Centrum Wiskunde & Informatica

Robust Secret Sharing

Serge Fehr

CWI Amsterdam

www.cwi.nl/~fehr

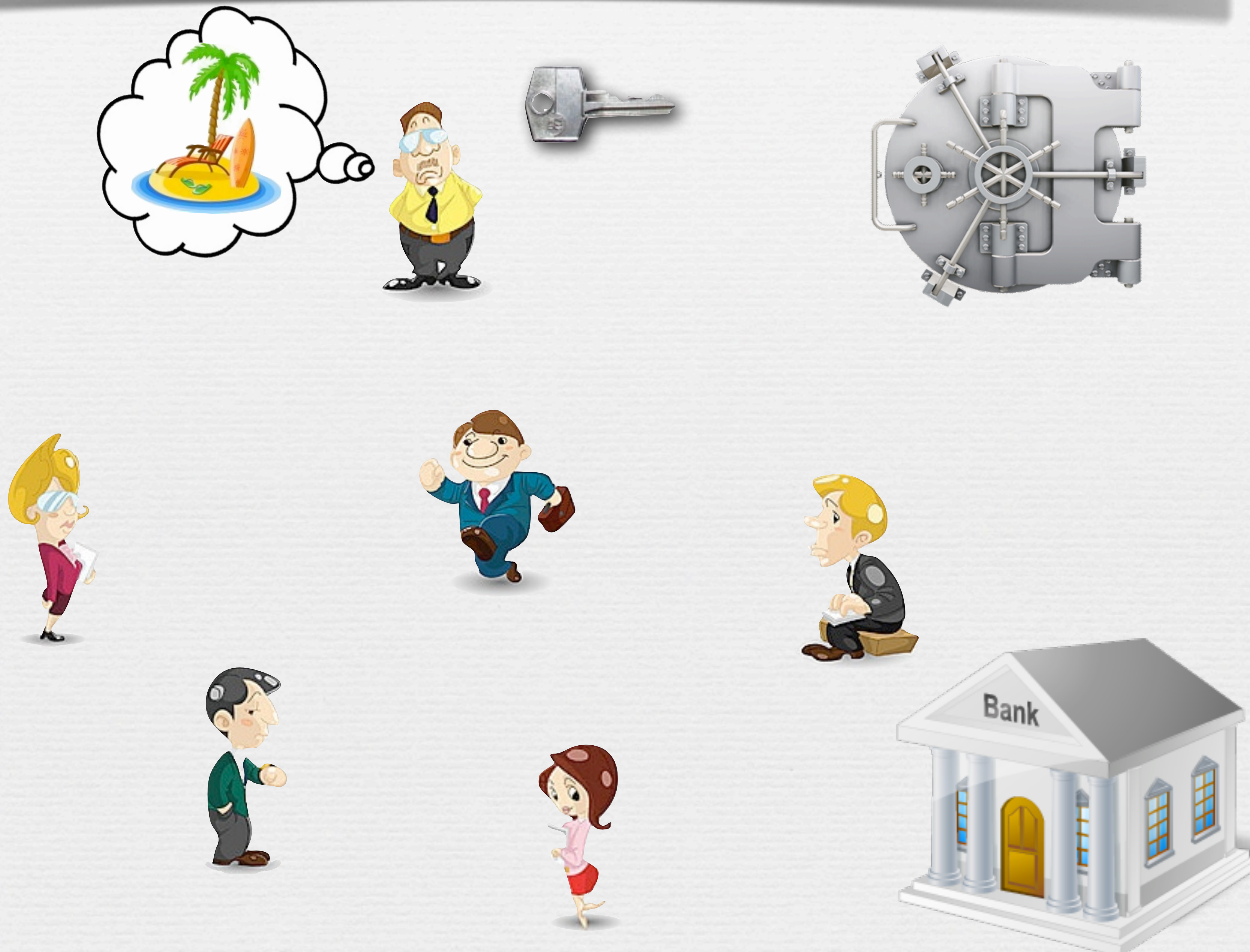
Based on joint work with:

Alfonso Cevallos (Leiden University / EPFL)
Rafail Ostrovsky (UCLA)
Yuval Rabani (Hebrew University of Jerusalem)

