On Perfect and Adaptive Security in Exposure-Resilient Cryptography

Yevgeniy Dodis, New York University

Amit Sahai, Princeton

Adam Smith, MIT

Problem: Partial Key Exposure

- Alice needs to store a cryptographic key
- She wants to store her key on a hard drive
- Eve may break in and access some limited number of bits
- How can Alice store her key so it remains secure?

Problem: Partial Key Exposure

- Standard cryptography:
  - No guarantees even if only tiny fraction of key is leaked

- Paradigm: "Exposure-Resilient Cryptography"
  - Build primitives that remain secure even when most of the key is leaked.

“Exposure-Resilient Cryptography”

- Two main primitives:
  - All-or-Nothing Transforms (AONT)
  - Exposure-Resilient Functions (ERF)

- This talk focuses on AONT.

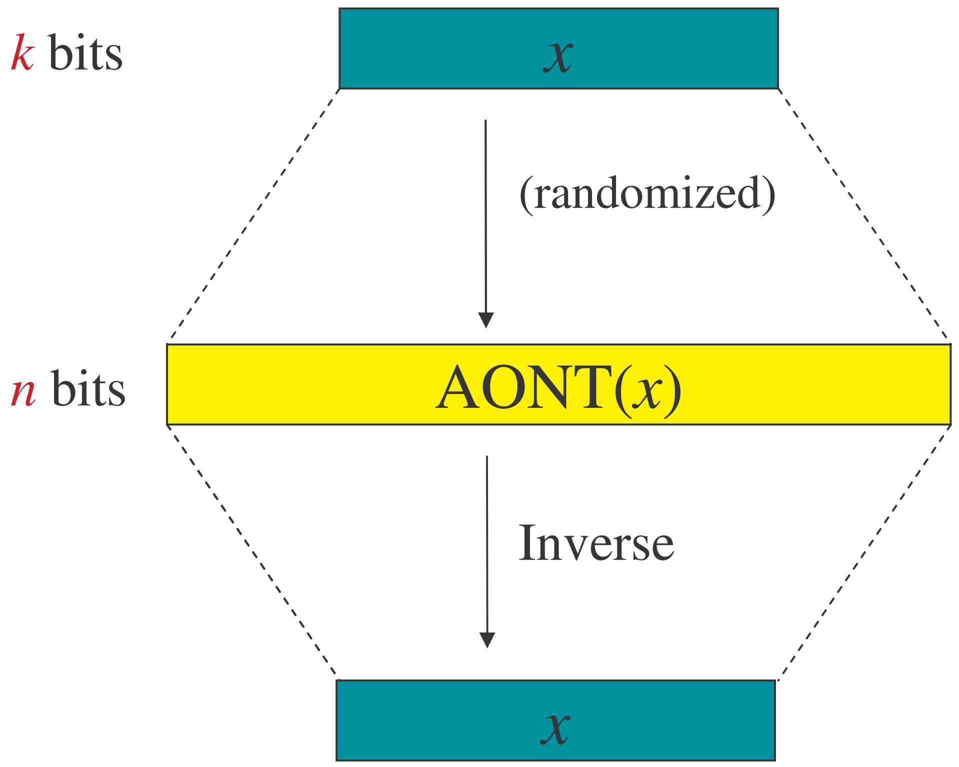
- ERF also discussed in the paper.

All-or-Nothing Transform (AONT)

