Cryptography against
CONTINUAL MEMORY LEAKAGE

Zvika Brakerski
(Weizmann)

Yael Kalai
(Microsoft Research)

Jonathan Katz
(Maryland)

Vinod Vaikuntanathan
(IBM T J Watson)

Secrets

Information accessible to one party and not to other(s)

essential to cryptography!

Secrets

Information accessible to one party and not to other(s)

essential to cryptography!

Theory

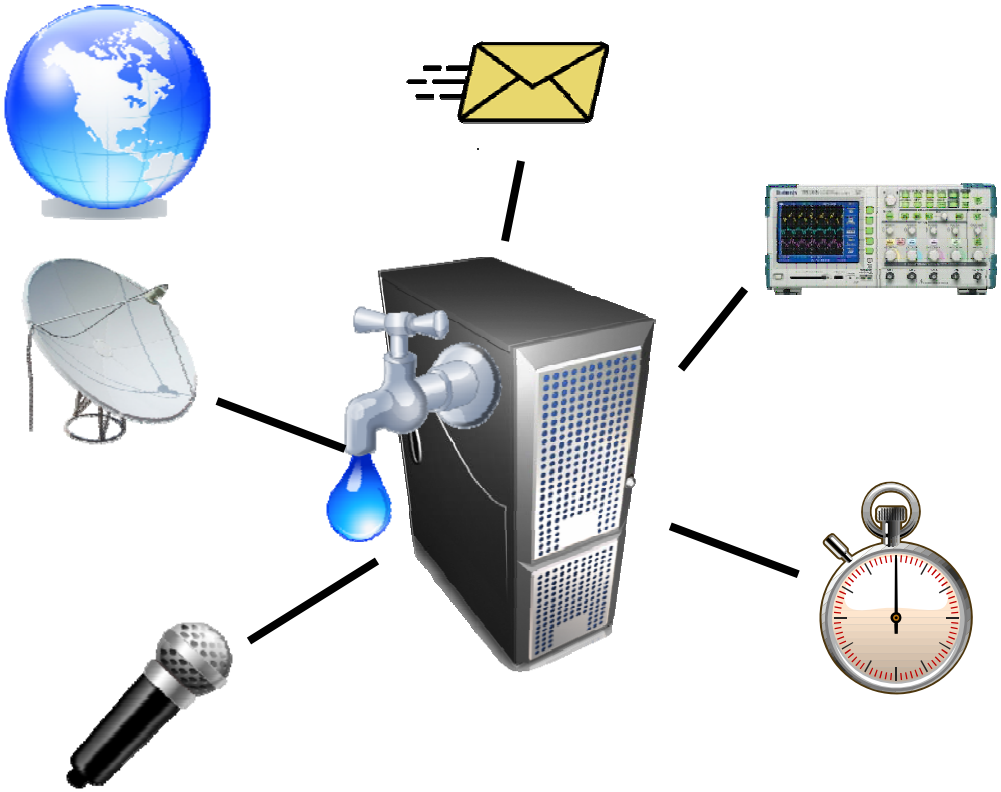
Secrets

Information accessible to one party and not to other(s)

essential to cryptography!

Theory

Real life



Secrets

Information accessible to one party and not to other(s)

essential to cryptography!

Theory

Real life

Secrets leak!

Secrets Leak

- From **EVERYWHERE**
  - HD, RAM, cache, registers, randomness…
- **ALL THE TIME**
  - Not necessarily a one time process.
  - No “protected” times.

Secrets Leak

- From **EVERYWHERE**
  - HD, RAM, cache, registers, randomness…
- **ALL THE TIME**
  - Not necessarily a one time process.
  - No “protected” times.

=

Continual Memory Leakage

Previous Leakage Models

“Only Computation Leaks”

[Micali-Reyzin’04]

“Memory Leakage”

[Akavia-Goldwasser-V’09] inspired by cold-boot attacks [HSH+08]

Previous Leakage Models

“Only Computation Leaks”

[Micali-Reyzin’04]

[DP08,P09,FKPR10,JV10,GR10,KP10]

“Memory Leakage”

[Akavia-Goldwasser-V’09] inspired by cold-boot attacks [HSH+08]

[AGV09,NS09,ADW09,KV09,…]

Previous Leakage Models

“Only Computation Leaks”

[Micali-Reyzin’04]

[DP08,P09,FKPR10,JV10,GR10,KP10]

Assumes a form of secure memory

Which does NOT leak as long as no computation is done on the data

“Memory Leakage”

[Akavia-Goldwasser-V’09] inspired by cold-boot attacks [HSH+08]

[AGV09,NS09,ADW09,KV09,…]

Previous Leakage Models

“Only Computation Leaks”

[Micali-Reyzin’04]

[DP08,P09,FKPR10,JV10,GR10,KP10]

Assumes a form of secure memory

Which does NOT leak as long as no computation is done on the data

“Memory Leakage”

[Akavia-Goldwasser-V’09] inspired by cold-boot attacks [HSH+08]

[AGV09,NS09,ADW09,KV09,…]