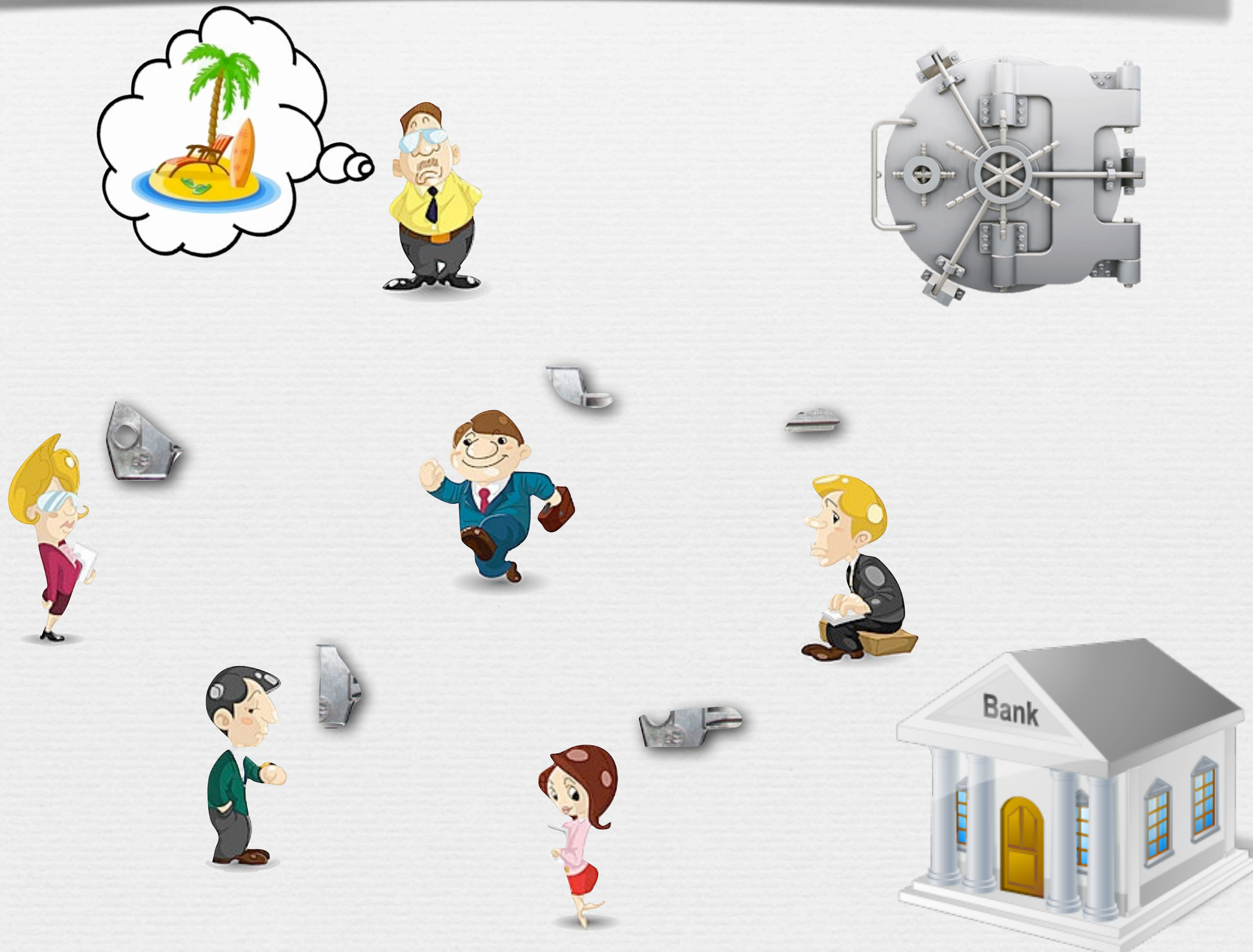
"Decentralizing Cryptographic Power"

- Cryptography relies on cryptographic keys
- Owner of the key has all the power to
  - decrypt ciphertexts, or
  - digitally sign messages,
  - etc.
- Vulnerable to:
  - dishonest owner who misuses the key
  - hackers breaking into the computer of the owner
  - unavailability of the owner
  - loss of the key
- **Goal**: decentralize cryptographic power