- Randomized encoding
- No key
- Increases length

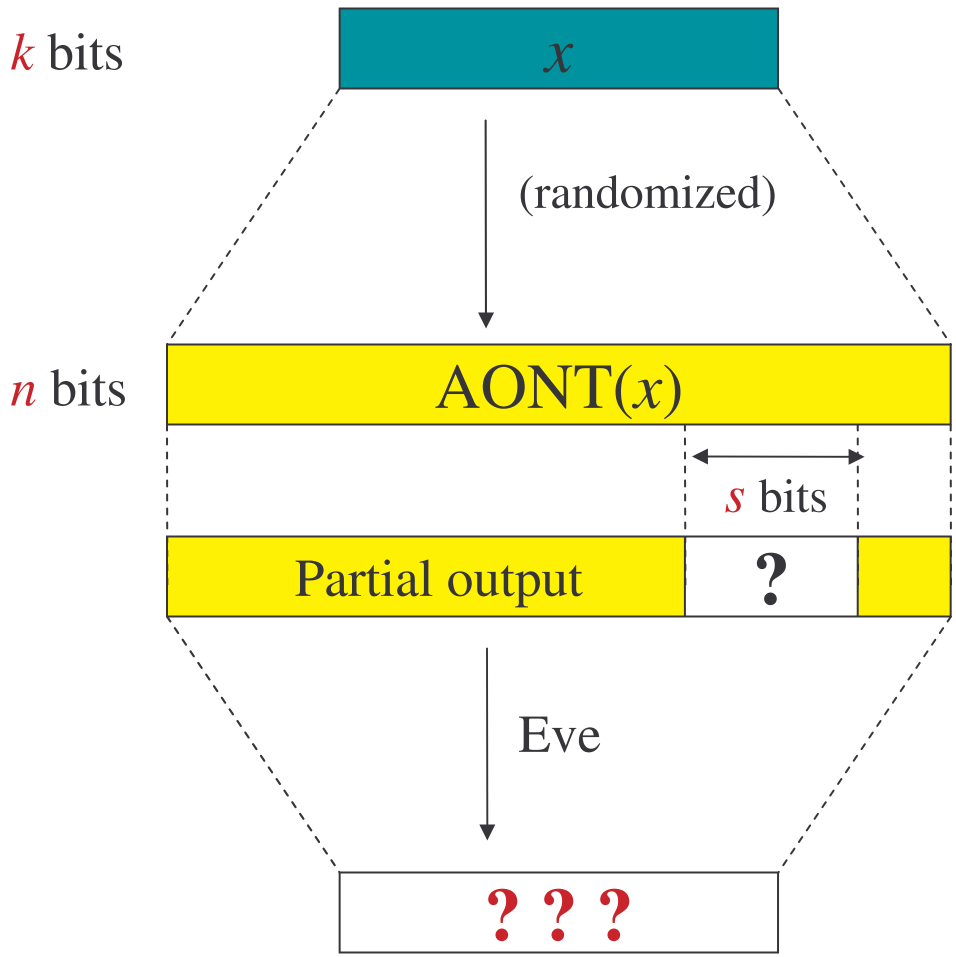
- If you know:
  - All of the output……………. Recover the whole input
  - Part of the output………..…. No information about input

All-or-Nothing Transform (AONT)



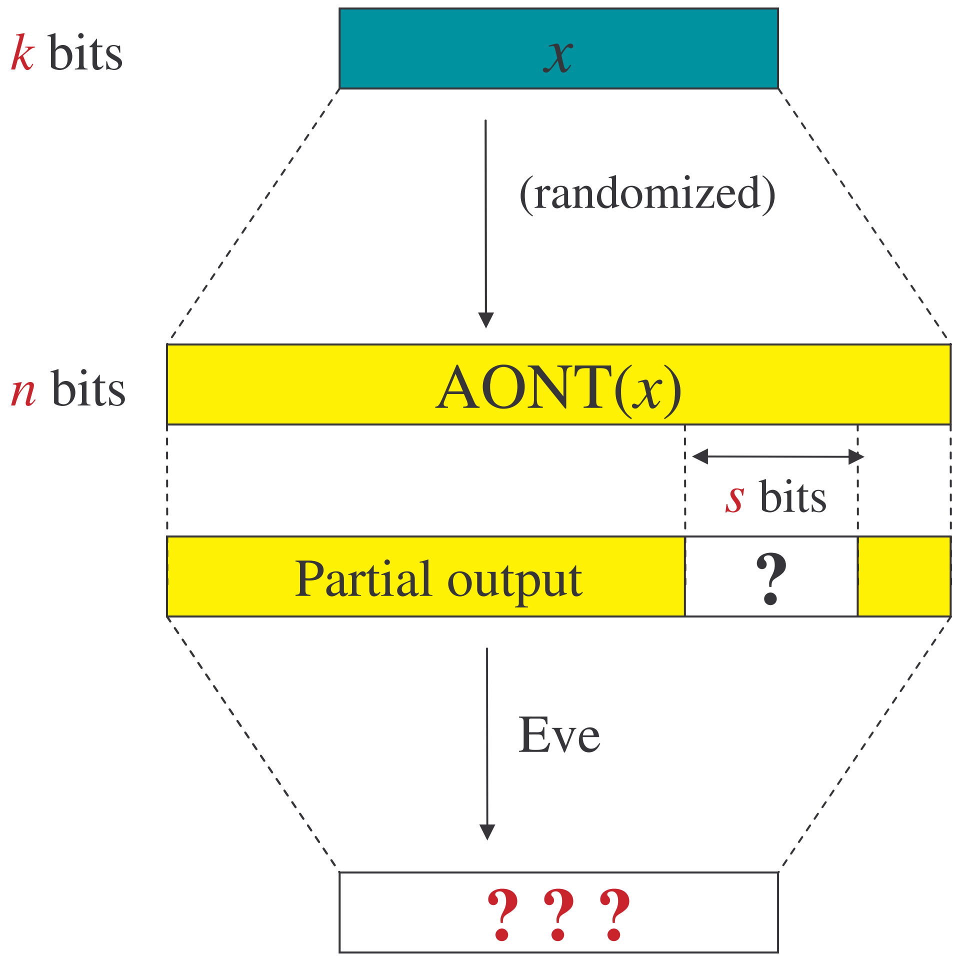
- Randomized encoding
- With all of the output can recover input

