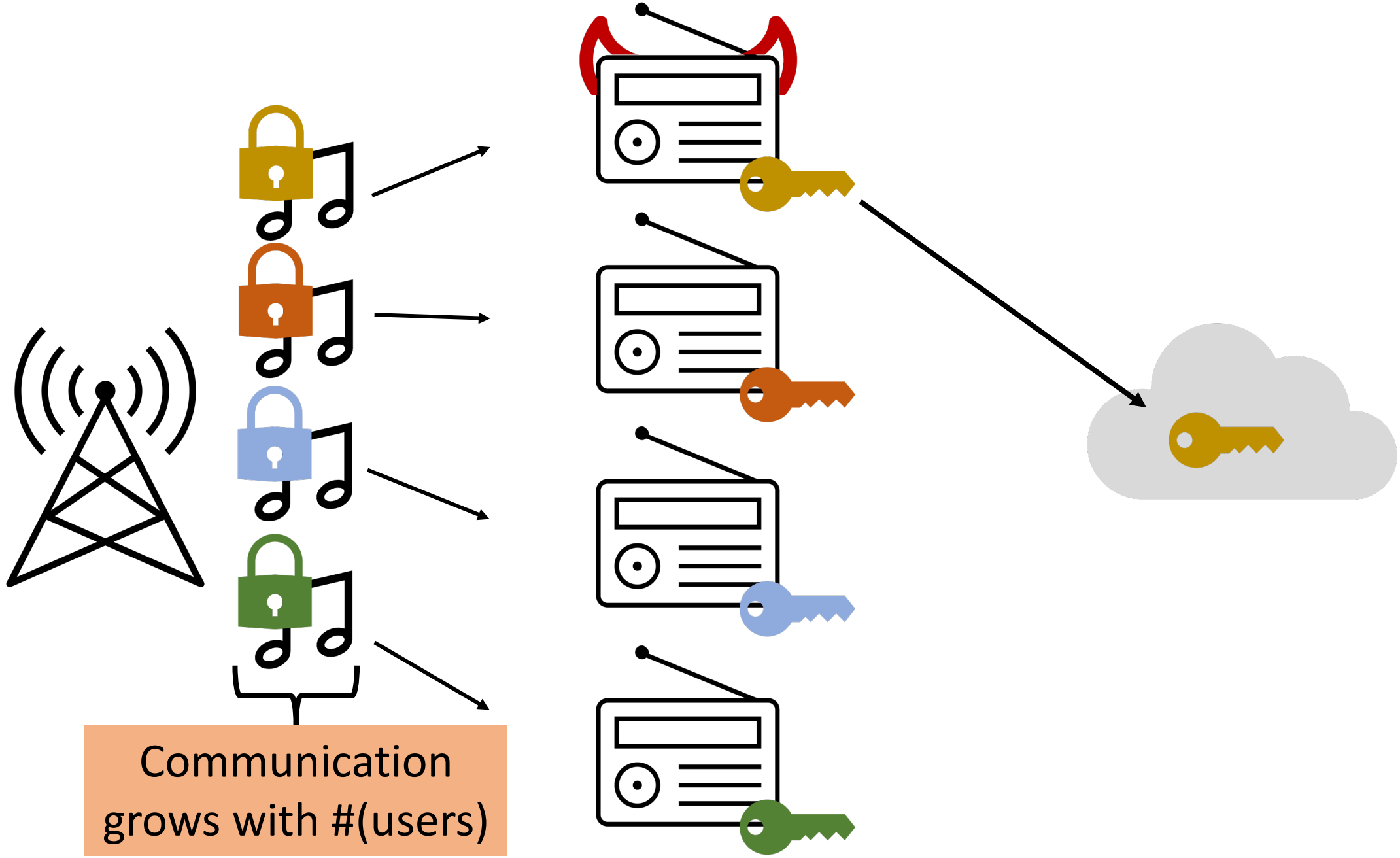
Mark Zhandry
NTT Research








Traitor Tracing [Chor-Fiat-Naor-Pinkas'94]:
Identify "traitor" who leaked key
Prosecute
Revoke key
Deter piracy


Communication
grows with #(users)

Major Goal in Cryptography:

Traitor tracing with small ciphertexts, decryption keys

Want successful tracing even if:

- Multiple traitors collude
- Leaked key embedded in obfuscated decoder program

What is known?

	Max (|ctxt| , |decr key|)	Tool
[Chor-Fiat-Naor-Pinkas'94]	N	Generic Enc
[Boneh-Naor'02, Billet-Phan'08, Z'20]	$N^{2/3}$	Generic Enc

Notes:

- Only showing collusion-resistant schemes
- Can sometimes trade-off between parameter sizes
- Sizes ignore polynomial terms in security parameter
- | encr key | also important

What is known?

	Max (|ctxt| , |decr key|)	Tool
[Chor-Fiat-Naor-Pinkas'94]	N	Generic Enc
[Boneh-Naor'02, Billet-Phan'08, Z'20]	$N^{2/3}$	Generic Enc
[Boneh-Sahai-Waters'06]	$N^{1/2}$	Pairings
[Z'20, Gong-Luo-Wee'23]	$N^{1/3}$	Pairings

Notes:

- Only showing collusion-resistant schemes
- Can sometimes trade-off between parameter sizes
- Sizes ignore polynomial terms in security parameter
- | encr key | also important

What is known?

	Max (|ctxt| , |decr key|)	Tool
[Chor-Fiat-Naor-Pinkas'94]	N	Generic Enc
[Boneh-Naor'02, Billet-Phan'08, Z'20]	$N^{2/3}$	Generic Enc
[Boneh-Sahai-Waters'06]	$N^{1/2}$	Pairings
[Z'20, Gong-Luo-Wee'23]	$N^{1/3}$	Pairings
[Garg-Gentry-Halevi-Raykova-Sahao-Waters'13, Boneh-Z'14]	1	Obfuscation
[Goyal-Koppula-Waters'18]	1	Lattices

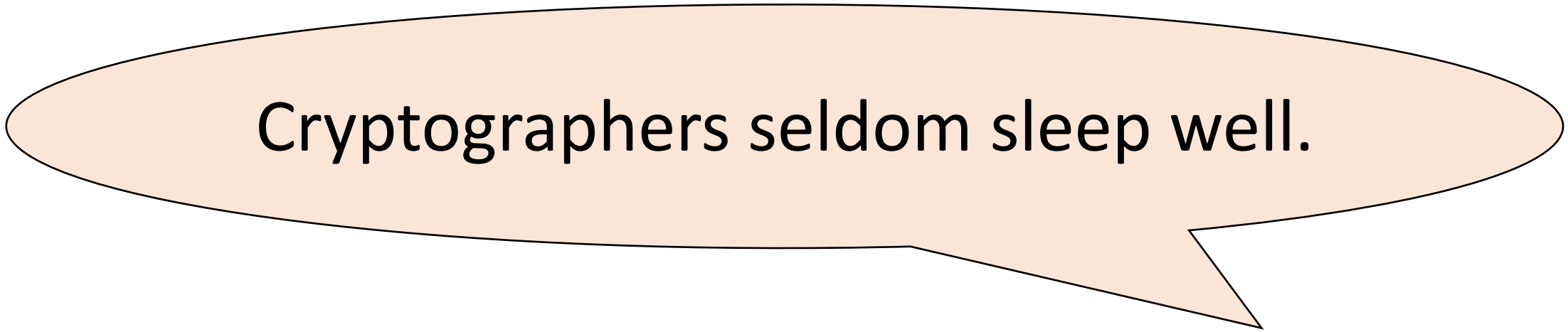
Notes:

- Only showing collusion-resistant schemes
- Can sometimes trade-off between parameter sizes
- Sizes ignore polynomial terms in security parameter
- | encr key | also important

What is known?

	Max (|ctxt| , |decr key|)	Tool
[Chor-Fiat-Naor-Pinkas'94]	N	Generic Enc
[Boneh-Naor'02, Billet-Phan'08, Z'20]	$N^{2/3}$	Generic Enc
[Boneh-Sahai-Waters'06]	$N^{1/2}$	Pairings
[Z'20, Gong-Luo-Wee'23]	$N^{1/3}$	Pairings
This work	1	**Pairings**
[Garg-Gentry-Halevi-Raykova-Sahao-Waters'13, Boneh-Z'14]	1	Obfuscation
[Goyal-Koppula-Waters'18]	1	Lattices

So, isn’t traitor tracing solved?



[Micali]

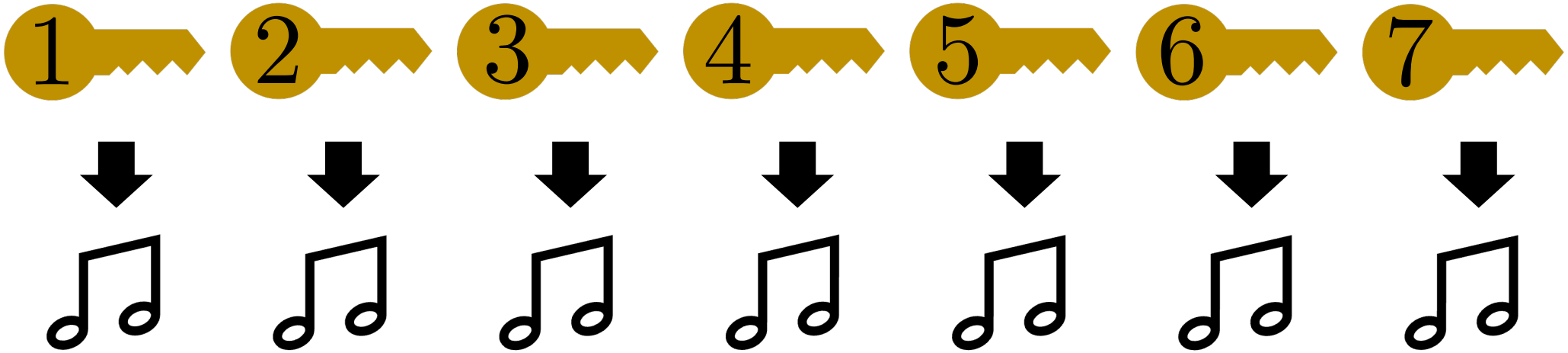
Traitor Tracing Background

The Private Linear Broadcast Approach

[Boneh-Sahai-Waters'06]

Publicly generated "normal" ciphertexts:

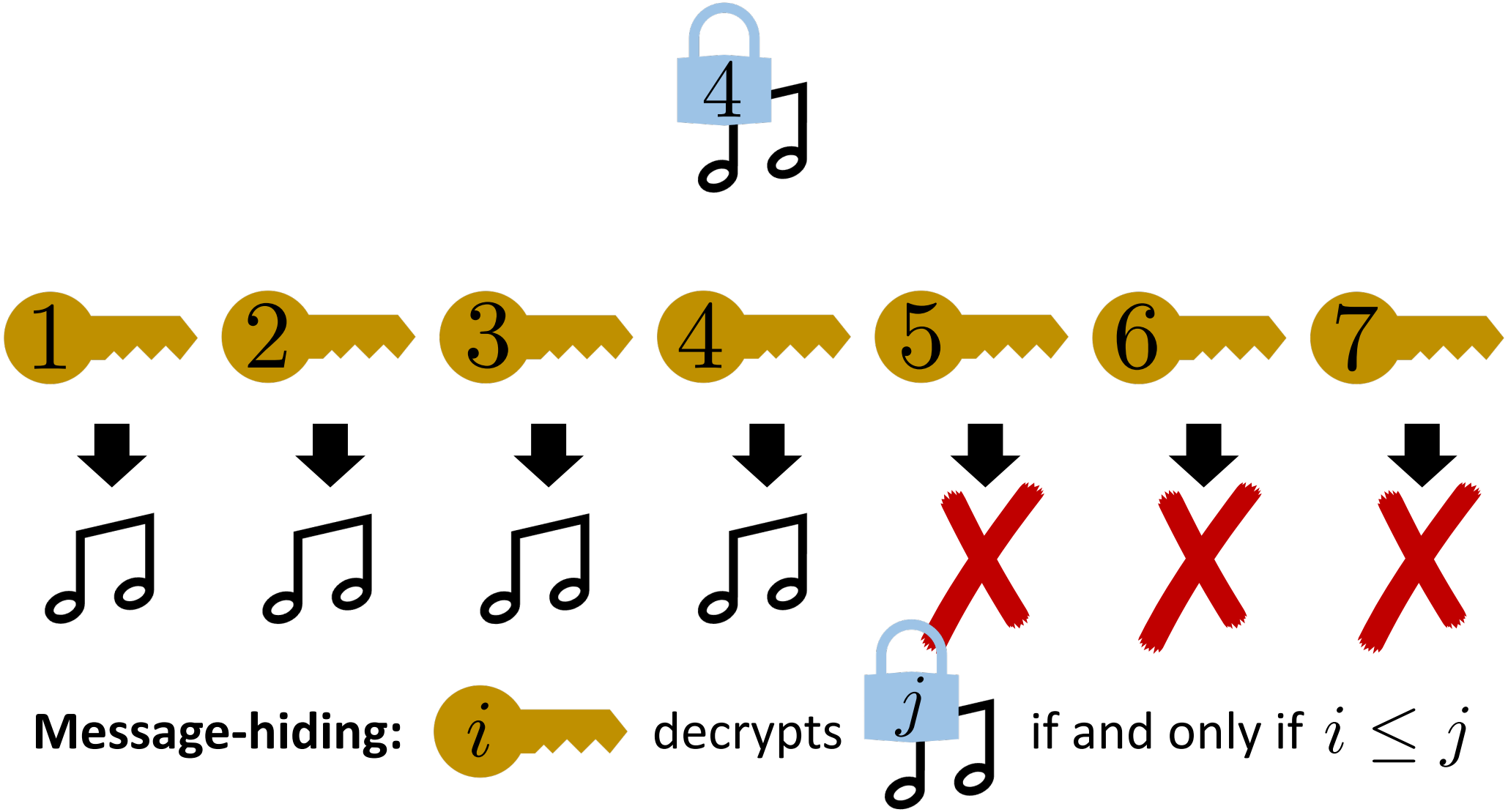
N secret keys, indexed by user #:



All secret keys decrypt normal ciphertexts

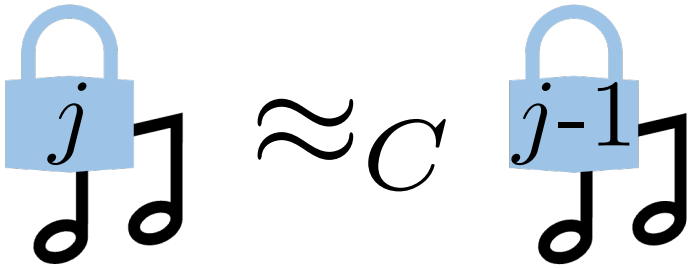
The Private Linear Broadcast Approach

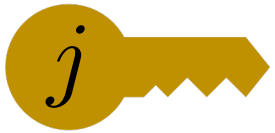
(privately-generated) indexed “tracing” ciphertexts:

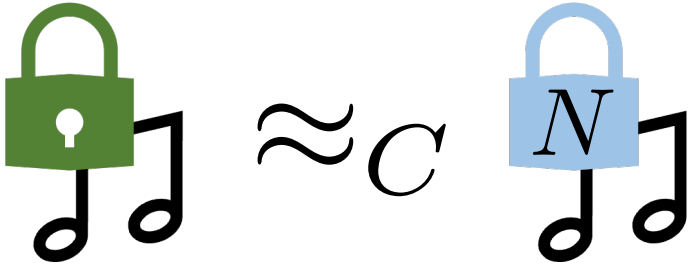


The Private Linear Broadcast Approach

Two additional requirements

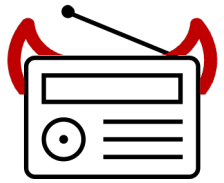
Index-hiding: 

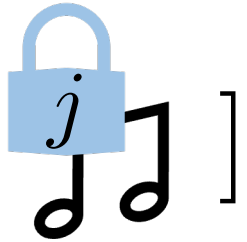
unless you possess 

Normal-hiding: 

even with all the keys

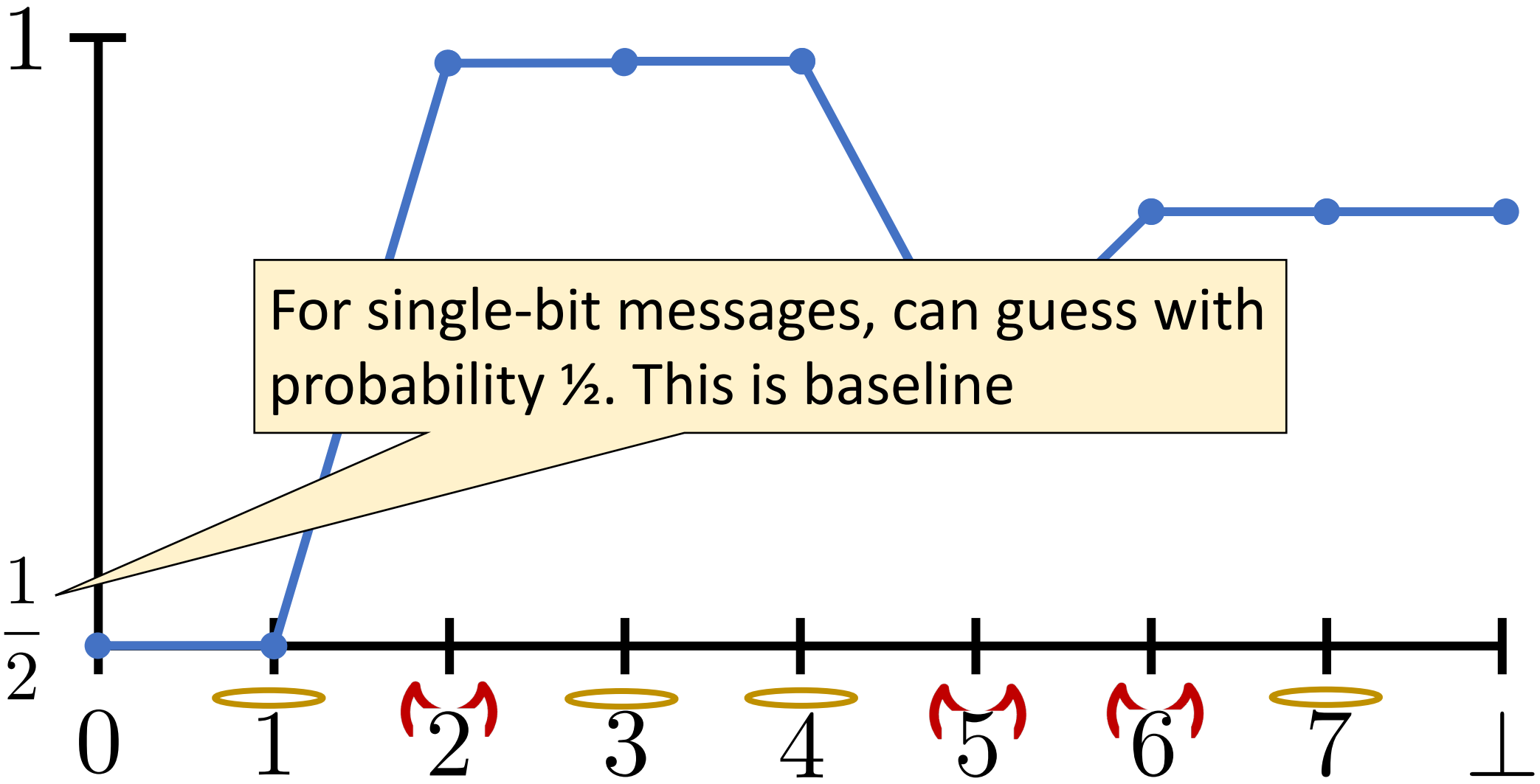
The Private Linear Broadcast Approach

Trace() :