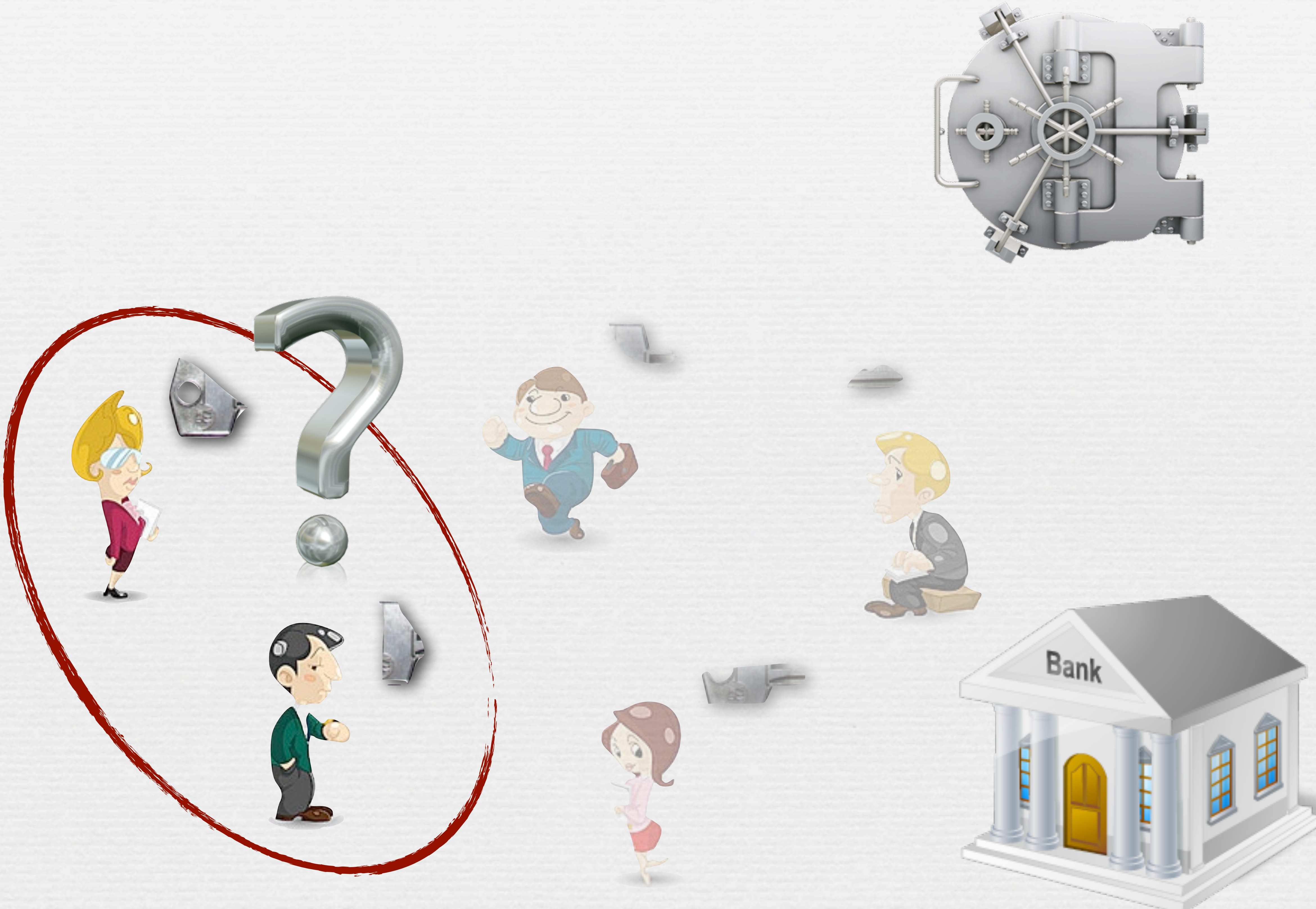
A (Non-Cryptographic) Toy Example



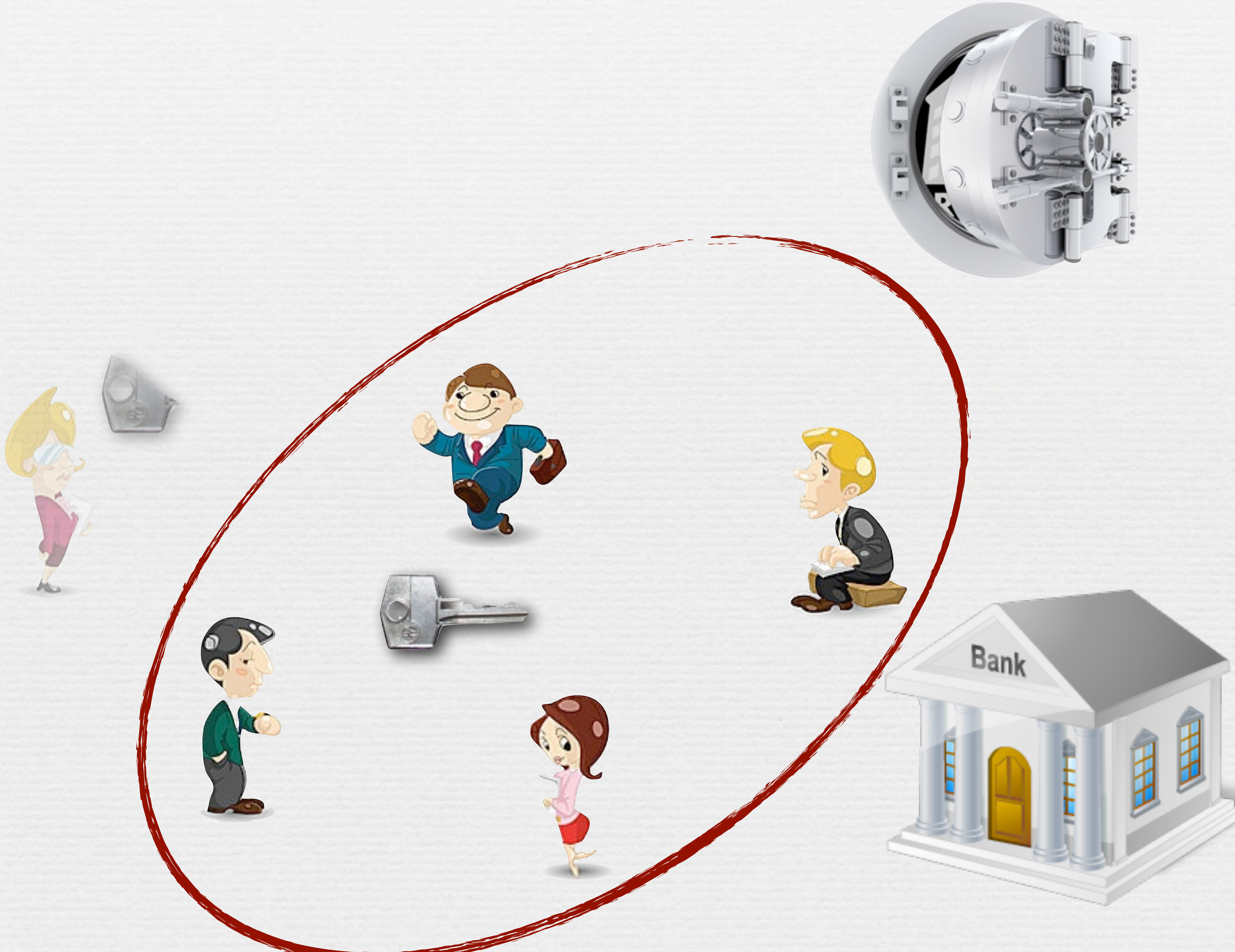
A (Non-Cryptographic) Toy Example



A (Non-Cryptographic) Toy Example



A (Non-Cryptographic) Toy Example



(t-out-of-n) Secret Sharing

secret: s

shares: $s_1 \quad s_2 \quad \cdots$

Not just:
- hard to compute
- some info missing

But: **statistically independent**

- **Privacy**: any t shares give no information on s

$$s_{i_1} \quad s_{i_2} \quad \cdots \quad s_{i_t} \quad \rightarrow \quad ?$$

- **Reconstructability**: any $t+1$ shares uniquely determine s

$$s_{i_1} \quad s_{i_2} \quad \cdots \quad s_{i_{t+1}} \quad \rightarrow \quad s$$

