Stony Brook University

† UC SANTA CRUZ BaskinEngineering

Center for Research in Systems and Storage

Secure Archival is Hard… *Really* Hard

Christopher Smith, Maliha Tabassum, Soumya Chowdary Daruru, Gaurav Kulhare, Arvin Wang, Ethan L. Miller*†, Erez Zadok

Stony Brook University, PureStorage, UC Santa Cruz†*

* PURESTORAGE®

NSF

The Problem at a Glance

How to keep long-term archival data **secure**?

- Confidentiality
- Integrity
- Availability

O(100yrs)

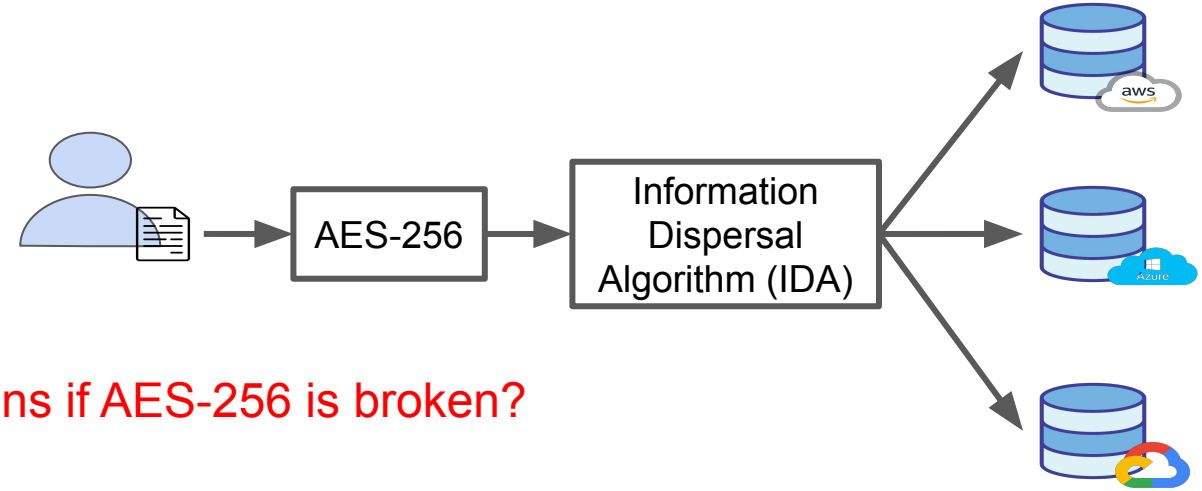
Use Cases

Challenges

- Cryptographic Obsolescence
- Harvest Now, Decrypt Later
- Storage cost
- Side-channel attacks
- And more…

PURESTORAGE® UC SANTA CRUZ BaskinEngineering Center for Research in Systems and Storage Stony Brook University