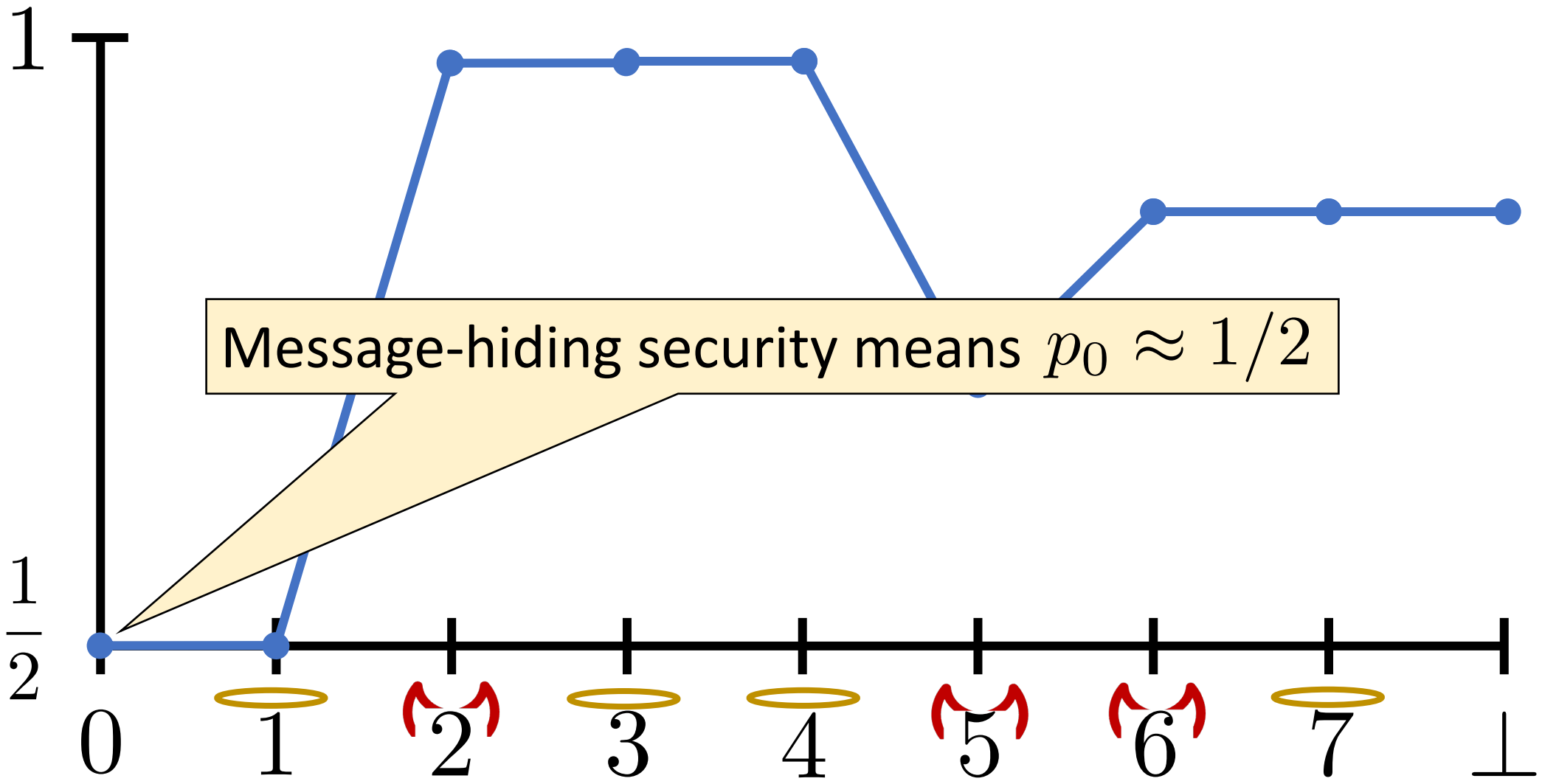
Define $p_j = \Pr[$  decrypts  $]$

$p_\perp = \Pr[$  decrypts  $]$

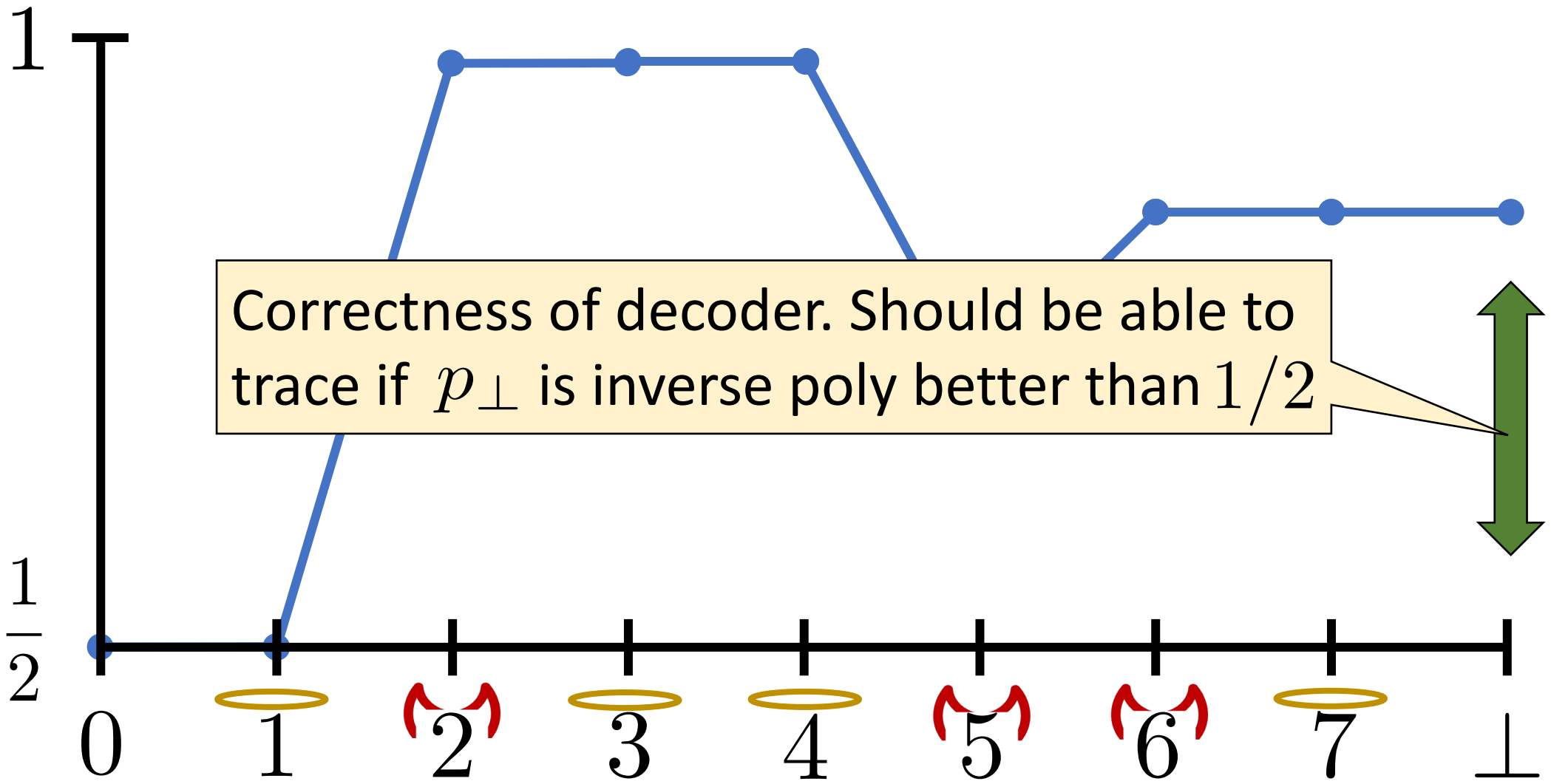
The Private Linear Broadcast Approach



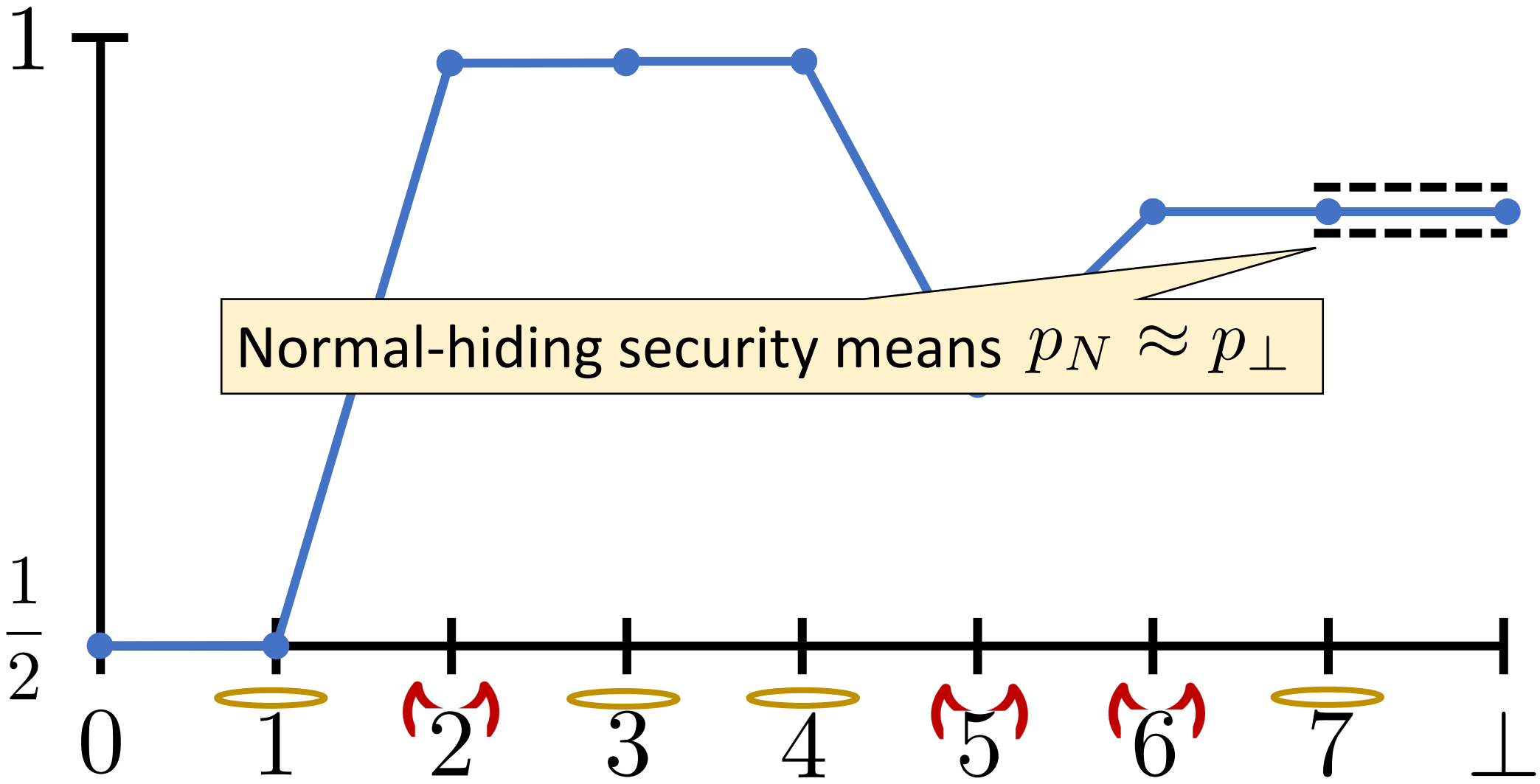
The Private Linear Broadcast Approach



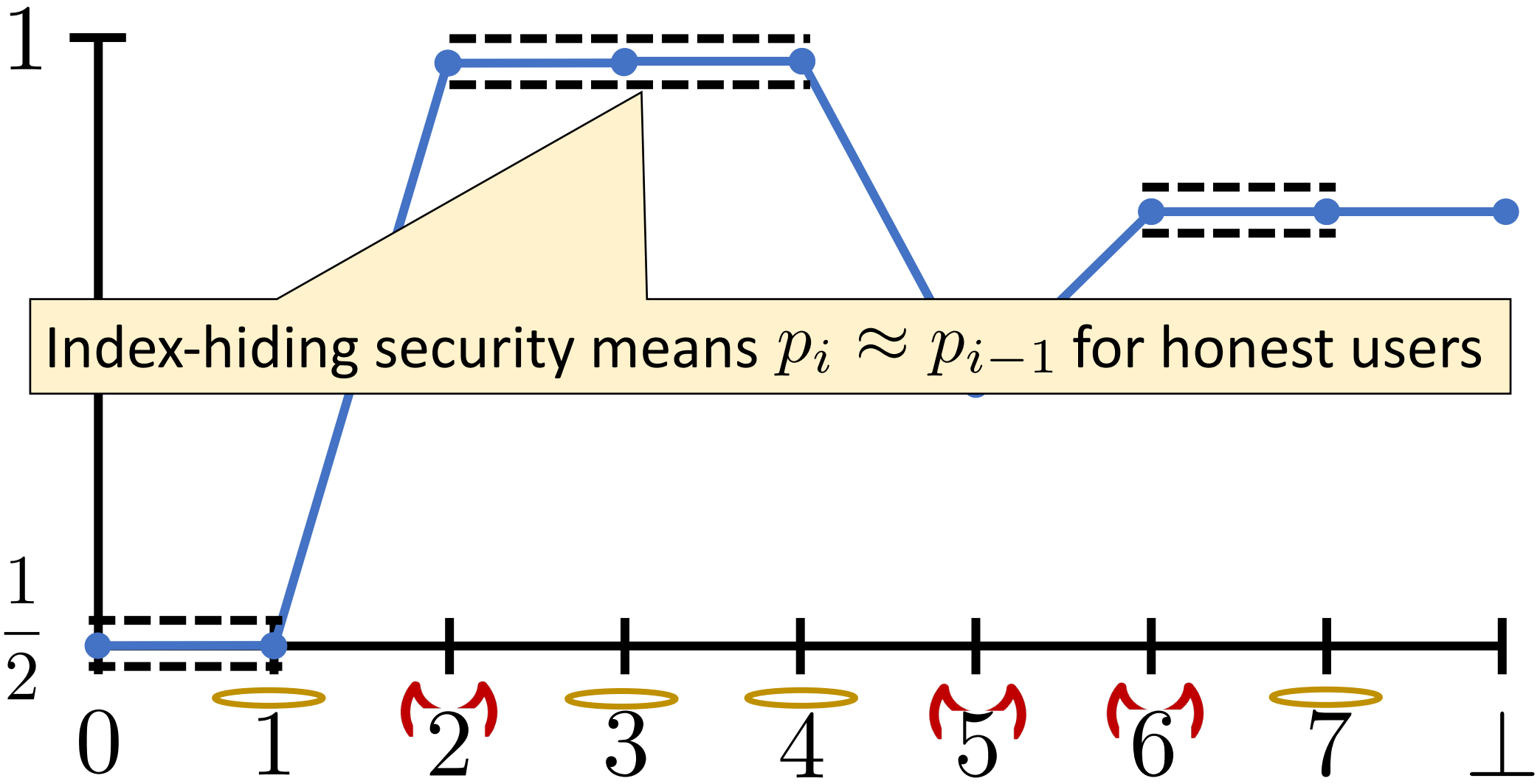