All-or-Nothing Transform (AONT)



- Randomized encoding
- With all of the output can recover input
- If missing s bits of output, No Information about input

All-or-Nothing Transform (AONT)



Special case of

"Ramp Secret Sharing"

– Shares are 1 bit each

– All n shares

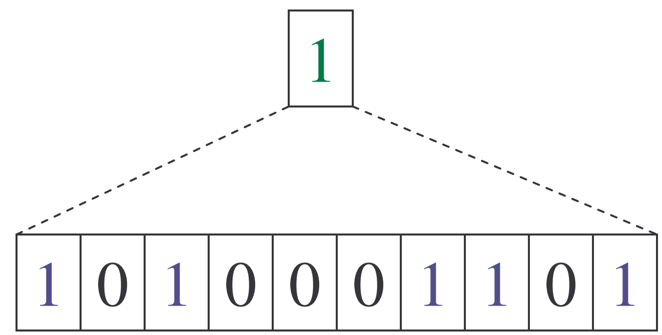
$\Rightarrow$ find x

– $\leq n\text{-}s$ shares

$\Rightarrow$ no information

Example of AONT

- Simplest example: parity
- Encode a single bit



- AONT(b) = random string y such that PARITY(y)=b
  - Given all of y, computing b is easy
  - If Eve misses any bit of y, then no information on b
  - $k = 1$
  - $s = 1$

Problem: Partial Key Exposure

- Alice needs to store a cryptographic key (say for signing or for encryption)
- She wants to store her key on a hard drive
- Eve may break in and access some limited number of bits on the hard drive.
- How can Alice store her key so it remains secure?

Solution: All-or-Nothing Transform

Alice stores encoding *AONT*(x)

instead of her key x

- Other applications:
  - Introduced by Rivest for strengthening block ciphers
  - Many recent works with constructions and applications:

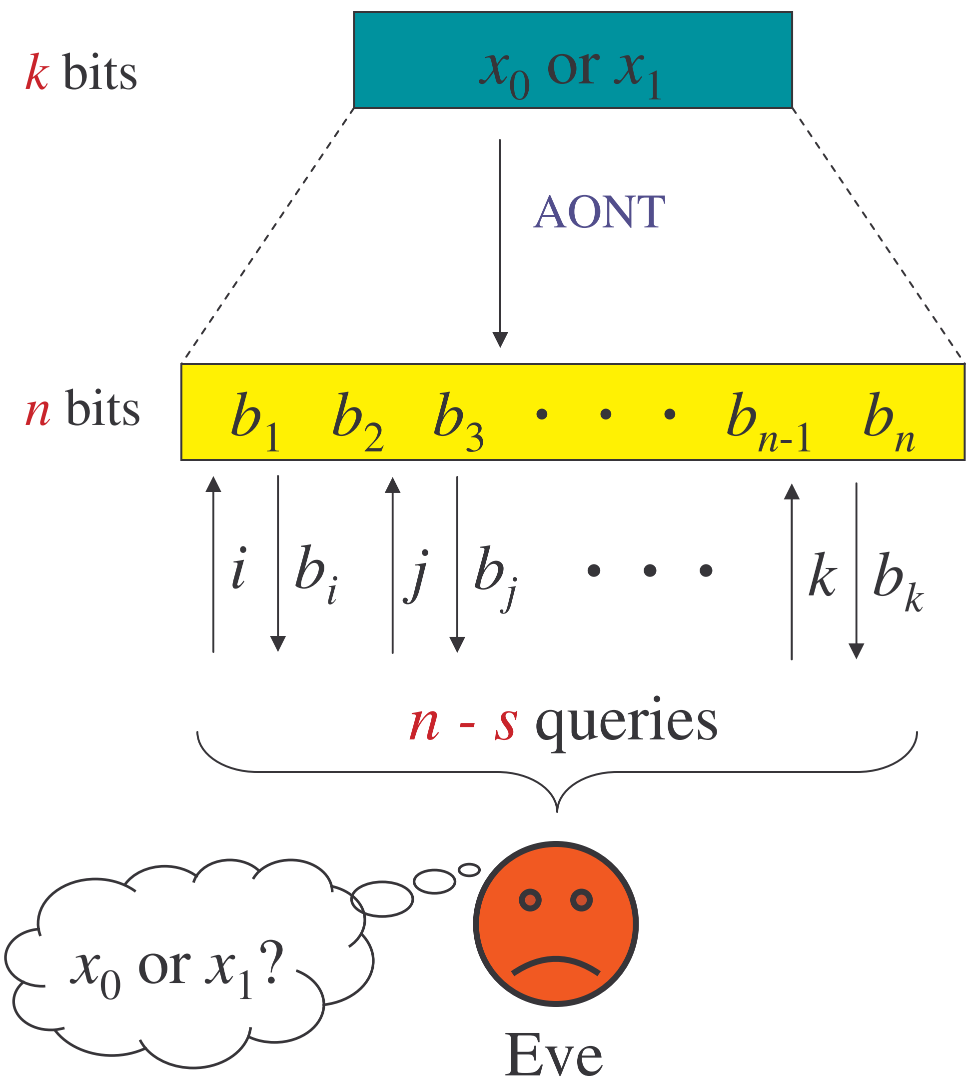
    Rivest '97, Boyko '99, Desai '00, Matyas et al. '96, Jackobson et al. '99, Blaze '96, Bellare-Boldyreva '00, Shin-Rhee '99, Canetti et al. '00, ...

This work: Adaptive Security

- In many situations, Eve might:
  - Learn bits one at a time
  - Choose which bit to learn next based on
    - what she has seen so far
    - any partial information she has about the input
- Questions:
  - How to define adaptive security?
  - Are previous constructions secure ?
  - How well can we do against adaptive adversaries?

    (i.e. what parameters can we achieve?)

How to define adaptive security?



- Similar to semantic security for cryptosystems
- Secret is one of two possibilities

  x_0 or x_1

- Eve queries output one bit at a time
- Eve tries to guess whether she saw AONT(x_0) or AONT(x_1)
- Success probability should be $\approx 1/2$

Strengths of secrecy

- Perfect:
  - secret independent of adversary's view
  - i.e. distribution on views the same for any two secrets

Success prob. = 1/2

- Statistical:
  - distributions on views are statistically close for any two secrets

Success prob. = 1/2 ± ε

- Computational:
  - distributions on views are computationally indistinguishable for any two secrets

Success prob. = 1/2 ± ε
when Eve is poly-time

This work: Adaptive Security

- In many situations, Eve might:
  - Learn bits one at a time
  - Choose which bit to learn next based on
    - what she has seen so far
    - any partial information she has about the input
- Questions:
  - ✓ How to define adaptive security?
  - Are previous constructions secure ?
  - How well can we do against adaptive adversaries?

    (i.e. what parameters can we achieve?)

Are Previous Constructions Secure ?

- Previous constructions……………… under Adaptive adversaries

  – Perfect secrecy: Secure, but bad parameters

  – Statistical secrecy: Insecure [CDHKS '00]

  – Standard computational secrecy: Insecure [CDHKS '00]

  – Random oracle/cipher model: Secure, but must assume random oracle [Boyko '99, Rivest '97, Desai '00,...]

This work: Adaptive Security

- In many situations, Eve might:
  - Learn bits one at a time
  - Choose which bit to learn next based on
    - what she has seen so far
    - any partial information she has about the input
- Questions:
  - ✓ How to define adaptive security?
  - ✓ Are previous constructions secure ?
  - How well can we do against adaptive adversaries?
    (i.e. what parameters can we achieve?)