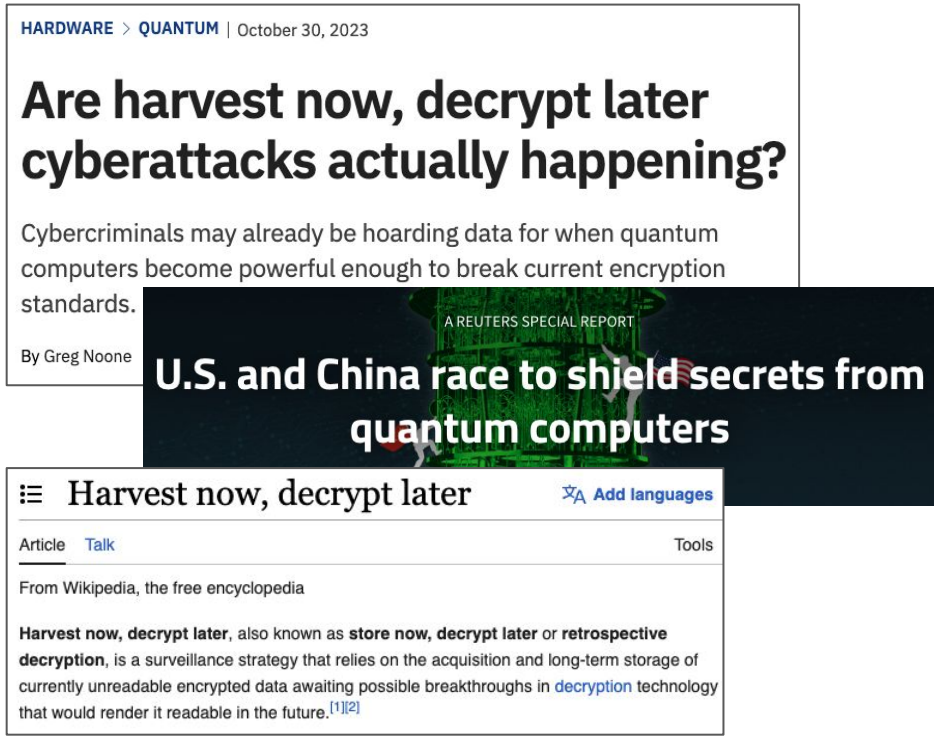
Warm-Up Attempt



What happens if AES-256 is broken?

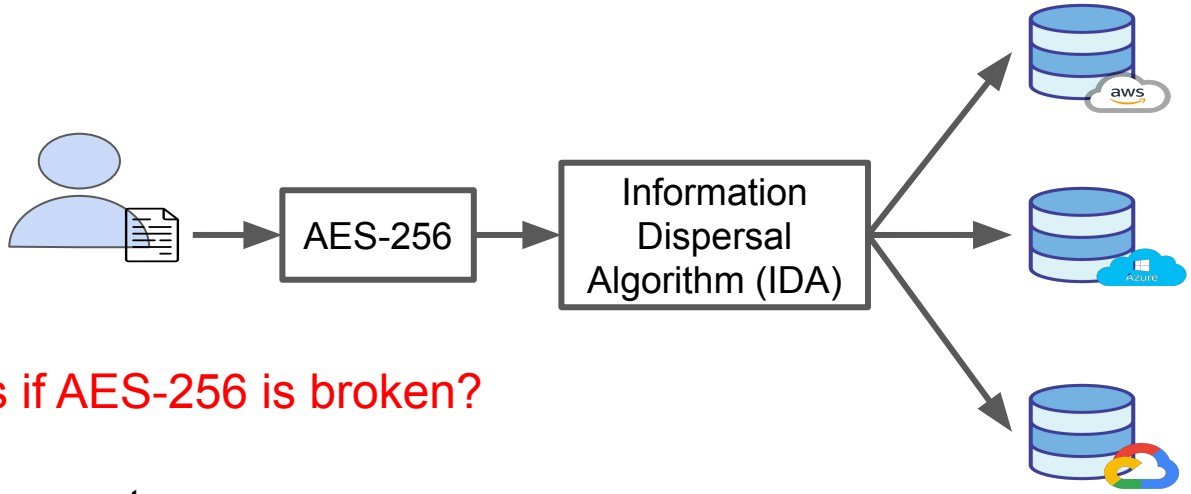
- Just re-encrypt…
  - What if encrypted data is already stolen? // i.e. Harvest Now, Decrypt Later

Harvest Now, Decrypt Later



Harvest Now, Decrypt Later (HNDL) attacks are happening now

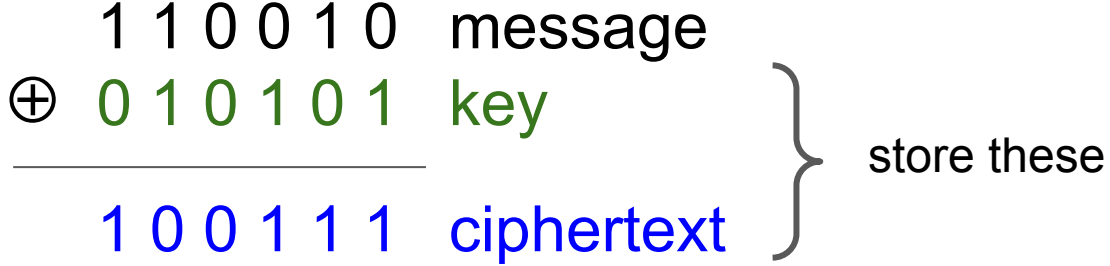
Warm-Up Attempt (cont.)



What happens if AES-256 is broken?