The Private Linear Broadcast Approach



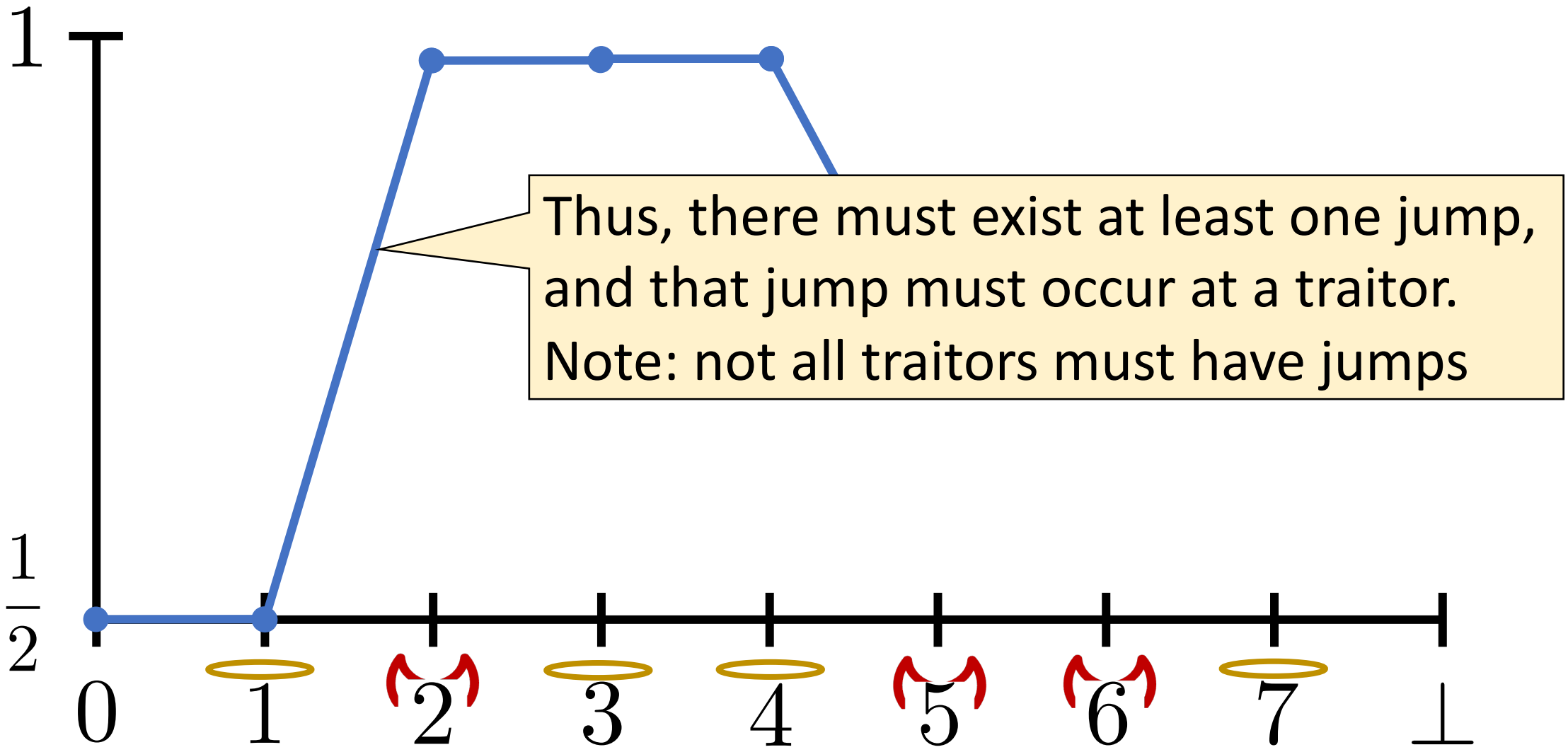
The Private Linear Broadcast Approach



The Private Linear Broadcast Approach



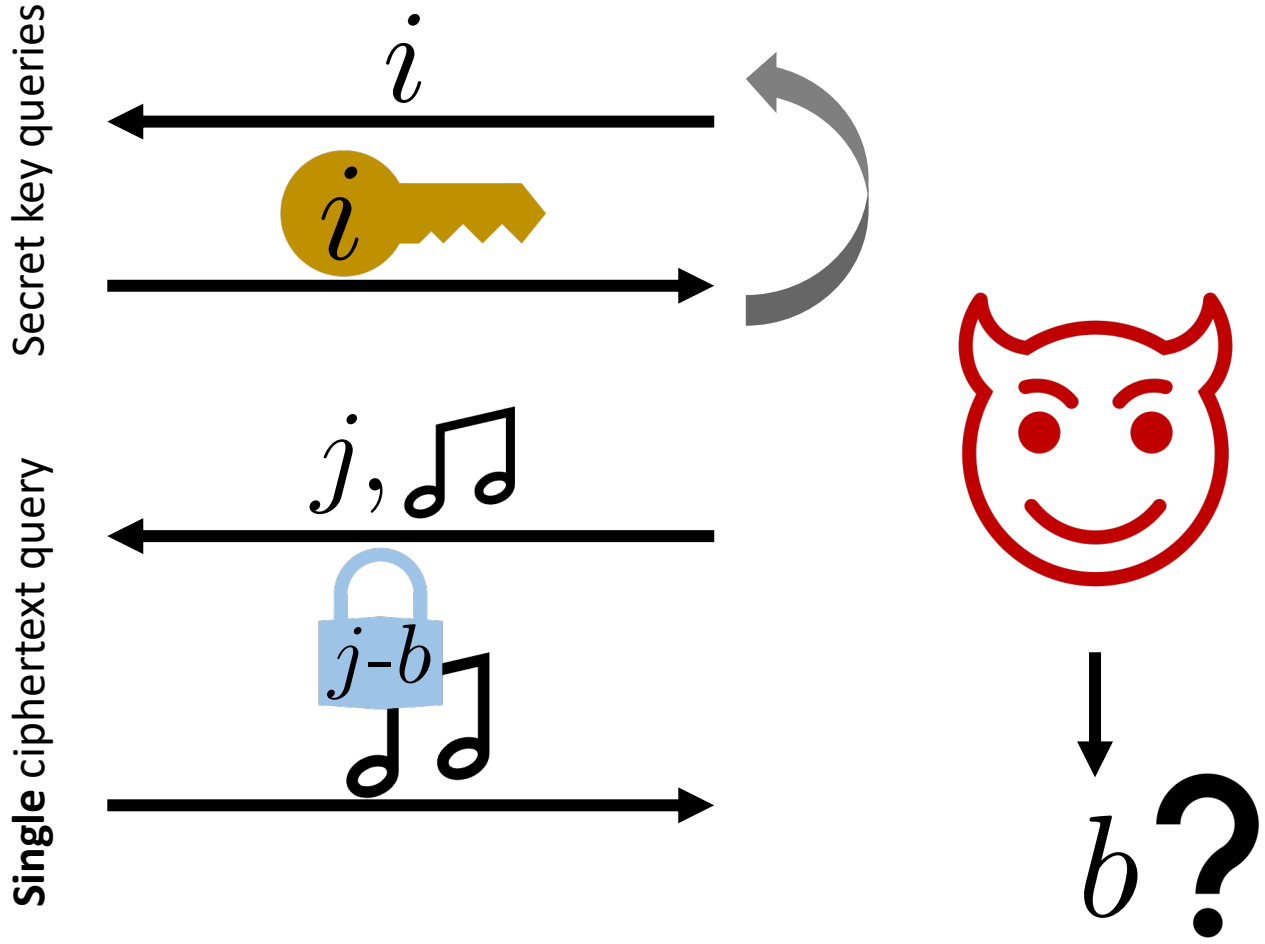
The Private Linear Broadcast Approach



Formalizing Security: 1-Ctxt Security is Insufficient

[Goyal-Koppula-Waters'18]