Shamir's Secret Sharing Scheme [Sha79]

secret: $s \in \mathbb{F}$ (finite field)

$$f(X) = s + a_1 X + \ldots + a_t X^t \in \mathbb{F}[X]$$

shares: $s_1 = f(1)$ $\quad\ldots\quad$ $s_n = f(n)$

- **Privacy** and **reconstructability** follow from Lagrange interpolation

Additional concern:

Dishonest "share holders" that hand in incorrect shares.

Robust Secret Sharing

secret: s

shares: $s_1 \quad s_2 \quad \cdots \quad s_n$

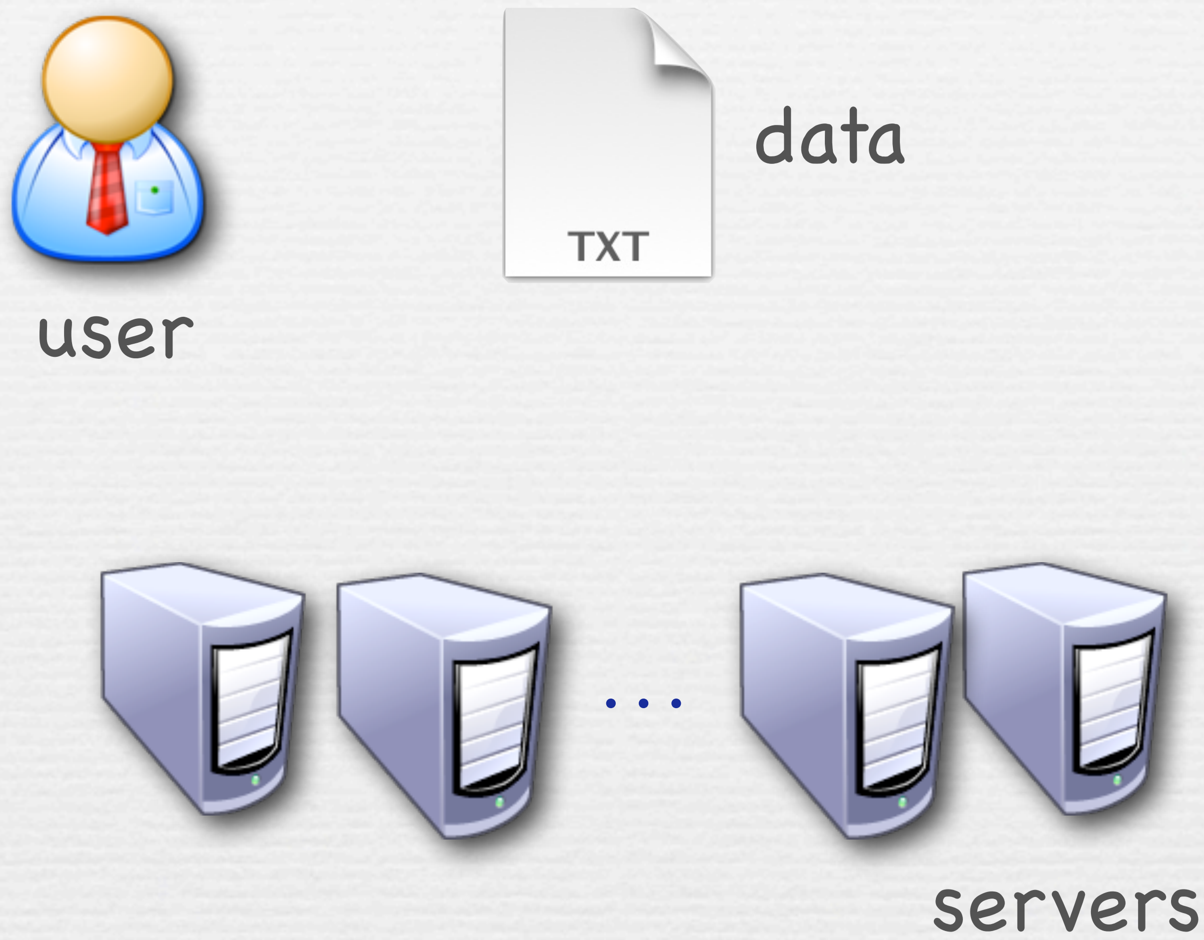
- **Privacy**: any t shares give no information on s