Everything leaks

… but only ONE-SHOT

Previous Leakage Models

“Only Computation Leaks”	**“Memory Leakage”**
[Micali-Reyzin’04]	[Akavia-Goldwasser-V’09] inspired by cold-boot attacks [HSH+08]
[DP08,P09,FKPR10,JV10,GR10,KP10]	[AGV09,NS09,ADW09,KV09,…]
Assumes a form of secure memory	Everything leaks
Which does NOT leak as long as no computation is done on the data	… but only ONE-SHOT

+ Work on restricted leakage functions [Riv,CDHKS,ISW,FRRTV]

Previous Leakage Models

“Only Computation Leaks”

[Micali-Reyzin’04]

[DP08,P09,FKPR10,JV10,GR10,KP10]

Assumes a form of secure memory

Which does NOT leak as long as no computation is done on the data

“Memory Leakage”

[Akavia-Goldwasser-V’09] inspired by cold-boot attacks [HSH+08]

[AGV09,NS09,ADW09,KV09,…]

Everything leaks

… but only ONE-SHOT

This work:

remove these restrictions!

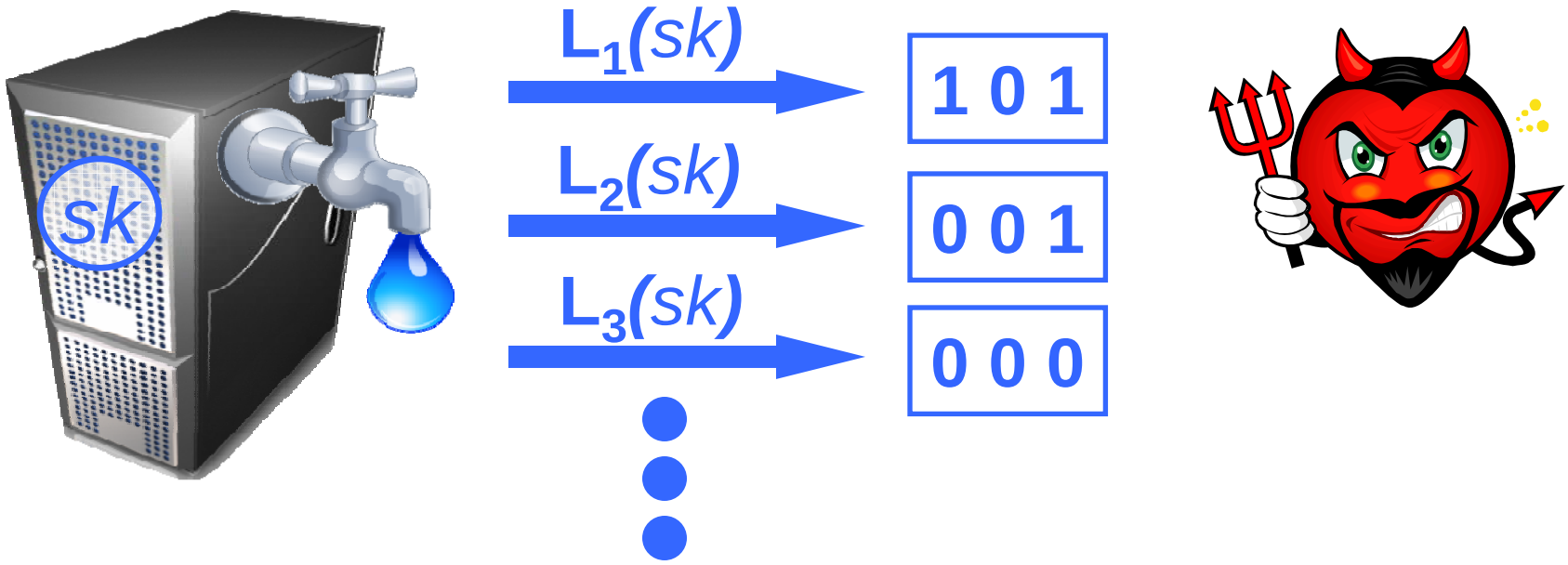
Our Model: Continual Memory Leakage

Our Model: Continual Memory Leakage

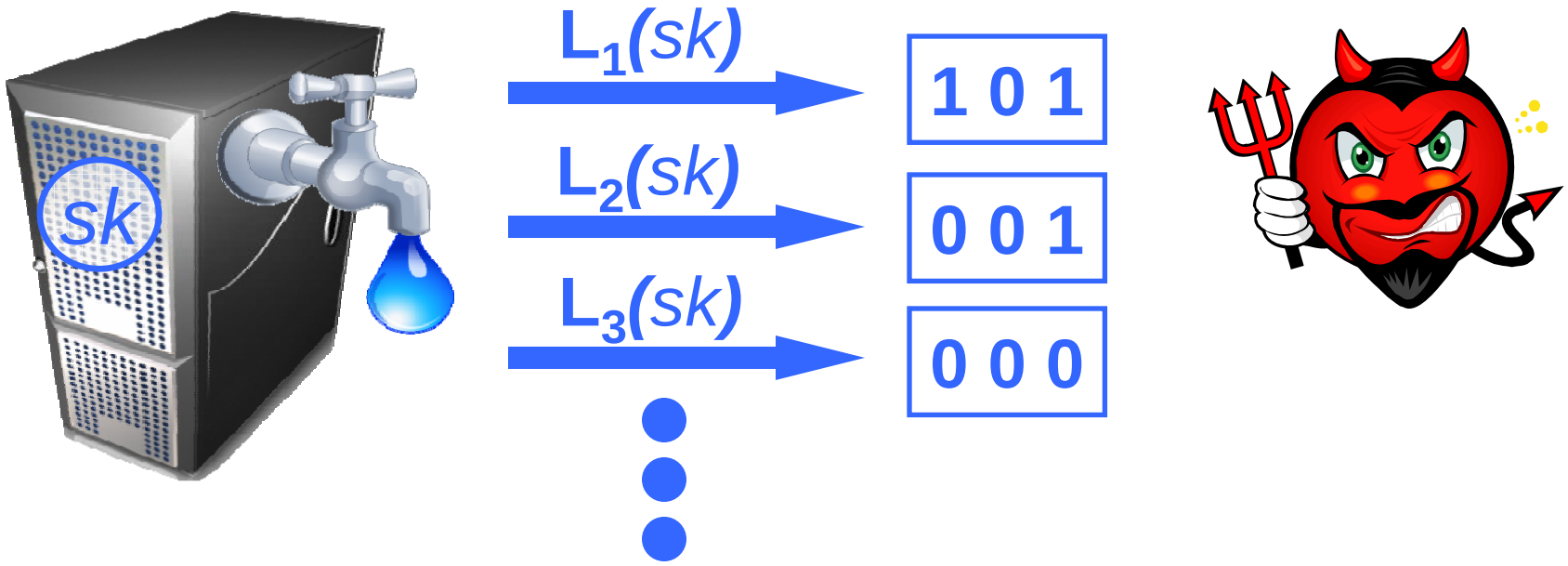
$L_1(sk)$

1 0 1

Our Model: Continual Memory Leakage

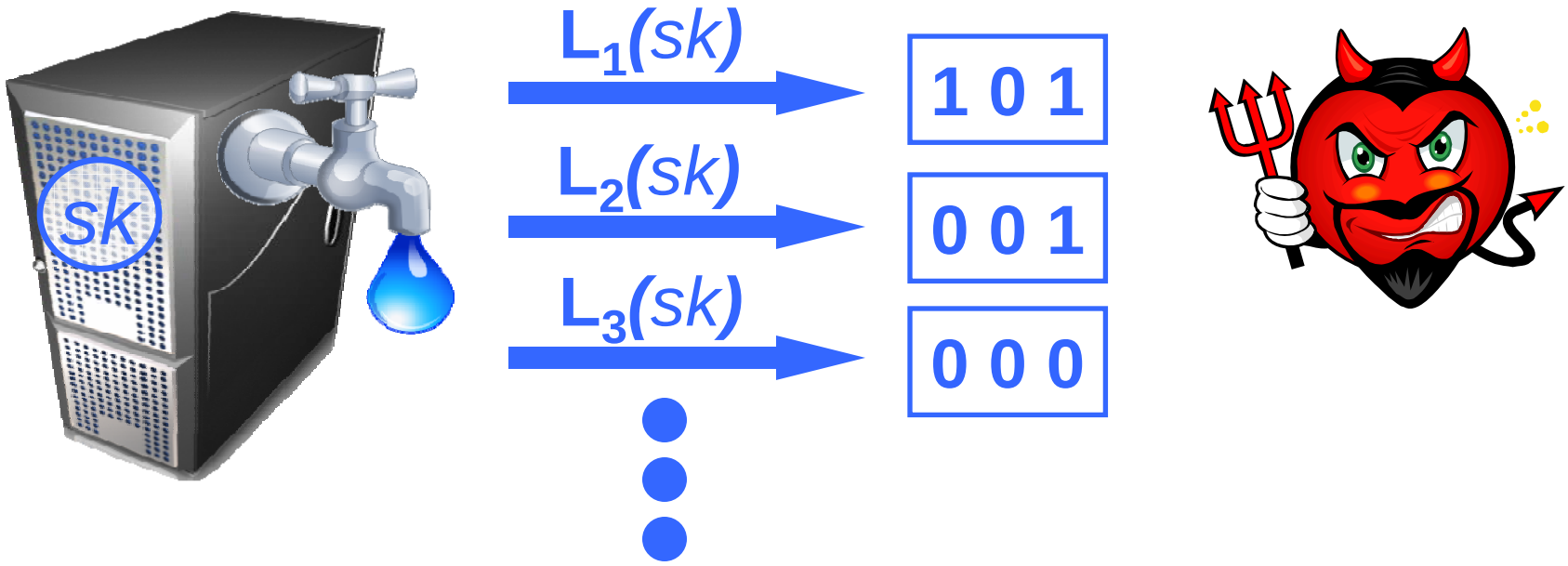


Our Model: Continual Memory Leakage



Can we protect against continual leakage?