$$b \leftarrow \{0,1\}$$

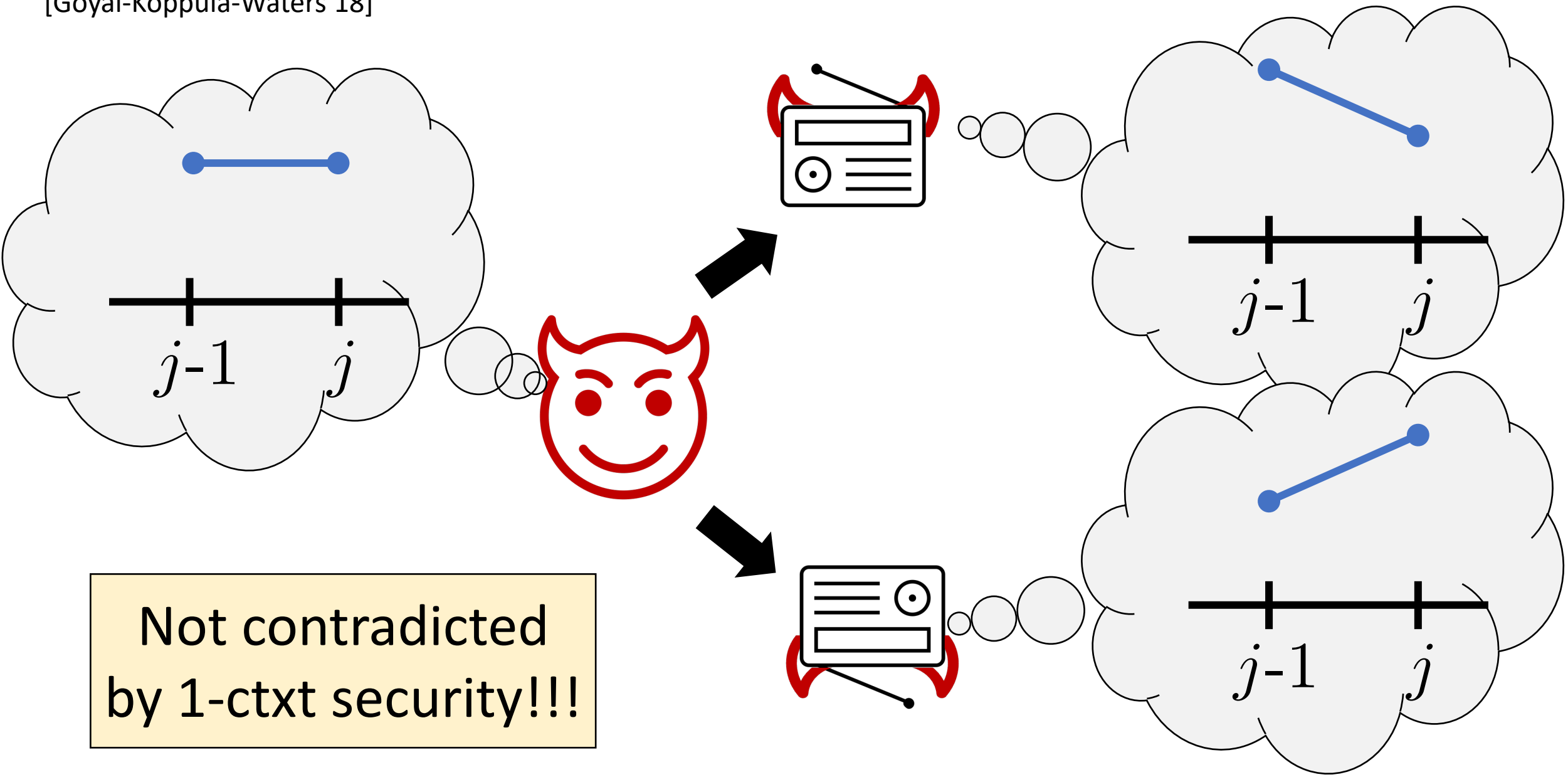


Constraints:

- For any secret key query, $i \neq j$

Formalizing Security: 1-Ctxt Security is Insufficient

[Goyal-Koppula-Waters'18]

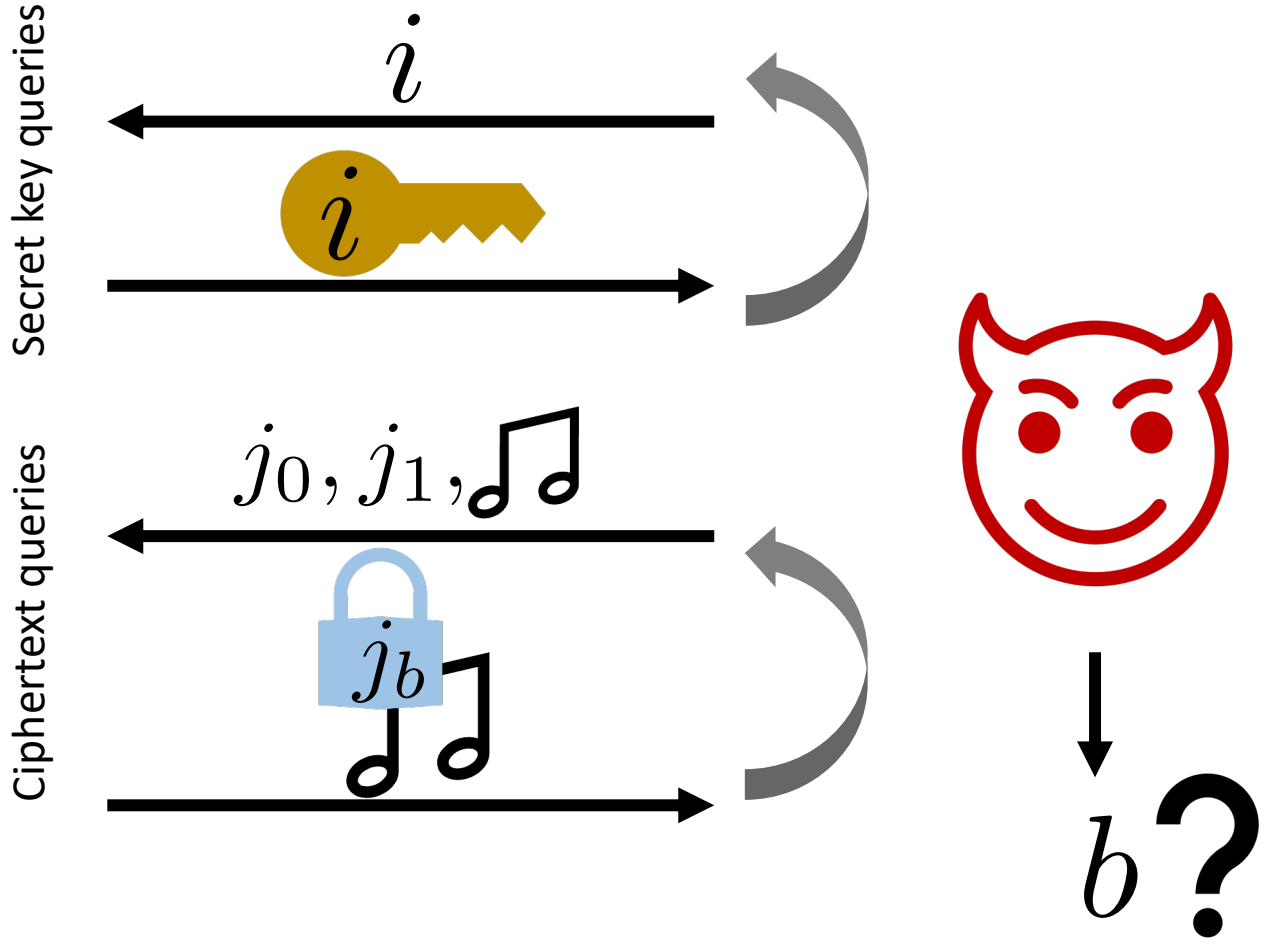


Not contradicted by 1-ctxt security!!!

Formalizing Security: **2**-ctxt security is **S**ufficient

[Goyal-Koppula-Waters'18]

$$b \leftarrow \{0,1\}$$

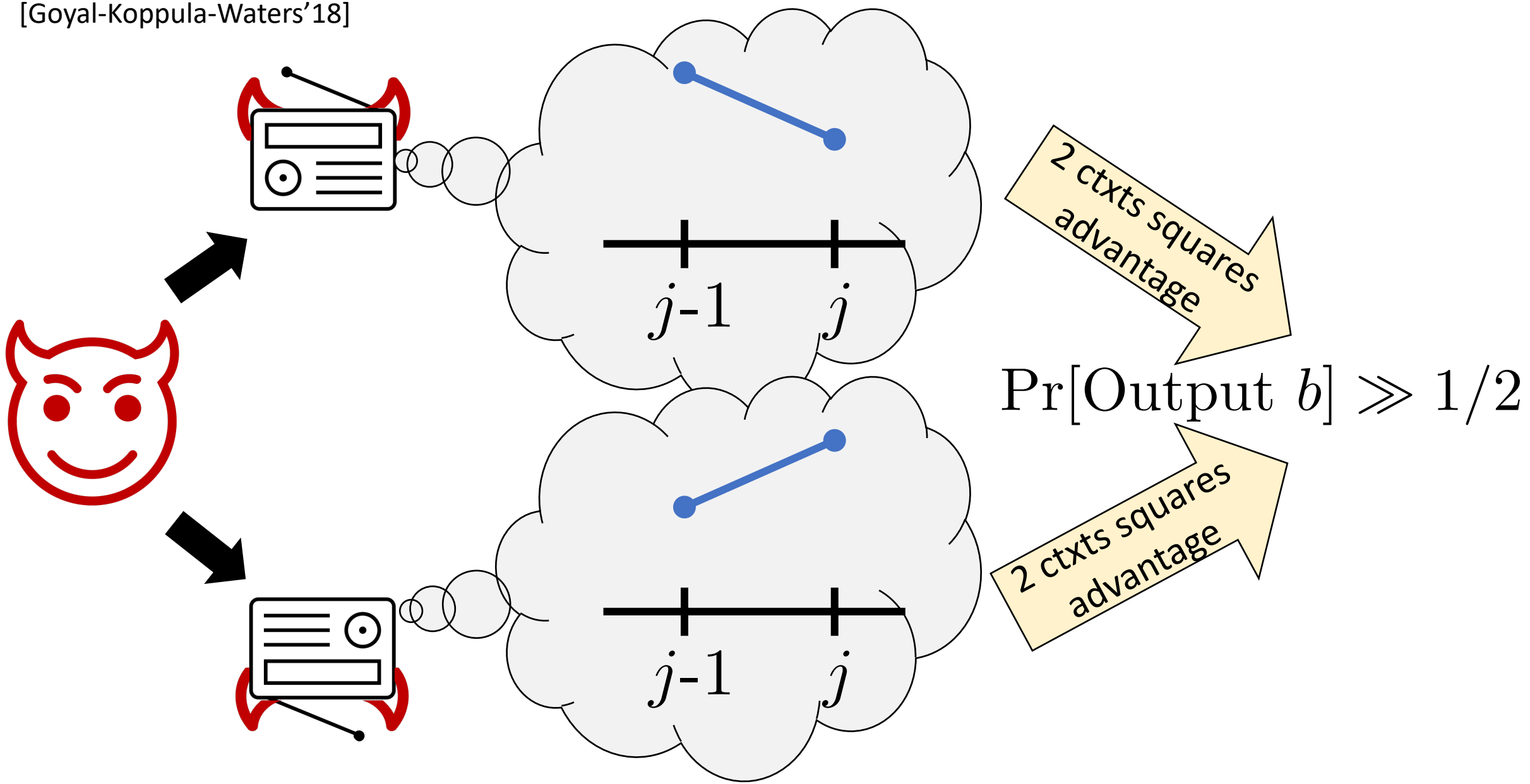


Constraints:

- For any pair of queries, $i \notin (j_0, j_1]$
- At most **2** ciphertext queries

Formalizing Security: **2**-ctxt security is **S**ufficient

[Goyal-Koppula-Waters'18]



Theorem [Boneh-Sahai-Waters'06, Goyal-Koppula-Waters'18]:

2-ctxt message
+ 2-ctxt index
+ 2-ctxt norm

g

empty

Decryption just indicates ✓ or ✗

Theorem [Goyal-Kop a-Waters'18]:

Message-less PLBE w/

2-ctxt index-hiding
2-ctxt normal-hiding

+ ABE for circuits

Plain PLBE w/

2-ctxt message-hiding
2-ctxt index-hiding
2-ctxt normal-hiding

From lattices [Gorbunov-Vaikuntanathan-Wee'13]

Theorem [Boneh-Sahai-Waters'06, Goyal-Koppula-Waters'18]:

2-ctxt message-hiding
+ 2-ctxt index-hiding
+ 2-ctxt normal-hiding

Traitor tracing

Theorem [Goyal-Koppula-Waters'18]:

Message-less PLBE w/

q_1-ctxt index-hiding
q_2-ctxt normal-hiding

+ ABE which handles PLBE decryption

Plain PLBE w/

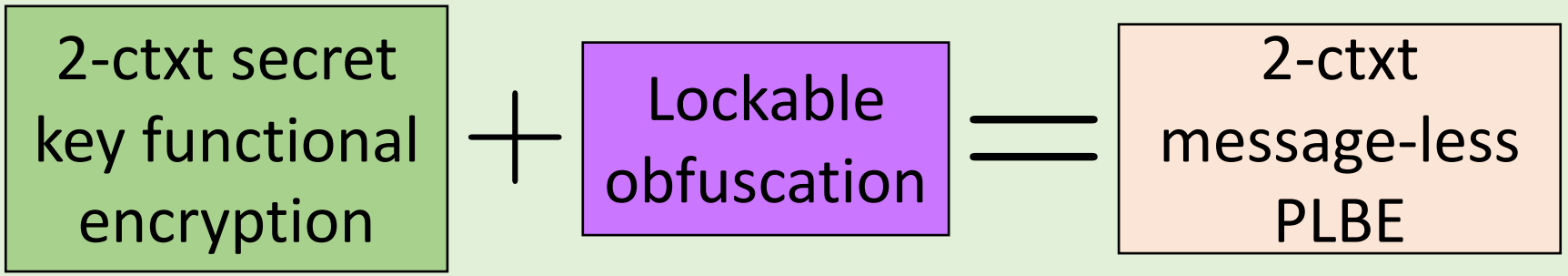
q_0-ctxt message-hiding
q_1-ctxt index-hiding
q_2-ctxt normal-hiding

ABE for log-depth from pairings [Goyal-Pandey-Sahai-Waters'06, Ishai-Wee'14, Chen-Gay-Wee'15, Lin-Luo'20]

Constructing 2-ctxt Message-less PLBE?

Theorem [Goyal-Koppula-Waters'18]: Lattices → 2-ctxt message-less PLBE

Theorem [Chen-Vaikuntanathan-Waters-Wee-Wichs '18]:



Thm [Goyal-Koppula-Waters'17, Wichs-Zirdelis'17]: Lattices → lockable Obf

Both approaches firmly rooted in lattices

Our Techniques

Theorem (This Work):

Plain PLBE w/
2-ctxt message-hiding
＋ 2-ctxt index-hiding
＋ **1**-ctxt normal-hiding

Traitor tracing

Theorem (This Work):

Message-less PLBE w/
2-ctxt index-hiding
＋ **1**-ctxt [illegible]

Completely removes lockable obfuscation from prior lattice-based schemes

In log-depth setting, both can be instantiated from pairings

Corollary (informal):

(log-depth) Weak PRFs
＋ ABE (for log-depth comp.)

Traitor Tracing

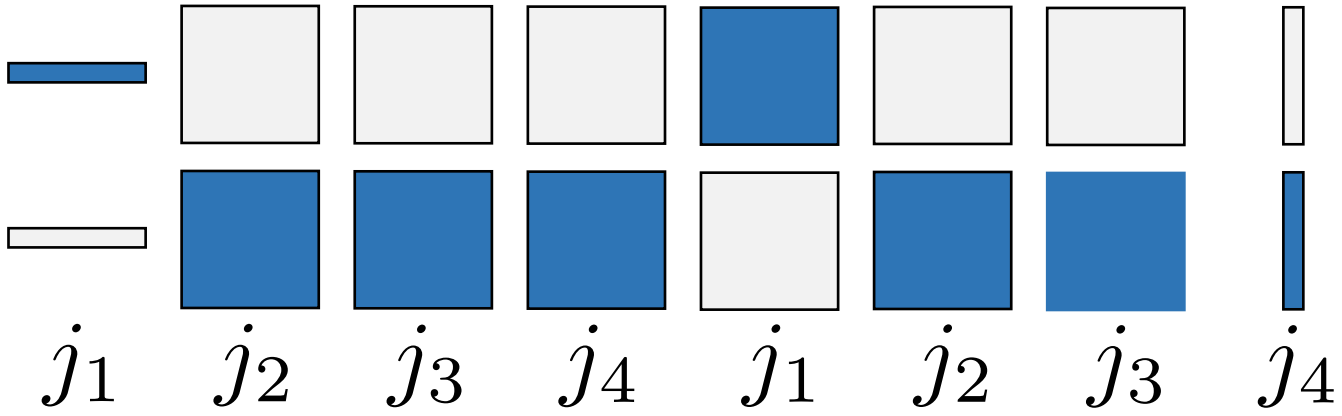
Message-less PLBE from weak PRFs

A 1-ctxt Secure Message-less PLBE

Based on [Sahai-Seyalioglu'10, Gorbunov-Vaikuntanathan-Wee'12]

Predicate $C_i(j) = (i \leq j)$ is computable in log-depth

[Barrington'86]: Can represent C_i as matrix branching program



$$j = 0111_b$$

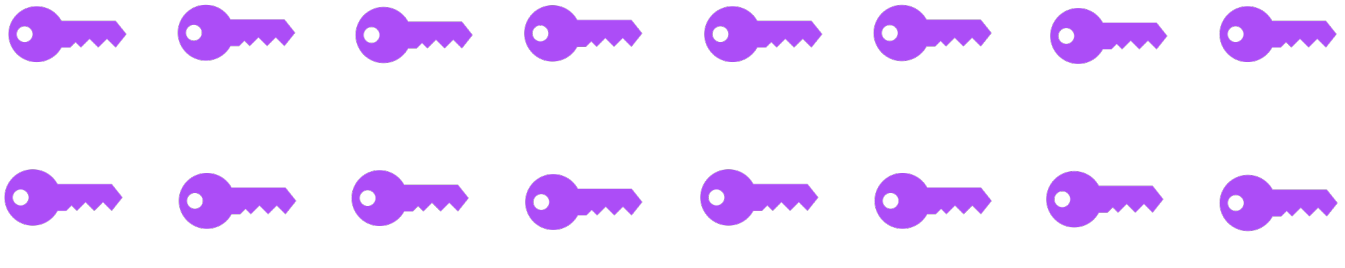
Iterated matrix product gives $C_i(j)$

[Kilian'88]: Can make **blue** matrices uniformly random conditioned given product. Simultaneously holds for any input

A 1-ctxt Secure Message-less PLBE

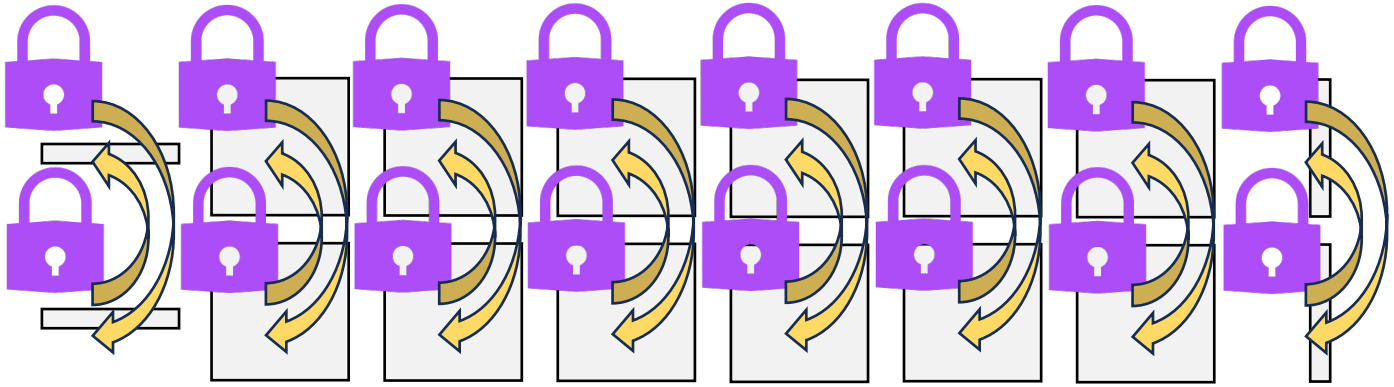
Based on [Sahai-Seyalioglu'10, Gorbunov-Vaikuntanathan-Wee'12]