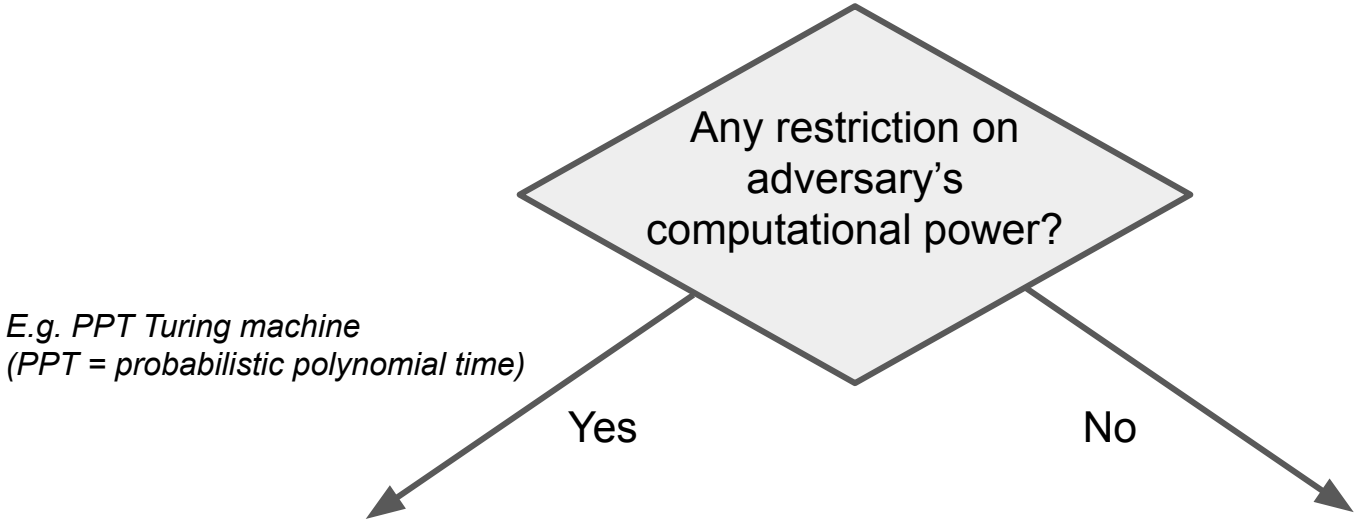
- Just re-encrypt…
  - What if encrypted data is already stolen? // i.e. Harvest Now, Decrypt Later
    - Use an unbreakable encryption scheme???
      - Where do I get one of those?

The One-Time Pad



- Reveals no information about the message (“perfect secrecy”)
- Key must be as long as the message
- Loss of key or ciphertext ⇒ loss of message

Two Security Notions



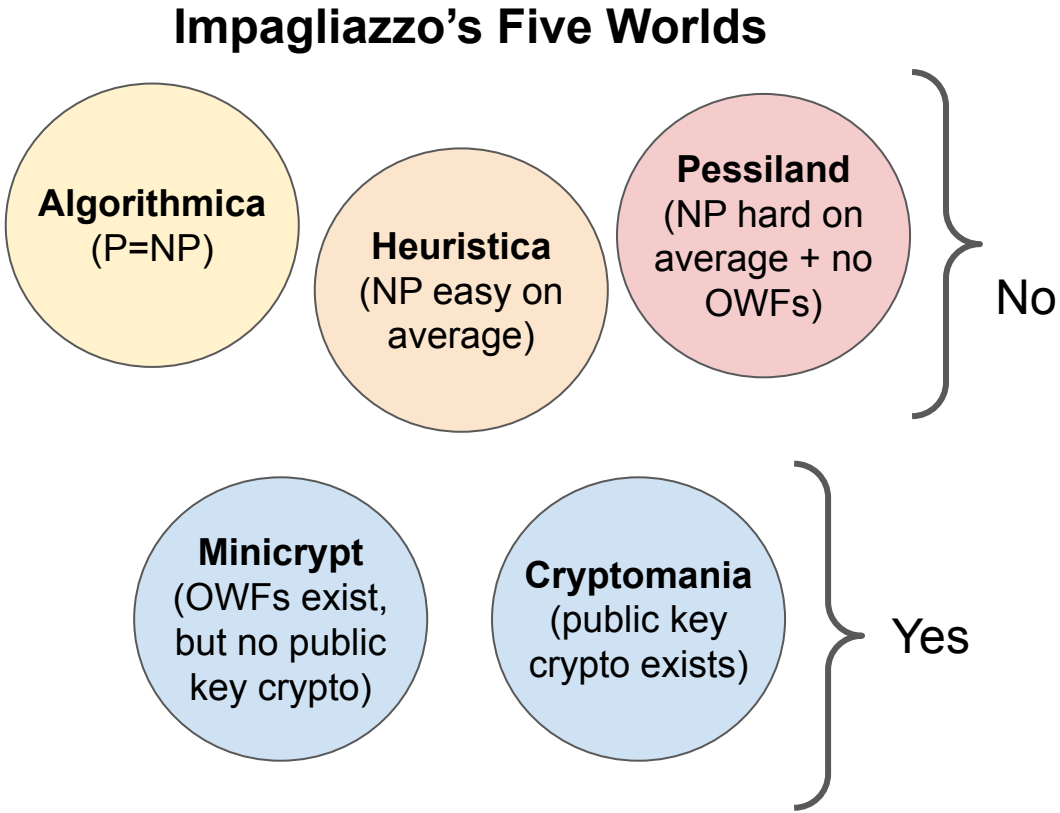
"Computational Security"