Main results

- Perfect secrecy:
  - Static security = adaptive security
  - New lower bound for AONT
    - Can’t reveal more than half of output! (when secret > log n bits)
- Statistical secrecy:
  - Previous constructions insecure against adaptive adversary
  - Simple near-optimal, probabilistic constructions of:
    - Almost-perfect resilient Functions (APRF)
    - All-or-Nothing Transform (AONT)
    - Exposure-resilient Functions (ERF)

    $s \approx k$
- Computational secrecy:
  - Combine statistical secrecy with pseudo-random generator [CDHKS ‘00]
  - (Almost) arbitrary parameters ($s << k$)

This work: Adaptive Security

- In many situations, Eve might:
  - Learn bits one at a time
  - Choose which bit to learn next based on
    - what she has seen so far
    - any partial information she has about the input
- Questions:
  - How to define adaptive security?
  - Are previous constructions secure ?
  - How well can we do against adaptive adversaries? (i.e. what parameters can we achieve?)
    - Lower bound for perfect secrecy
    - Near-optimal statistical constructions

Lower bound on perfect AONT

Lower bound on perfect AONT

- Recall AONT encodes k bits into n bits
  - Know all n bits of output $\Rightarrow$ Recover whole input
  - Know $n - s$ bits $\Rightarrow$ No information
- We show:

$$s \geq \frac{n}{2} + \left(1 - \frac{n}{2 \cdot (2^k - 1)}\right)$$

- In particular:

When $k > \log n$ we must hide at least 1/2 of the the output!

Lower bound on perfect AONT

- Main ideas:
  - View $\{0,1\}^n$ as a graph (a.k.a. "the hypercube")
  - Perfect AONT are "balanced, weighted colorings" of hypercube
  - Use Fourier analysis over graph $\{0,1\}^n$ (*i.e.* over group Z_2^n)
  - Details in the paper.

- Previously proven only for a related primitive:
  Resilient Functions [Friedman '92, Bierbrauer et al '96]

- Generalizes technique of Friedman '92
  - Previous bound is a special case

This work: Adaptive Security

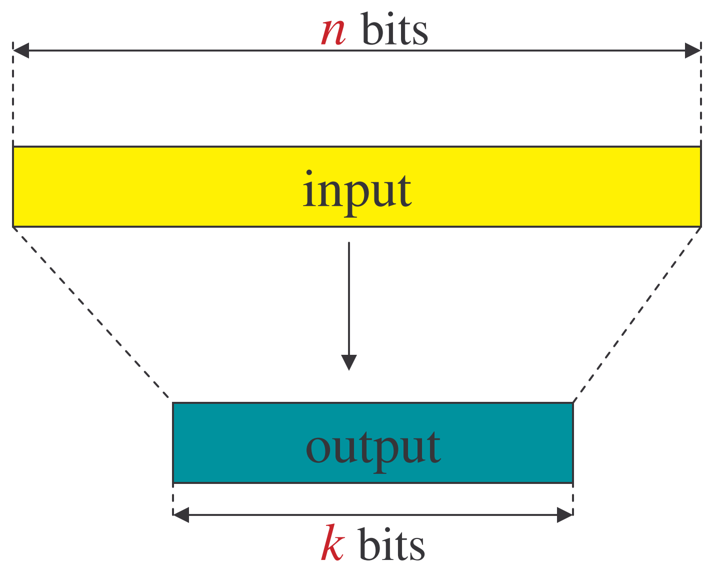
- In many situations, Eve might:
  - Learn bits one at a time
  - Choose which bit to learn next based on
    - what she has seen so far
    - any partial information she has about the input
- Questions:
  - How to define adaptive security?
  - Are previous constructions secure ?
  - How well can we do against adaptive adversaries?
    (i.e. what parameters can we achieve?)
    - Lower bound for perfect secrecy
    - Near-optimal statistical constructions

Statistical Security against adaptive adversaries

How to achieve adaptive statistical security

- Reduce adaptive security of AONT to a static property
- Main tool:
  - "Almost-perfect resilient functions" (APRF) [Kurosawa-Johansson-Stinson '97]
- Simple construction of APRF: Pick random hash function
  - Will be an APRF with high probability
  - Near-optimal parameters
- Good APRF's imply good AONT