$$s_{i_1} \quad \cdots \quad s_{i_t} \quad \longrightarrow \quad ?$$

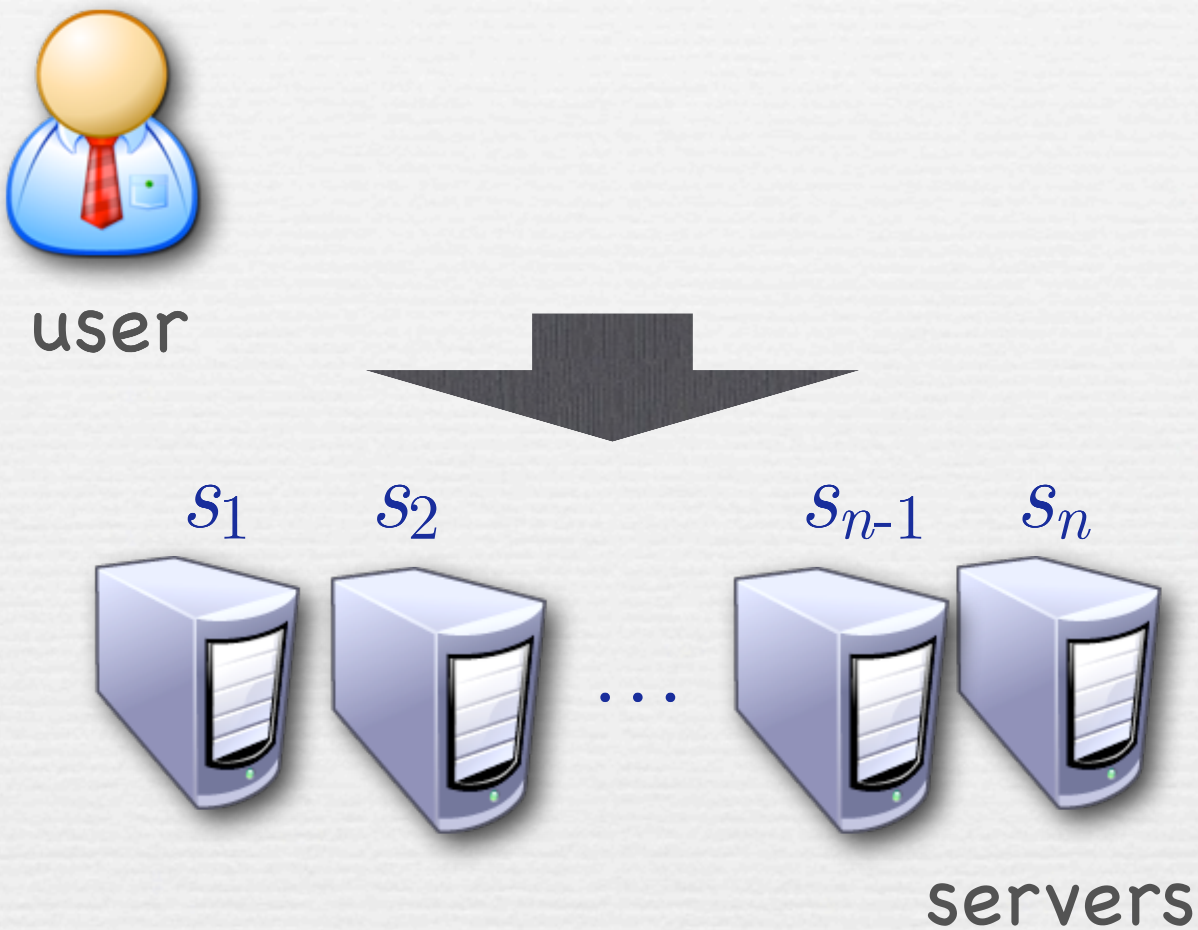
- **Robust reconstructability**:
  the set of all n shares determines s, even if t of them are faulty

$$\hat{s}_1 \quad \cdots \quad \hat{s}_t \quad s_{t+1} \quad \cdots \quad s_n \quad \longrightarrow \quad s$$