- E.g., AES, SHA
- Short keys

"Information-Theoretic Security"

- E.g,. One-Time Pad, secret sharing
- Long keys
- a.k.a "unconditional" / "non-cryptographic"

Does Computational Security Exist?



Which world do we live in???

OWF = One-Way Function

Cryptographic Obsolescence

DES, SHA-1, MD5

*Using Shor's algorithm

- Discrete Logarithm Problem
- Factoring

"Hope that whoever you're trying to keep the secret from is not a better mathematician than you are" - Michael Sipser

Computationally Secure Archives

"Cascade cipher"

$AES\text{-}256_{k1}(Blowfish_{k2}(3DES_{k3}(m)))$

Used in ArchiveSafeLT (Sabry & Samavi. ACSAC'22)

"All-or-nothing Transform"

Parse m as $m_1 m_2 m_3$

$c_i := m_i \oplus AES\text{-}256_k(i+1)$ for i=1,2,3

$c_4 := k \oplus SHA\text{-}256(c_1,c_2,c_3)$

$Reed\text{-}Solomon(c_1,c_2,c_3,c_4)$

Used in AONT-RS (Resch & Plank. FAST'11)

Regardless – computational security is susceptible to Harvest Now, Decrypt Later

Additive Secret Sharing

What if we generalized the One-Time Pad?

$$m \oplus \underbrace{r_1 \oplus r_2 \oplus \cdots \oplus r_{n-1}}_{\text{Give each of these “shares” to a different party}} =: c$$

Seems even more useless than One-Time Pad:

- Need *n* parties to store each “share” (and each share has same size as m)
- Cannot tolerate loss of a single share // what if we fixed this issue?

Shamir's Secret Sharing

Goal: share a secret m among n parties such that any $t < n$ parties can reconstruct m, but fewer than t parties learn no information about m.

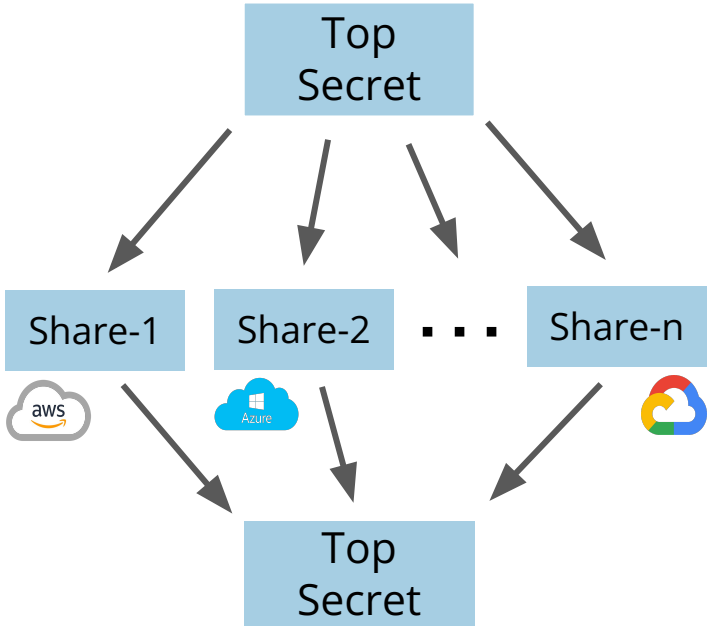
Given m (as a finite field element), and integers $t < n$, do the following:

1. Pick t -1 random field elements $r_1,...,r_{t-1}$
2. Define polynomial $p(x) = m + r_1 x + \cdots + r_{t-1} x^{t-1}$
3. Pick n arbitrary non-zero field elements $x_1,...,x_n$
4. Return shares $(x_1,p(x_1)),...,(x_n,p(x_n))$

Given any t shares, can uniquely interpolate p and retrieve $m = p(0)$

Just a non-systematic Reed-Solomon code in disguise…

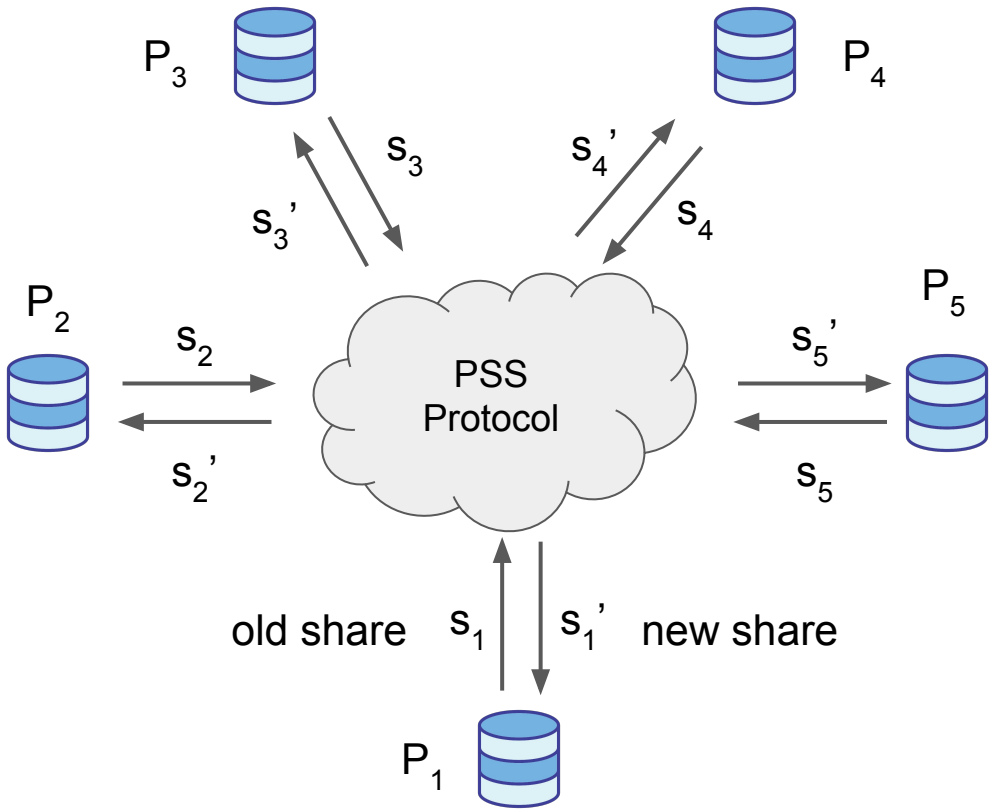
Secret Shared Archives



- Shamir's secret sharing provides information-theoretic "perfect secrecy" (like OTP)
- Tolerates lost/stolen shares
- High storage overhead
- Given time, adversary may eventually steal a threshold number of shares

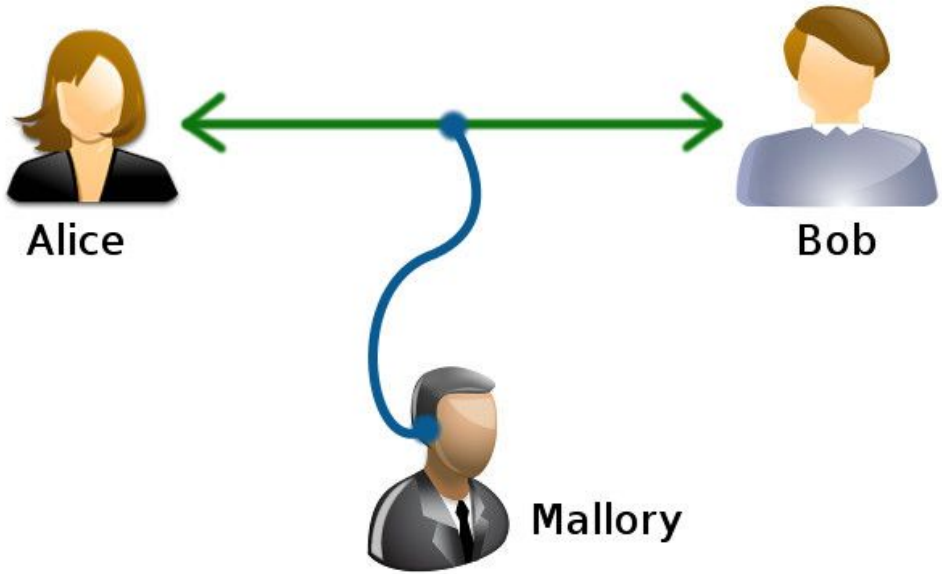
Used in many secure archival works: POTSHARDS, PASIS, LINCOS, etc.