$=$

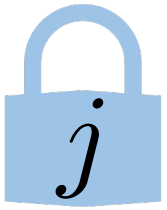


Ordinary encryption keys

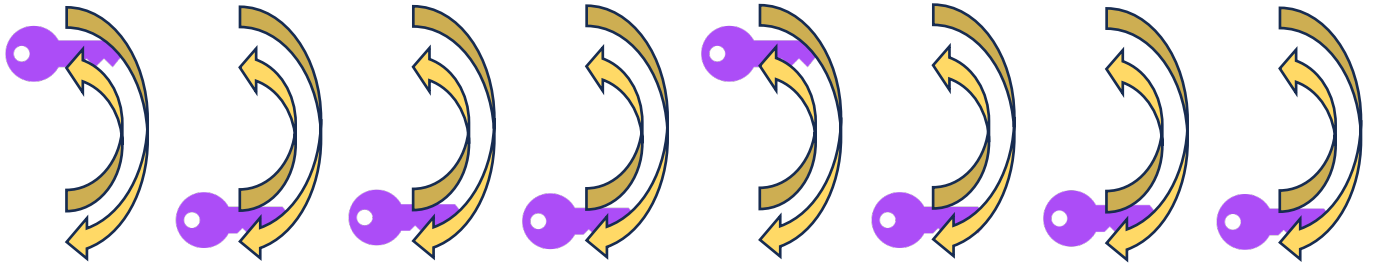
$=$



C_i



$=$

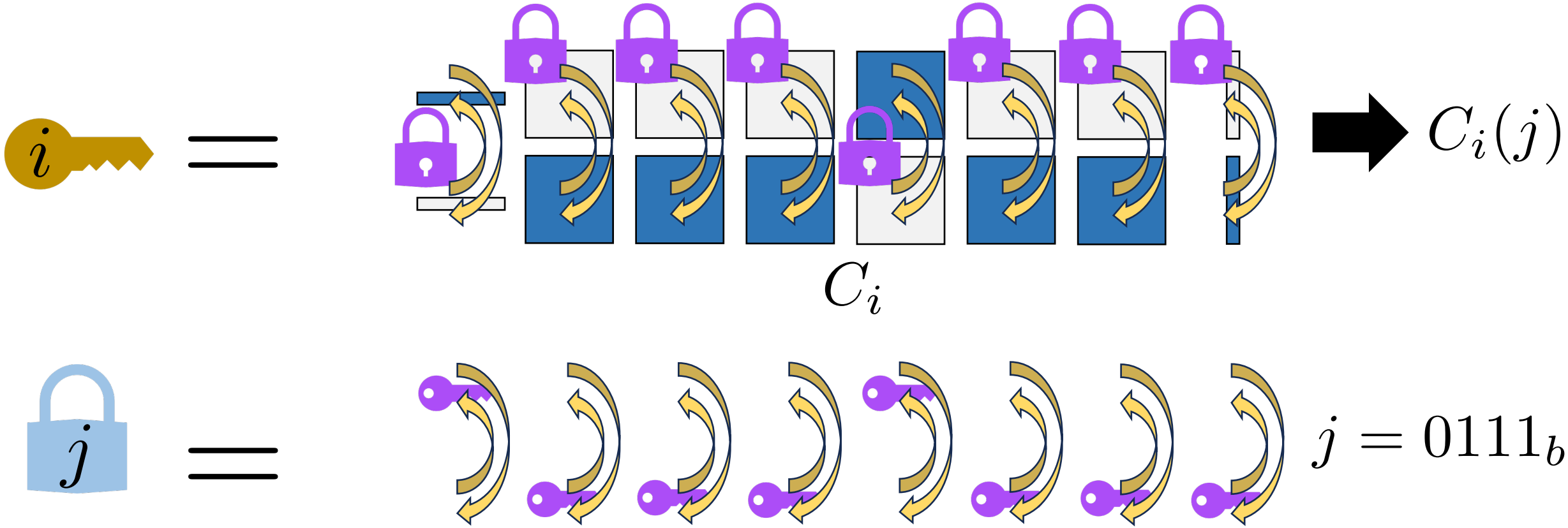


$j = 0111_b$

A 1-ctxt Secure Message-less PLBE

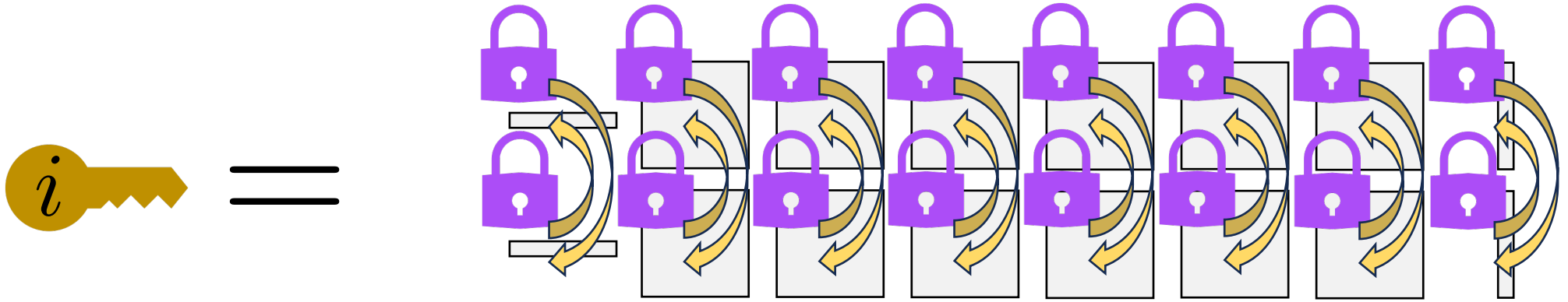
Based on [Sahai-Seyalioglu'10, Gorbunov-Vaikuntanathan-Wee'12]

Decryption:



A 1-ctxt Secure Message-less PLBE

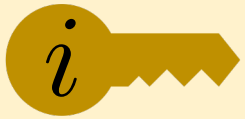
Based on [Sahai-Seyalioglu'10, Gorbunov-Vaikuntanathan-Wee'12]



Our crucial modification: $\alpha_i C_i$ α_i uniform for each key

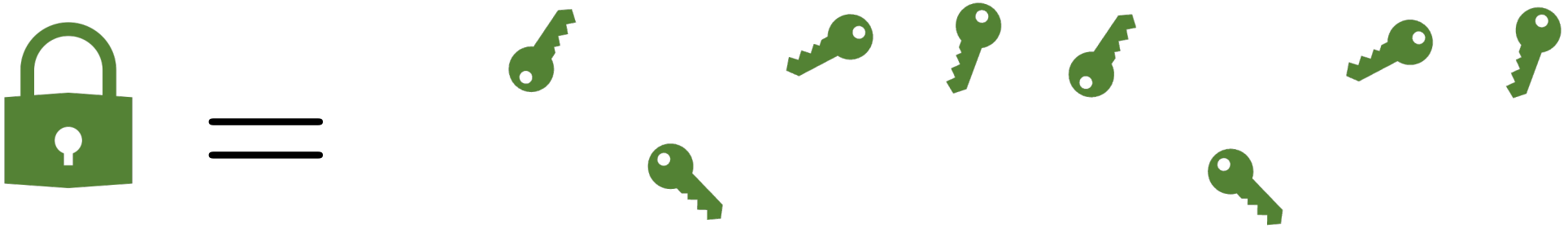
$i \leq j$ ➡ $C_i(j) = 1$

➡ $\alpha_i C_i(j) = \alpha_i$ uniformly random

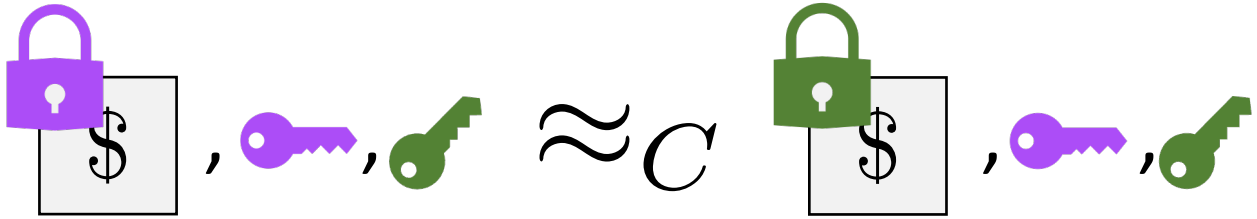
➡  decrypts to uniformly random matrices

Our crucial modification: $\alpha_i C_i$ α_i uniform for each key

Adding Public Mode to Obtain Message-less PLE



Random keys in random positions

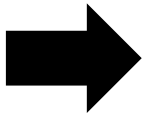
Suppose  $\approx_C$

Easy to obtain from weak PRFs

Upgrading to 2-ctxt Security

[Gorbunov-Vaikuntanathan-Wee'12]:
non-trivial techniques → upgrade to 2-ctxt security

Techniques do not *upgrade* normal-hiding. Nevertheless, they do *preserve* 1-ctxt normal-hiding



Theorem (This Work):

Weak PRFs

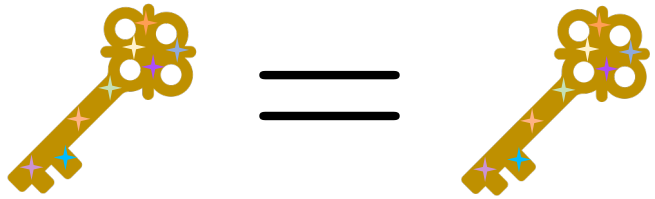
Message-less PLBE w/
2-ctxt index-hiding
+ **1**-ctxt normal-hiding
Similar depth to weak PRF

Traitor Tracing from 1-ctxt Normal-Hiding

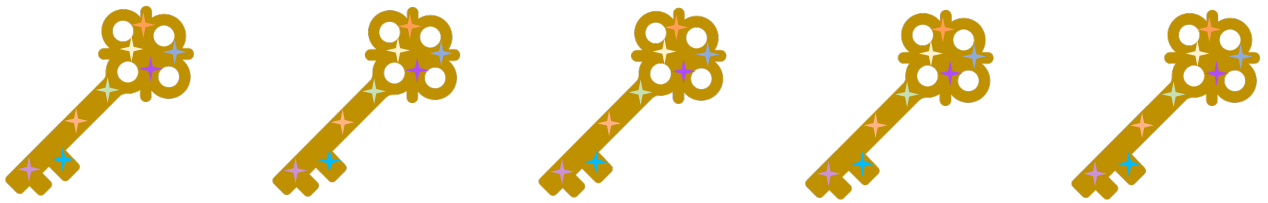
Can We Upgrade to 2-Bounded Security?

Simple black-box Idea: several parallel instances

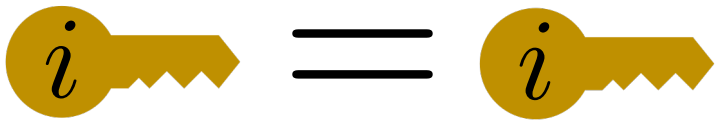
2-bounded master key



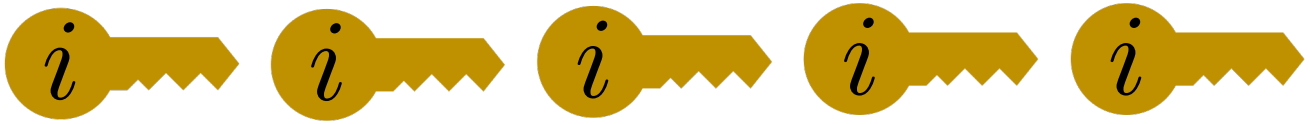
Several independent 1-bounded master keys



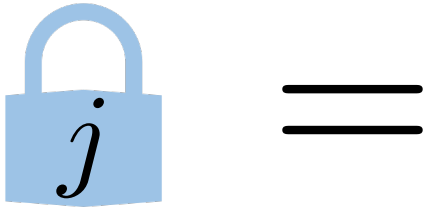
2-bounded secret key



Several independent 1-bounded secret keys



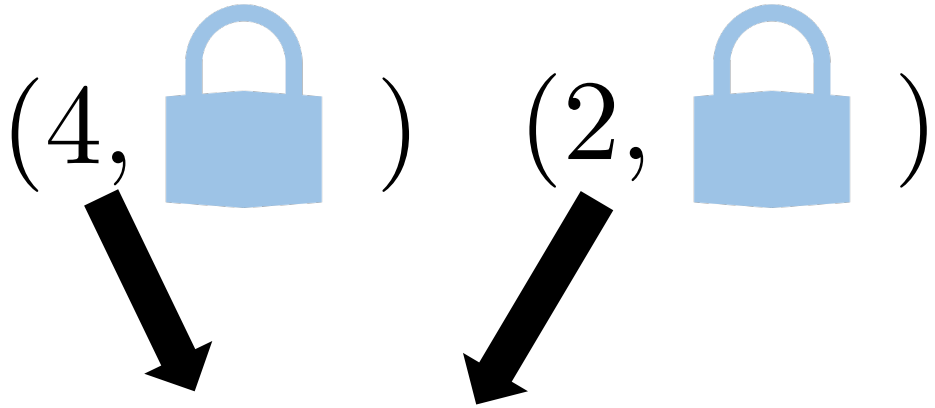
2-bounded ciphertext



Random choice of single 1-bounded ciphertext

$(4,$  $)$

Can We Upgrade to 2-Bounded Security?