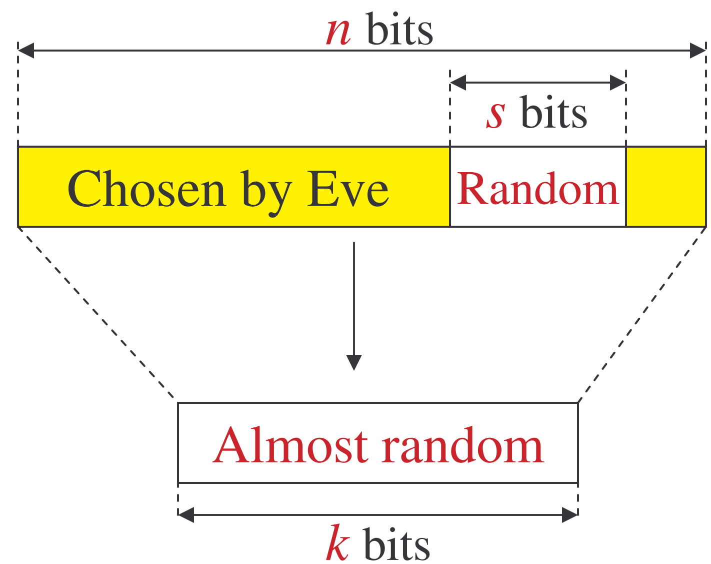
Main tool: Resilient Functions



- Functions from n bits to k bits
- Intuition:
  - Choosing all but s bits of input gives no information on output.
- Various formalizations of this in statistical setting (see paper).
- Strongest notion is APRF ("almost-perfect resilient function")
  - Eve can fix n-s bits of input
  - Remaining s bits chosen at random
  - Distribution on output still close to uniform on all points

Main tool: Resilient Functions

- Functions from n bits to k bits



- Intuition:
  - Choosing all but s bits of input gives no information on output.
- Various formalizations of this in statistical setting (see paper).
- Strongest notion is APRF ("almost-perfect resilient function")
  - Eve can fix n-s bits of input
  - Remaining s bits chosen at random
  - Distribution on output still close to uniform on all points

How to construct APRF?

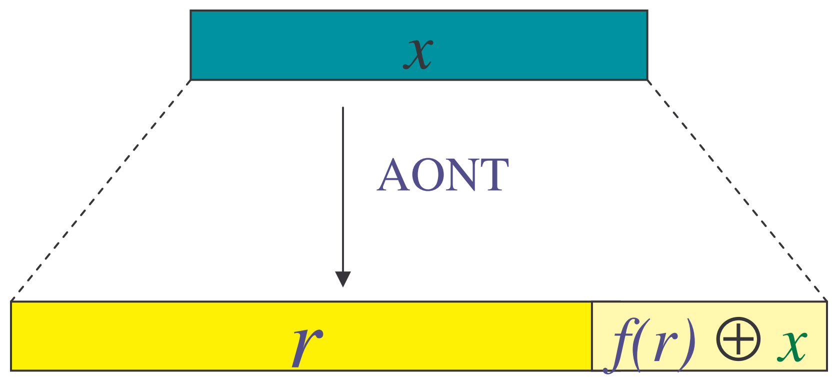
- Kurosawa et al. '97:
  - Coding-theoretic construction
  - Achieves $s \geq \frac{n+k}{2}$
  - Very far from trivial bound of $s \geq k$

- Intuition: A random function is an APRF with high probability (proof by Chernoff bounds)
- Construction: Pick a random (n/log n)-wise independent hash function
  - $s = k + 2\log(1/\varepsilon) + O(\log n)$ with high probability

    $= k + o(k)$ when $k >> \log n$
  - Proof relies on tail bounds on n-wise independent variables
  - Also used in Trevisan-Vadhan'00 for constructing deterministic extractors

APRF ⇒ AONT: One-time pad

- Say f is an APRF with output length n
- AONT(x) = (r , $f(r) \oplus x$) for random string r
  - $n' = n + k$
  - $s' = s$
  - $k' = k$



- Still get $s' \approx k'$
- Slight expansion of output
- Note: If f is efficiently invertible, no pad is necessary

This work: Adaptive Security

- In many situations, Eve might:
  - Learn bits one at a time
  - Choose which bit to learn next based on
    - what she has seen so far
    - any partial information she has about the input
- Questions:
  - ☑ How to define adaptive security?
  - ☑ Are previous constructions secure ?
  - How well can we do against adaptive adversaries?
    (i.e. what parameters can we achieve?)
    - ☑ Lower bound for perfect secrecy
    - ☑ Near-optimal statistical constructions

Conclusions

- Considered adaptive security for exposure-resilient primitives
- Perfect setting
  - New lower bound for AONT, matches previous bounds for Exposure-Resilient Functions
- Statistical setting
  - Reduce to construction of Almost-Perfect Resilient Functions
  - Near-optimal constructions of APRF, ERF, AONT