Proactive Secret Sharing (PSS)



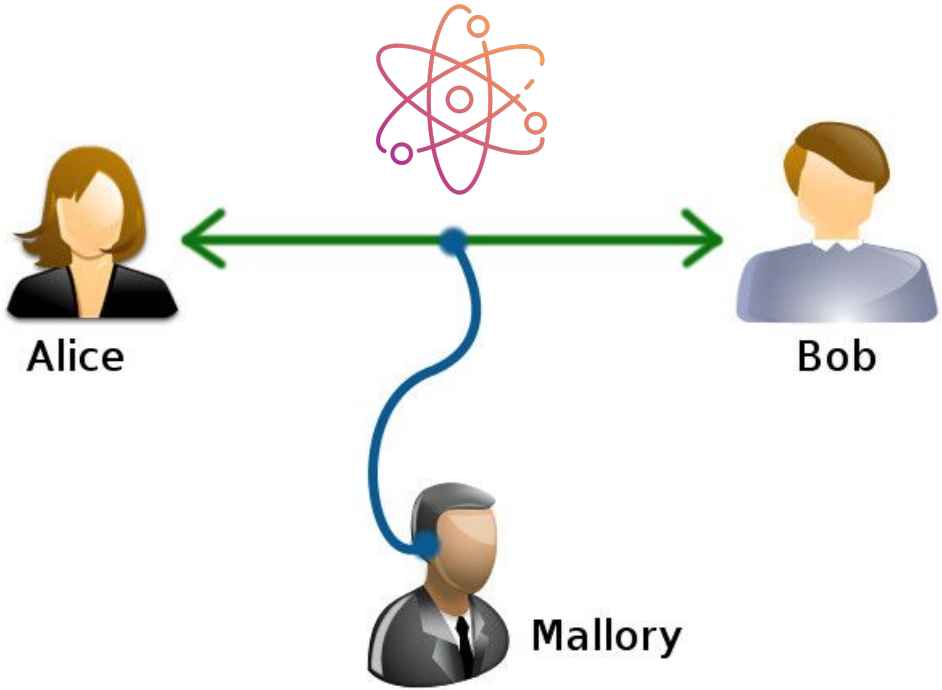
- Each party gets a new share independent of its old share
- Old/stolen shares obsolete *only if* honest parties delete their shares
- Nobody learns anything other than their new share
- Byzantine fault tolerant
- High communication overheads

Data in Transit



- May be easier for adversary to eavesdrop data in transit
- TLS encryption is only computationally secure!
- Can we protect data in transit information-theoretically?

Data in Transit



Yes: Quantum Key Distribution (QKD)

- Can generate a OTP key in a way impervious to eavesdropping and tampering
- Specialized infrastructure
- Not practical yet

Used in LINCOS (Braun et al. ASIA CCS'17)

Integrity

- Normally we use digital signatures for integrity against malicious tampering
- But digital signatures are also susceptible to cryptographic obsolescence
- Solution: use a chain of digitally signed timestamps

$$\text{CRYSTALS-Dilithium}_{sk3}(t_3, \text{ECDSA}_{sk2}(t_2, \text{RSA}_{sk1}(t_1, \text{SHA}(m))))$$