Our Model: Continual Memory Leakage



Can we protect against continual leakage?

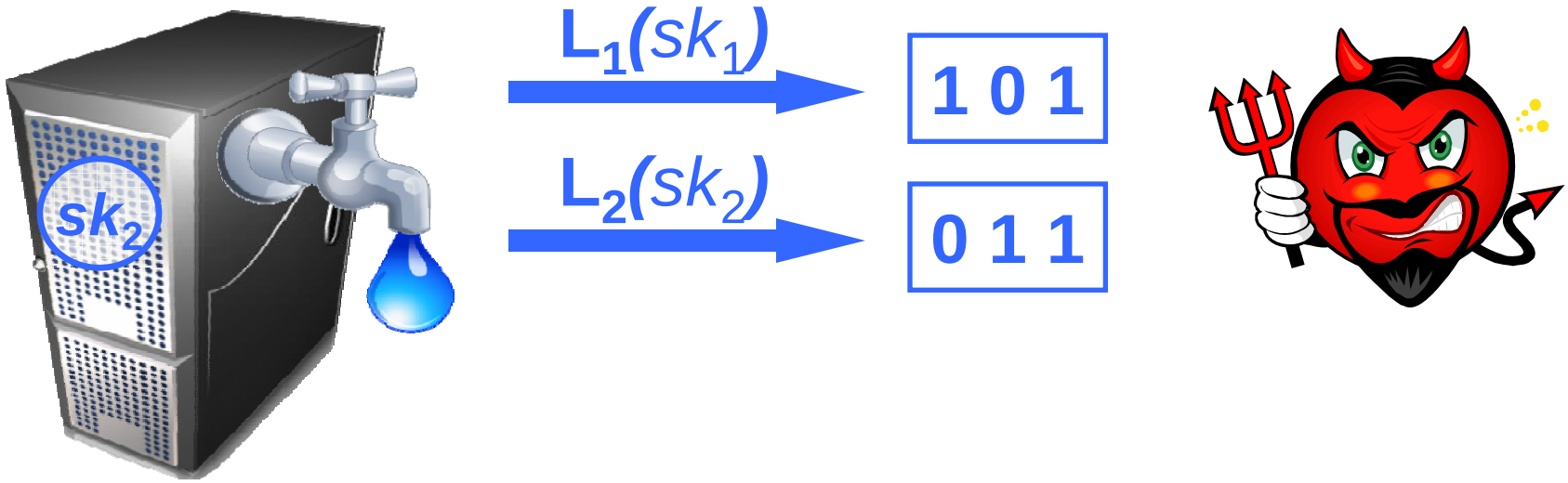
Need to (periodically) update ***SK***.