As long as instances are different, each instance gets single ciphertext

In this case, security reduces to 1-ctxt security

Problem: always non-trivial probability instances are same

In these cases, no security

Weak Decoder-Based Normal-Hiding

Lemma (This Work, informal): Instantiate with 5 parallel instances. Then among decoders with $p_\perp \geq 39/40$, at least a fraction $1/82$ of them have $p_N \geq 21/40$

That is, ***very*** good decoders can't have tiny p_N too often

Weak Decoder-Based Normal-Hiding

Lemma (This Work, informal): Instantiate with 5 parallel instances. Then among decoders with $p_\perp \geq 39/40$, at least a fraction $1/82$ of them have $p_N \geq 21/40$

Re-interpreting Proof of Tracing from 2-ctxt Security:

Define $\Delta\left(\right.$  $\left.\right) = p_\perp - p_N$

2-ctxt normal-hiding ➡ $\mathbb{E}[\Delta^2] \approx 0$ ➡ $\Pr[\Delta \not\approx 0] \approx 0$

Weak Decoder-Based Normal-Hiding

Lemma (This Work, informal): Instantiate with 5 parallel instances. Then among decoders with $p_\perp \geq 39/40$, at least a fraction $1/82$ of them have $p_N \geq 21/40$

Proof: Assume for simplicity that all decoders have $p_\perp \geq 39/40$

Define $p_{\perp,k}$  as $p_\perp$ when encrypting to instance k

Similarly define $p_{N,k}$, Δ_k

Adapting same proof ➡ $\mathbb{E}[\Delta_k \Delta_\ell] \approx 0$ for $k \neq \ell$

Weak Decoder-Based Normal-Hiding

Lemma (This Work, informal): Instantiate with 5 parallel instances. Then among decoders with $p_\perp \geq 39/40$, at least a fraction $1/82$ of them have $p_N \geq 21/40$

Proof: $\mathbb{E}[\Delta_k \Delta_\ell] \approx 0$ for $k \neq \ell$, $\mathbb{E}[\Delta_k^2] \leq 1$ trivially

$$\Rightarrow \mathbb{E}[\Delta^2] = \mathbb{E}\left[\left(\frac{1}{5}\sum_k \Delta_k\right)^2\right] \lessapprox \frac{1}{5}$$

$$\Rightarrow \Pr[\Delta > 9/20] \leq \Pr[\Delta^2 \leq 81/400] \lessapprox \frac{1}{5} \times \frac{400}{81} = 1 - \frac{1}{81}$$

Say, $\Pr[\Delta > 9/20] \leq 1 - 1/82 \Longrightarrow \Pr[\Delta \leq 9/20] \geq 1/82$

Weak Decoder-Based Normal-Hiding

Lemma (This Work, informal): Instantiate with 5 parallel instances. Then among decoders with $p_\perp \geq 39/40$, at least a fraction $1/82$ of them have $p_N \geq 21/40$

Proof: $\Pr[\Delta \leq 9/20] \geq 1/82$

$$\Pr[p_N \geq 21/40] \geq \Pr[p_\perp \geq 39/40 \text{ and } \Delta \leq 9/20]$$

Recall we made simplifying assumption that $p_\perp \geq 39/40$ always. Can extend to general case by careful conditioning

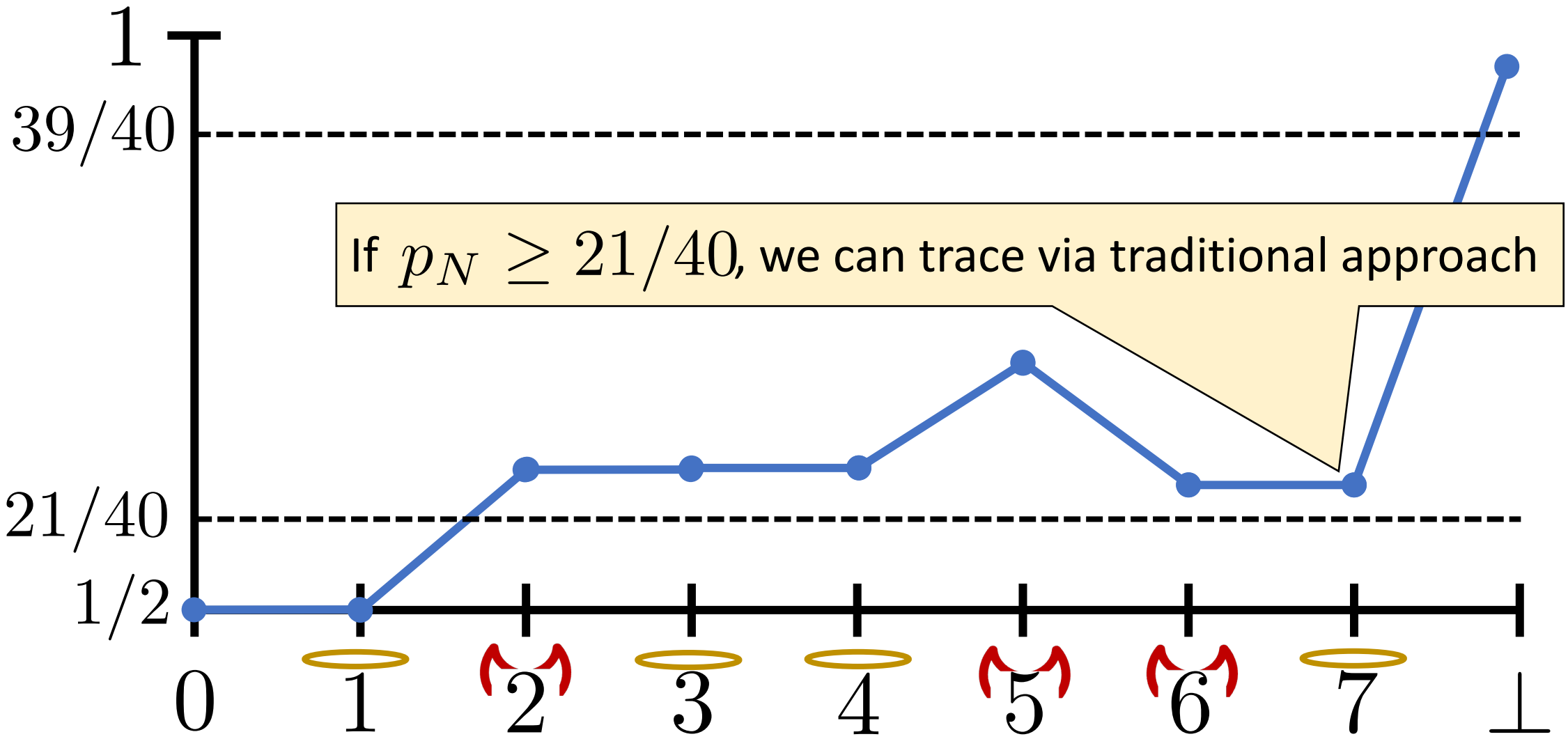
Weak Decoder-Based Normal-Hiding

Lemma (This Work, informal): Instantiate with 5 parallel instances. Then among decoders with $p_\perp \geq 39/40$, at least a fraction $1/82$ of them have $p_N \geq 21/40$

Proof: $\Pr[\Delta \leq 9/20] \geq 1/82$

$$\Pr[p_N \geq 21/40] \geq \Pr[p_\perp \geq 39/40 \text{ and } \Delta \leq 9/20]$$
$$= \Pr[\Delta \leq 9/20] \geq 1/82$$

Our Tweaked Private Linear Broadcast Approach



Our Tweaked Private Linear Broadcast Approach

Called “threshold” traitor tracing [Naor-Pinkas’98]

Problem: Our tracing algorithm

- Only has guarantees on decoders with high constant decryption probability
- Tracing of such decoders only successful with low constant probability

Called “risky” traitor tracing [Goyal-Koppula-Russell-Waters’17]

Theorem [Z’20]: Can generically remove both risky and threshold limitations. As long as probabilities are constant, no asymptotic change to parameters.

Theorem (This Work):

Plain PLBE w/
2-ctxt message-hiding
＋ 2-ctxt index-hiding
＋**1**-ctxt normal-hiding

Traitor tracing

Corollary (informal):

(log-depth) Weak PRFs
＋ ABE (for log-depth comp.)

Traitor Tracing

Finishing Touches:
Log-Depth Weak PRFs from Pairings

Naor–Reingold PRF

Let $\mathbb{G}$ be a cyclic group where DDH holds:

$$(g, g^a, g^b, g^c) \approx_C (g, g^a, g^b, g^{ab})$$

Theorem [Naor-Reingold'97]: If DDH holds in $\mathbb{G}$, then

$$F(k, x) = g^{\prod_{i=1}^{n} a_{i,x_i}} \qquad k = (a_{i,b})_{i \in [n], b \in \{0,1\}}$$

is a *strongly* secure PRF

Evaluation = iterated multiplication over $\mathbb{Z}$ followed by iter. mult. over $\mathbb{G}$

➡ If $\mathbb{G} \subseteq \mathbb{Z}_p^*$, then two iter mult., followed by modular reduction

Theorem [Beame-Cook-Hoover'84]: Iter. mult. and mod. reduction in log-depth

Naor–Reingold From Pairings

Problem: in known cryptographic pairings, two (or three) groups:

- Source group(s): elliptic curve groups – unclear if iter. mult. Is log-depth
- Target group: subgroup of extension field $\mathbb{F}_{p^k}$, $k > 1$, [Beame-Cook-Hoover'84] doesn't trivially apply

Solution:

Theorem (This work?): iterated multiplication over finite fields in log-depth

Proof Sketch: view scalar multiplication over $\mathbb{F}_{p^k}$ as polynomial multiplication over $\mathbb{F}_p$. Reduces to (1) polynomial evaluation, (2) parallel iterated product of evaluated points, (3) polynomial interpolation

What about k-LIN?

Many ABE systems can rely on the weaker assumption called k-LIN:

$$(g^{\mathbf{A}}, g^{\mathbf{A}\cdot\mathbf{s}}) \approx_C (g^{\mathbf{A}}, g^{\mathbf{u}})$$

$$\mathbf{s} \leftarrow \mathbb{Z}_p^k, \mathbf{A} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \mathbf{u} \leftarrow \mathbb{Z}_p^{k+1}$$

Would therefore like to get traitor tracing from k-LIN.

Problem: Extensions of Naor-Reingold to k-LIN (e.g. [Lewko-Waters'09, Escala-Herold-Kiltz-Ràfols-Villar'13]) don't appear computable in log-depth

Theorem (This work?): Assuming k-LIN, there exists log-depth weak PRF

Proof Sketch: Let

$$F(\mathbf{k}, \mathbf{x}) = \prod_{i=1}^{k} x_i^{k_i} \qquad \mathbf{k} \leftarrow \mathbb{Z}_p^k, \mathbf{x} \in \mathbb{F}^k$$

By repeated squaring, evaluation is just one big iterated multiplication over field. Security proof under k-LIN straightforward.

Corollary: Assuming k-LIN on pairings over elliptic curve, there exists optimal traitor tracing

Open Questions

1

All optimal traitor tracing schemes make *non-black box* use of cryptography (including ours). What is the best black-box pairing-based construction?

2

Tracing in our construction (and also optimal LWE-based TT) requires the master secret key. What can be achieved with pairings for *public tracing*?

3

Strong PRFs in log-depth from elliptic curves without pairings?
Strong PRFs in log-depth from k-LIN?