New post-quantum signature

3rd timestamp

Doesn't matter if broken after ECDSA applied

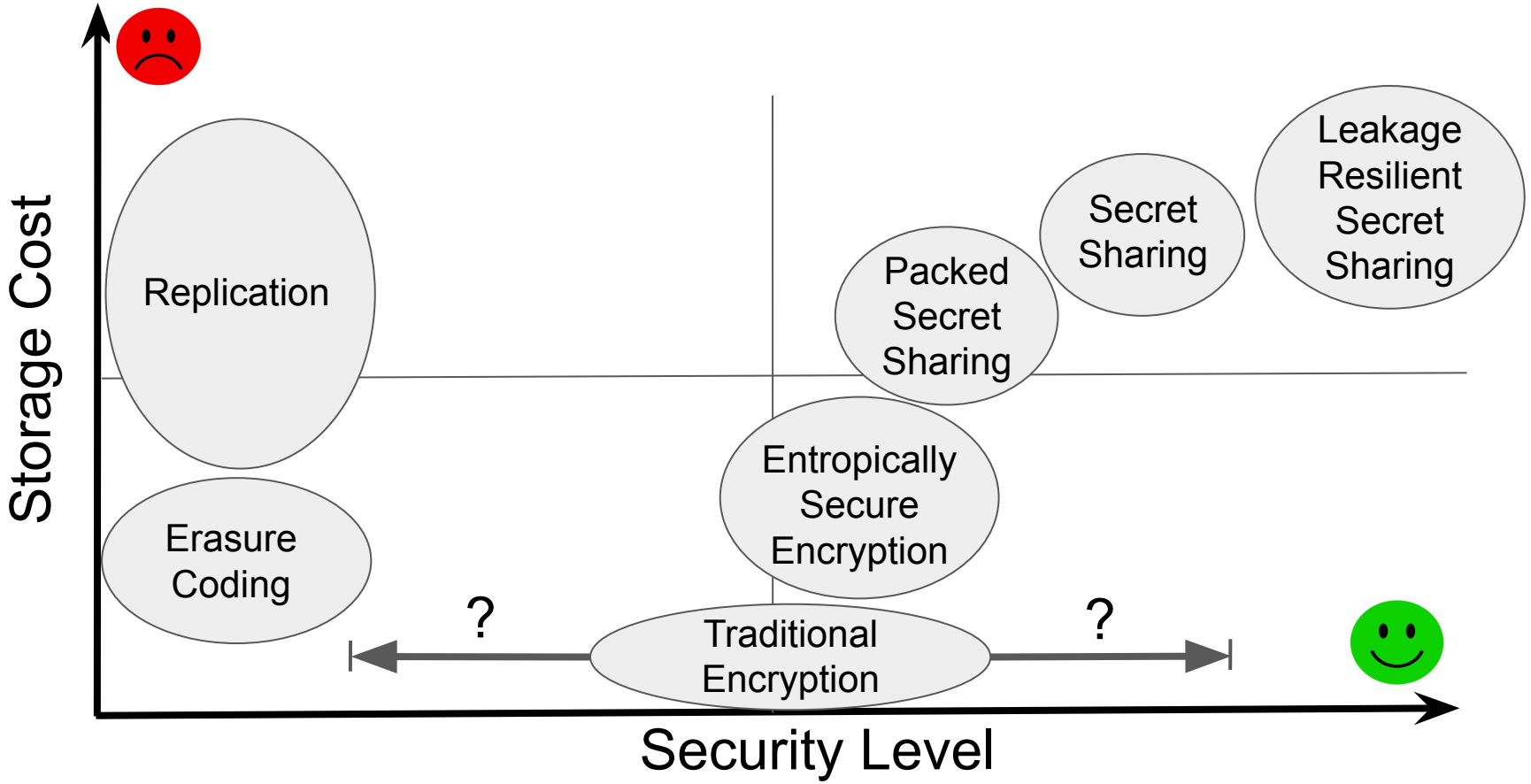
Use Pedersen commitment for perfect secrecy

We're all caught up

Systems	Confidentiality		Storage Cost
	In Transit	At Rest	
ArchiveSafeLT	Computational	Computational	Low
AONT-RS	Computational	Computational	Low
HasDPSS	Computational	ITS	High
LINCOS	ITS	ITS	High
PASIS	Computational	ITS (sometimes)	Low-High
POTSHARDS	Computational	ITS	High
VSR Archive	Computational	ITS	High
AWS, Azure, Google Cloud	Computational	Computational	Low

What can be improved?