Our Model: Continual Memory Leakage

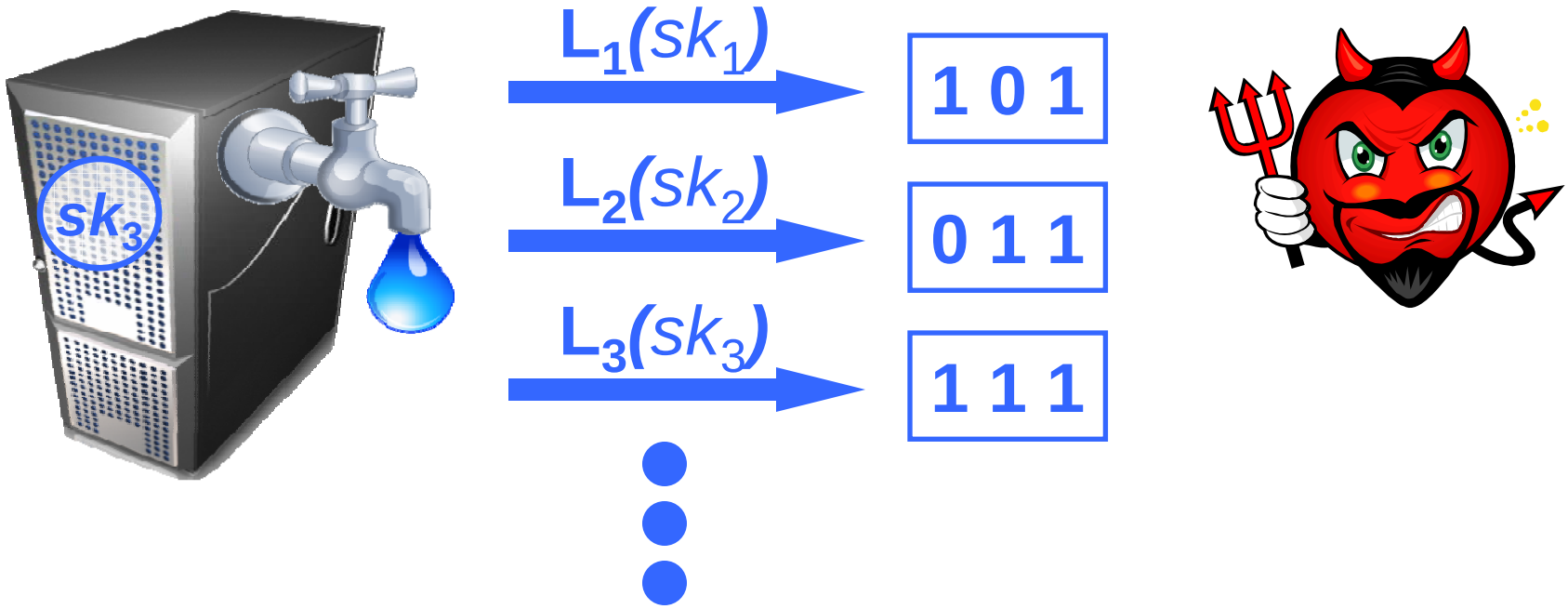
$L_1(sk_1)$

1 0 1

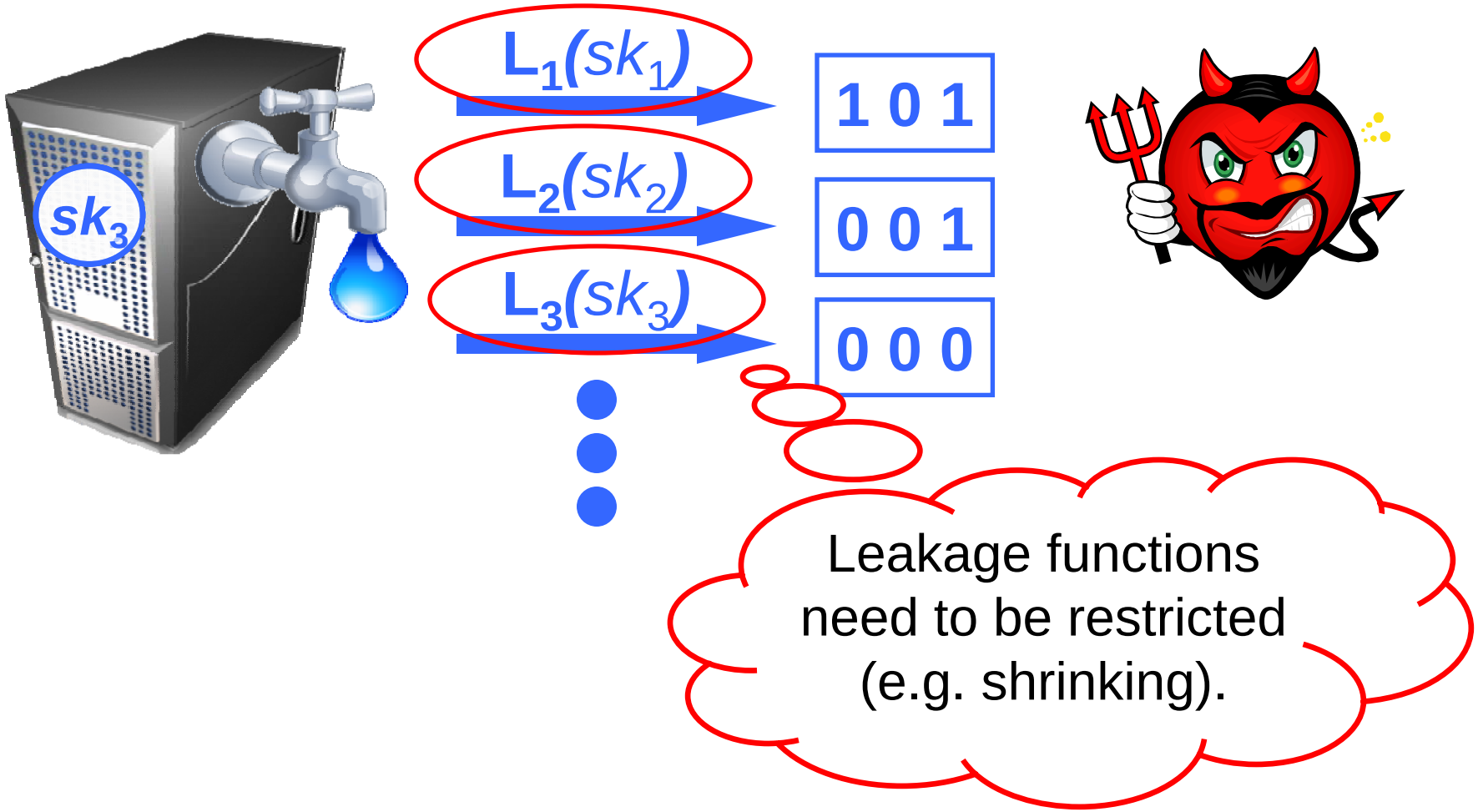
Our Model: Continual Memory Leakage



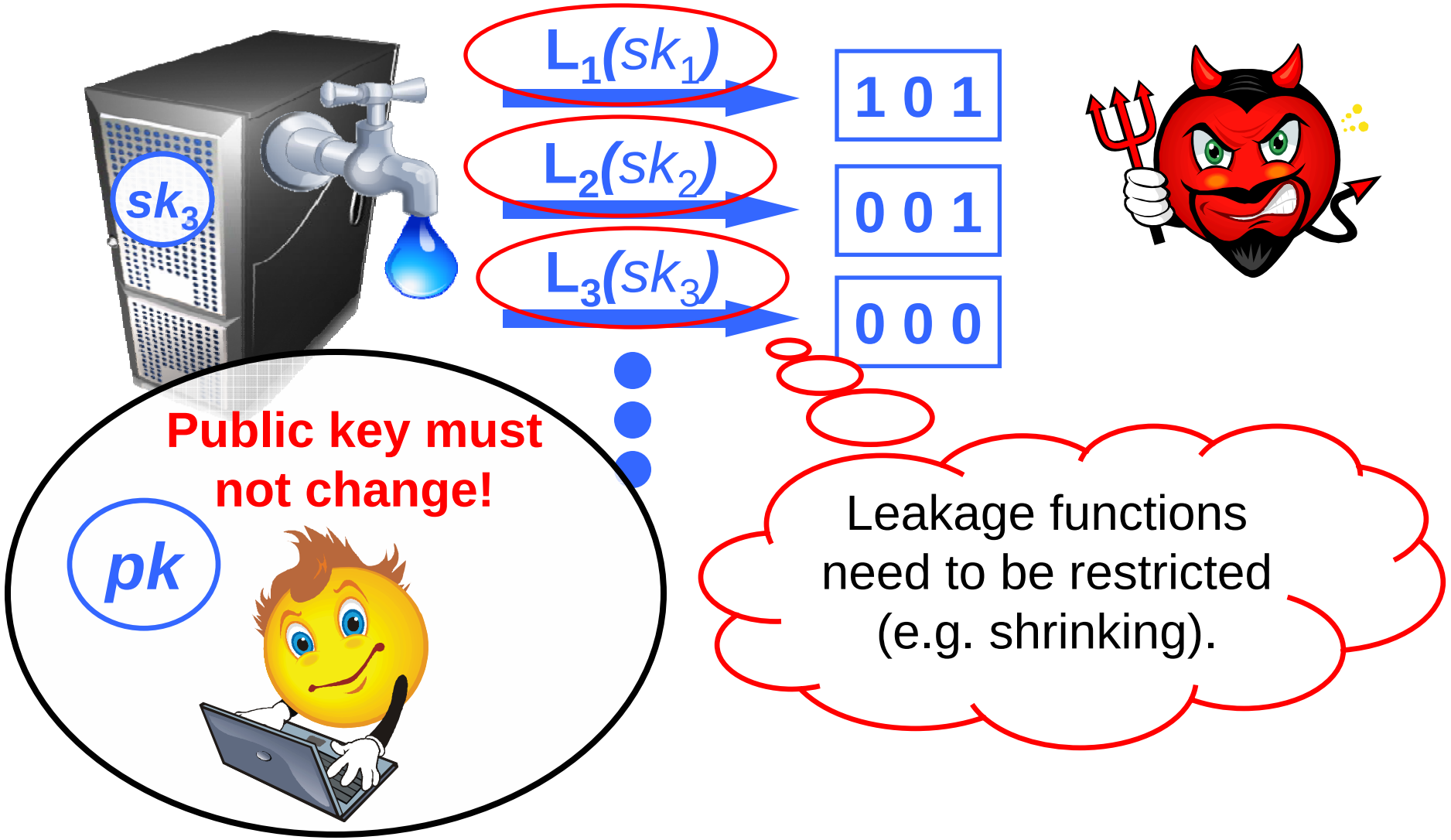
Our Model: Continual Memory Leakage



Our Model: Continual Memory Leakage



Our Model: Continual Memory Leakage



Features of Continual Memory Leakage (CML)

- Leakage function applied to **entire** memory
- Key updates are oblivious to users (public key doesn't change).
- Leakage can occur at any time point!
  - Including during key updates.
  - Including during decrypting/signing.
- Total amount of leakage – **unbounded**.
  - Only the rate (leaked bits/sec) is bounded.

Our Results: Cryptography in the CML Model

- **Public-key encryption.**
  - **Linear assumption:** leakage rate (1/2-o(1)).
  - **SXDH assumption:** leakage rate (1-o(1)).
  - **KeyGen and Update:** logarithmic no. of bits.
- Identity based encryption.
- Encryption $\Rightarrow$ Signatures.

Our Results: Cryptography in the CML Model

- **Public-key encryption.**
  - **Linear assumption** kage rate (1/2-o(1)).
  - **SXDH assumption** ate (1-o(1)).
  - **KeyGen and U**
- Iden
- Encryption ⇒ Signa

Previously not known even in the Micali-Reyzin model!

(under standard assumptions)

Our Results: Cryptography in the CML Model

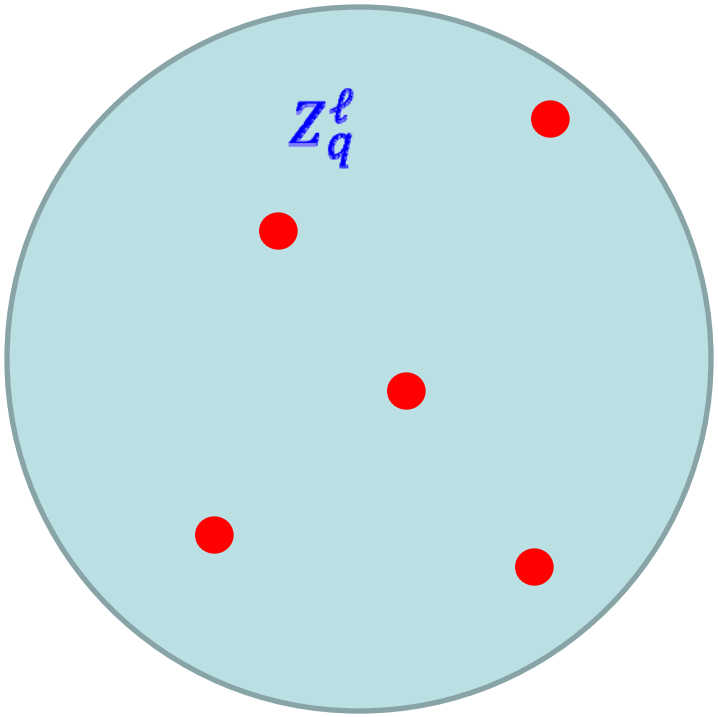
- **Public-key encryption.**
  - **Linear assumption:** leakage rate (1/2-o(1)).
  - **SXDH assumption:** leakage rate (1-o(1)).
  - **KeyGen and Update:** logarithmic no. of bits.
- Identity based encryption.
- Encryption $\Rightarrow$ Signatures.

Concurrently [DHLW]: *efficient* signatures, ID schemes and AKA in the CML model under linear and SXDH.

- Different techniques (re-randomizable NIZK).