A Fundamental Trade-off



Make Storage Cheaper

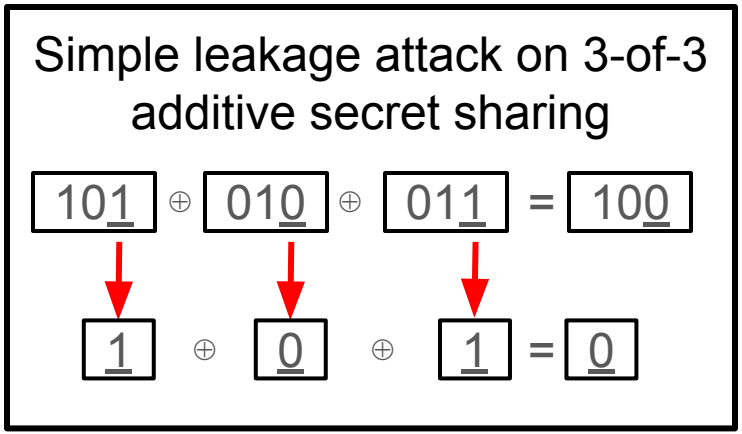
Glass

- Dense (~429 TB/in^3)
- Survives millenia
- No maintenance

DNA

- Super dense (~1 EB/mm^3)
- Survives centuries
- DNA synthesis slow + costly

Leakage-Resilient Secret Sharing



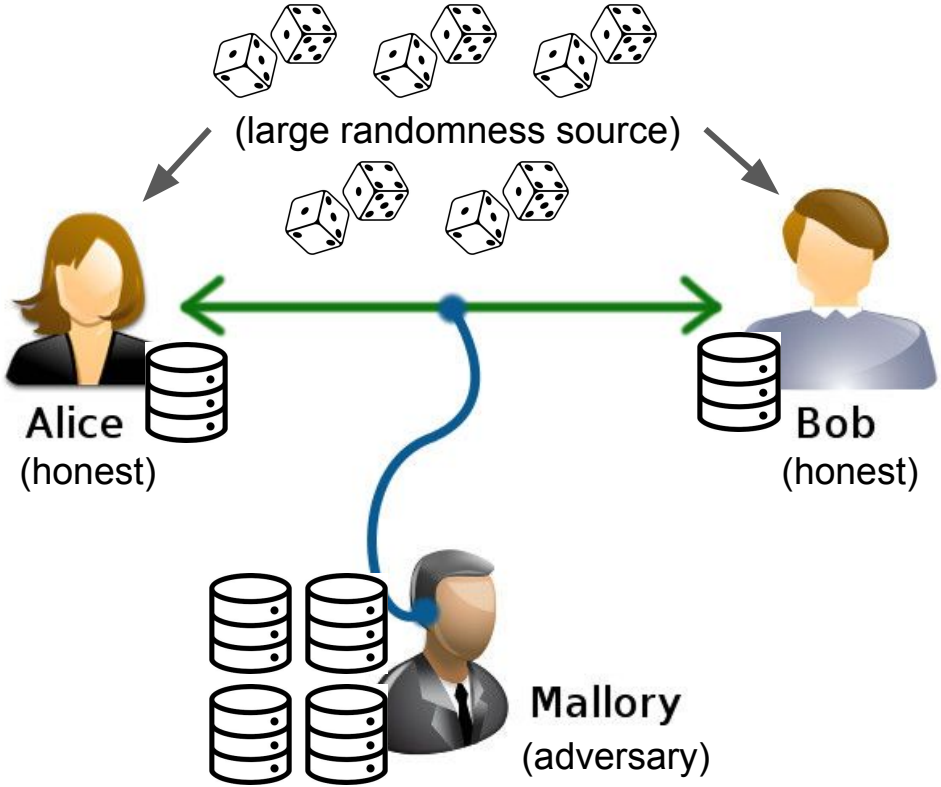
Leakage attack: adversary may leak a few bits of information about each secret share undetected via hidden side-channel.

Leakage-resilient secret sharing (LRSS) to the rescue?

Questions:

- Proactive LRSS?
- What is the right leakage model?

Bounded Storage Model (BSM)



BSM in a nutshell:

- Assume restrictions on adversarial storage capacity, and use a lot of public randomness.
- Yields information-theoretic channels.
- overdue for practical evaluation

Q&A

Secure Archival is Hard… *Really* Hard

Christopher Smith, Maliha Tabassum, Soumya Chowdary Daruru, Gaurav Kulhare, Arvin Wang, Ethan L. Miller*†, Erez Zadok

Stony Brook University, PureStorage, UC Santa Cruz†*