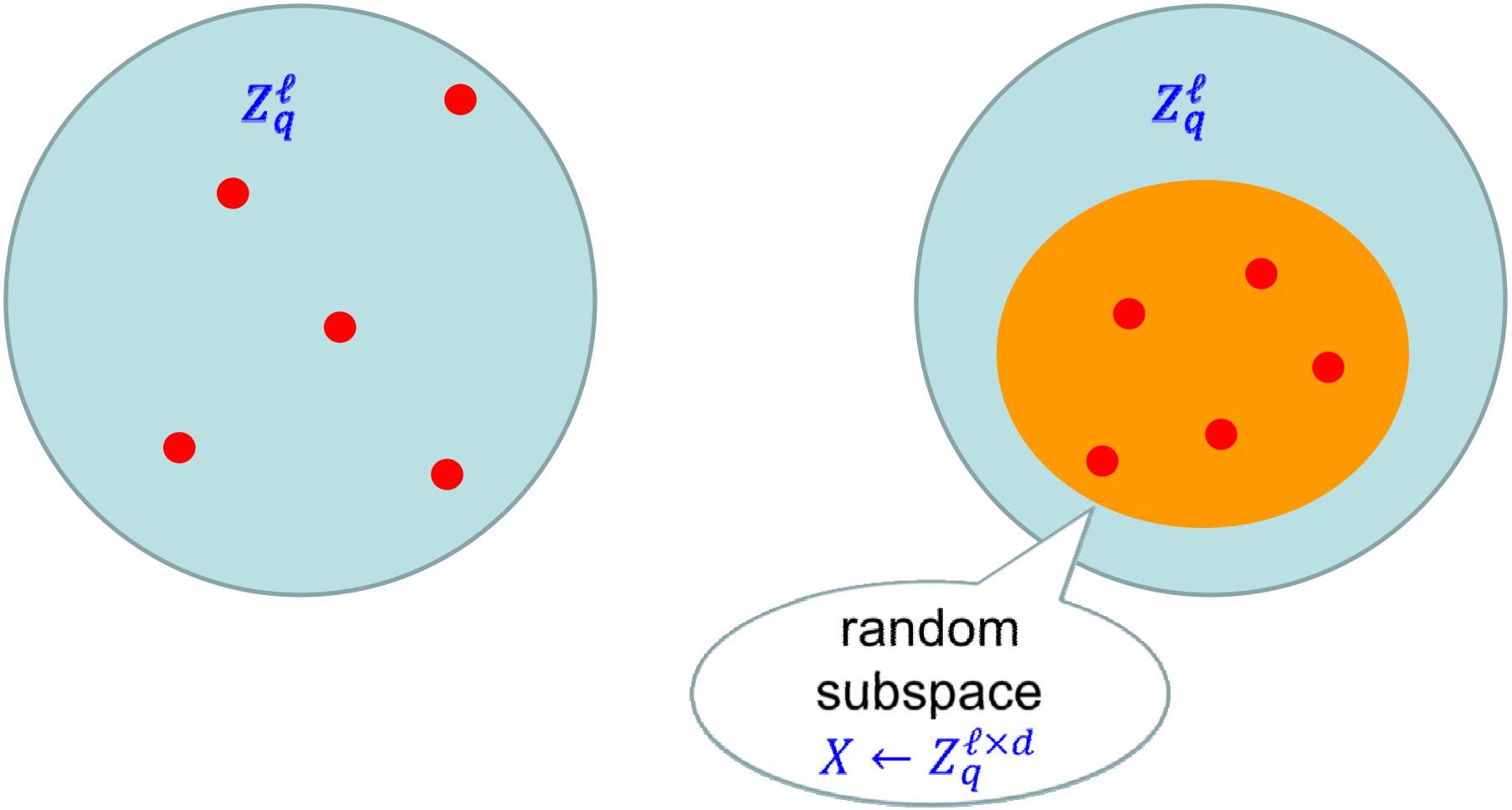
A Sneak Peek

Lemma: Random Subspaces are Leakage-Resilient



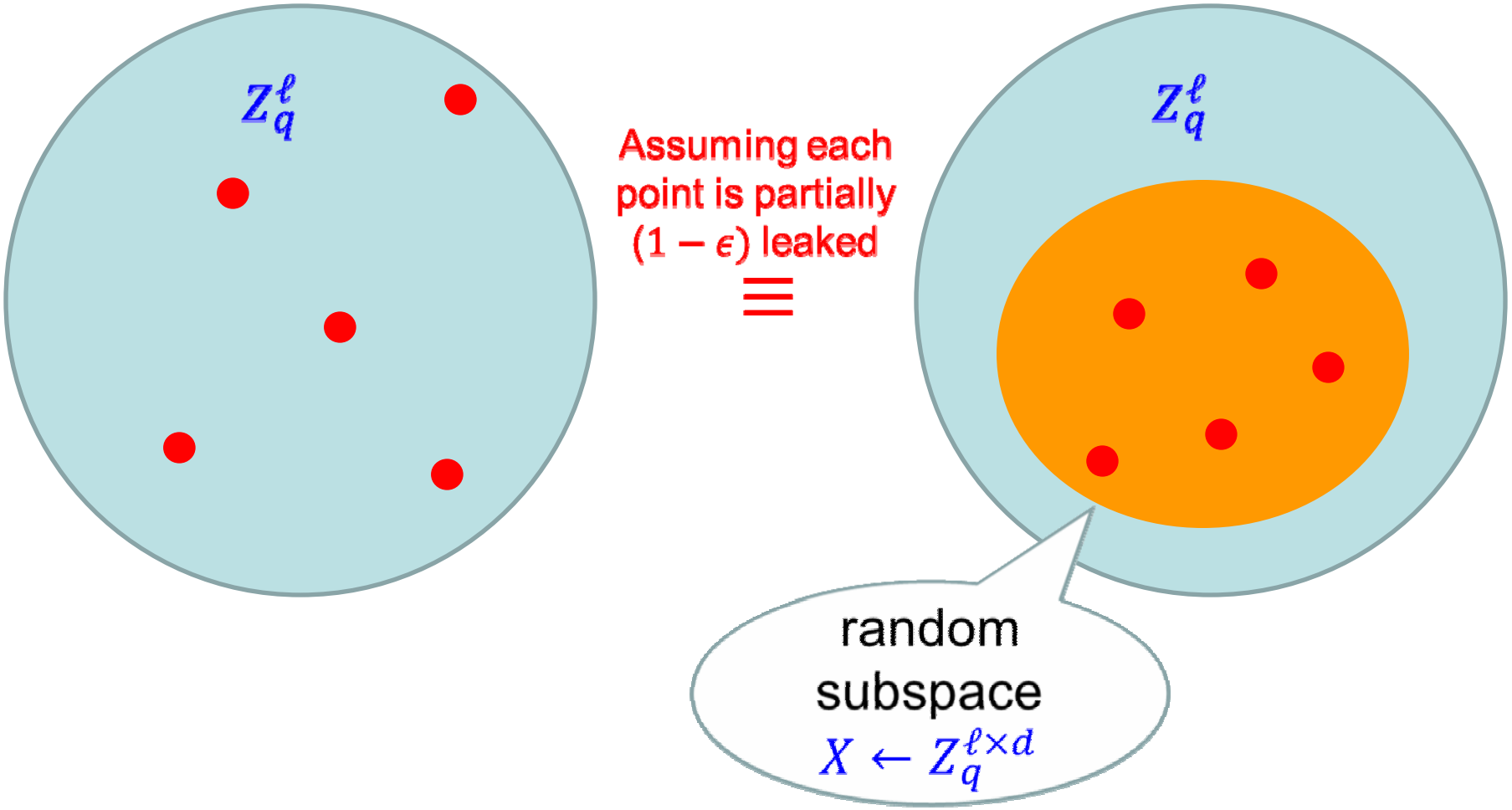
A Sneak Peek

Lemma: Random Subspaces are Leakage-Resilient



A Sneak Peek

Lemma: Random Subspaces are Leakage-Resilient



Paper on Eprint:

“Cryptography against Continual Memory Leakage”, http://eprint.iacr.org/2010/278.