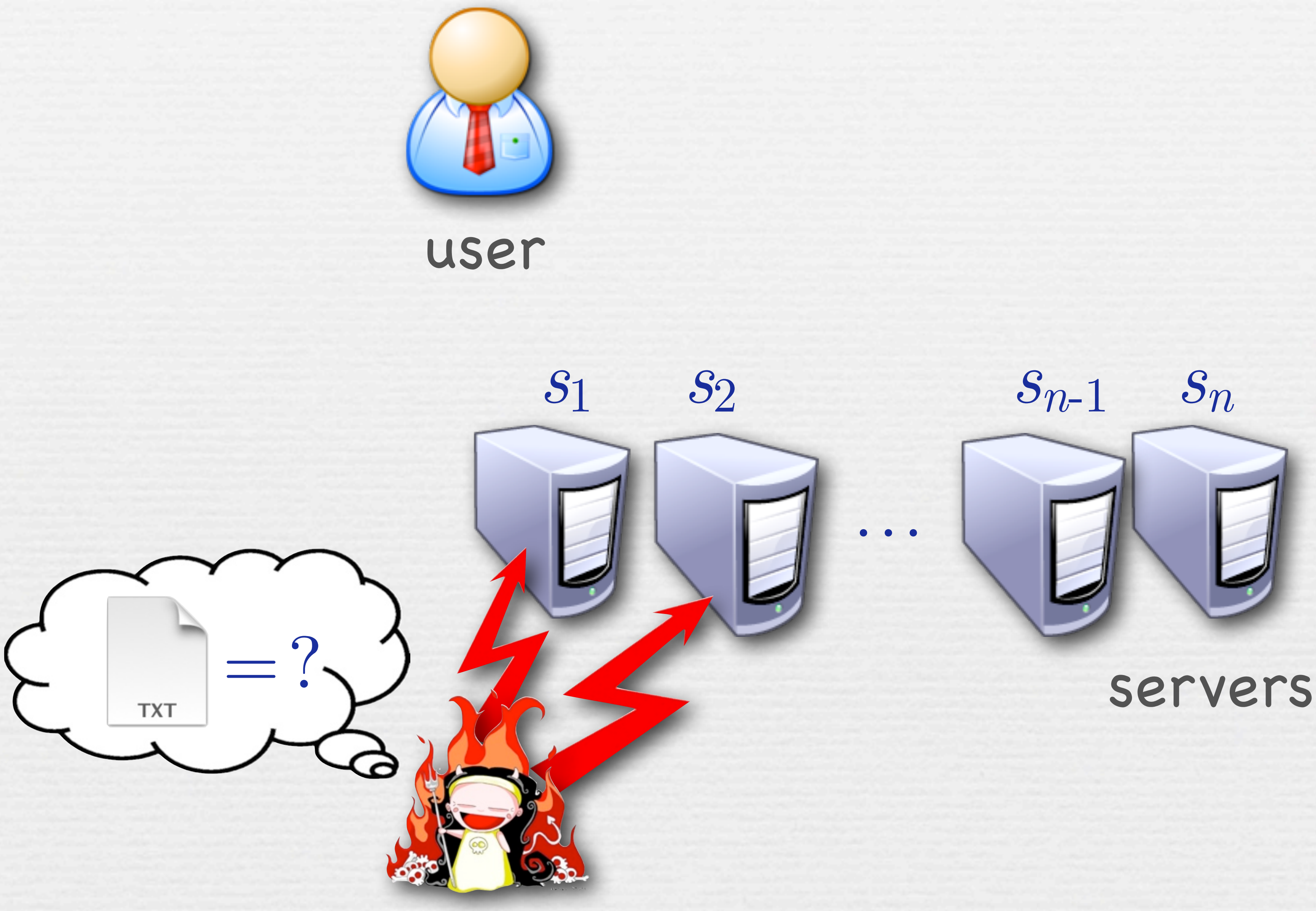
Application: Secure Data Storage



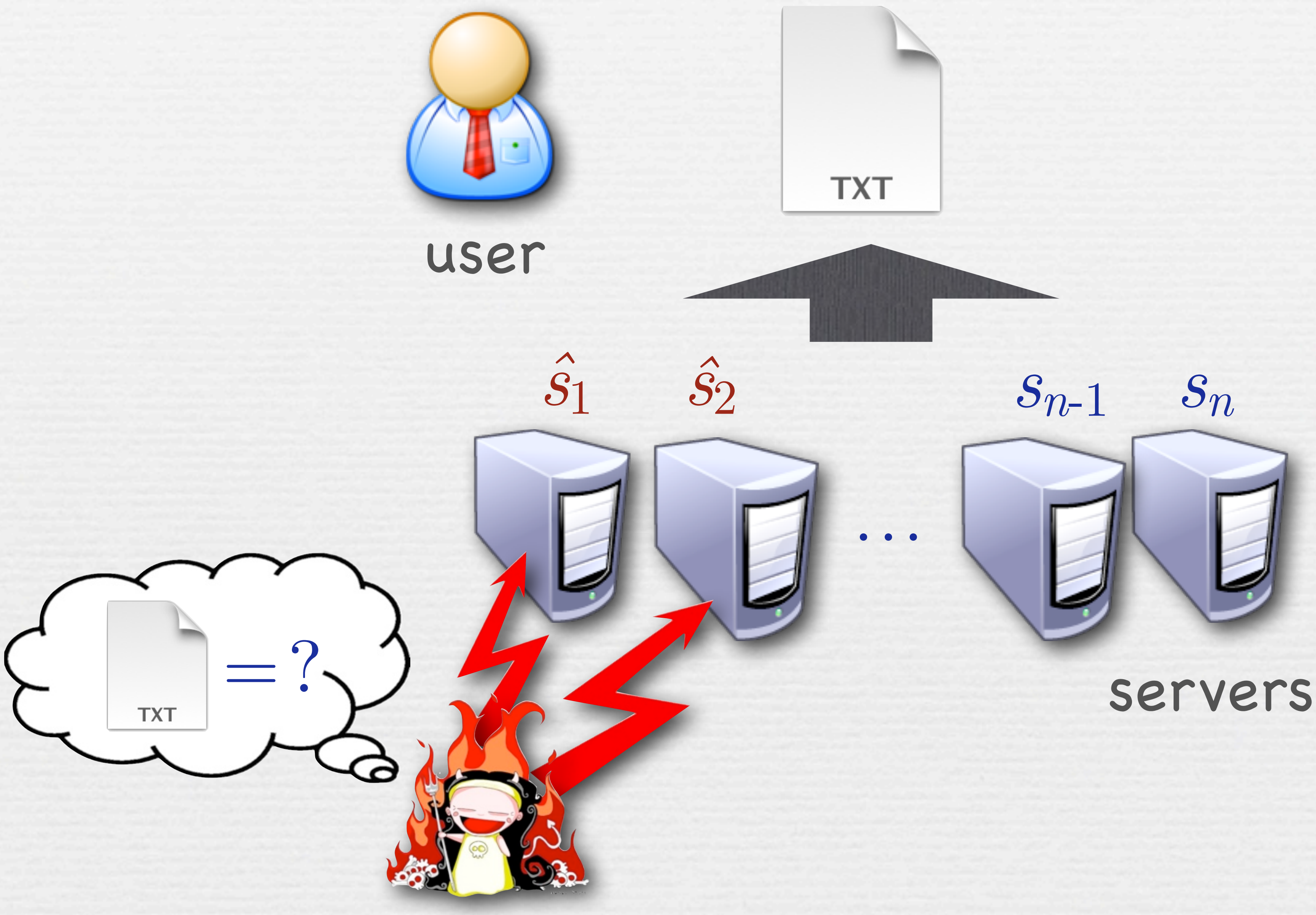
Application: Secure Data Storage



Application: Secure Data Storage



Application: Secure Data Storage



(Im)possibility

This talk: $n = 2t+1$, with information-theoretic security

plain Shamir sharing
plus RS decoding,
no error probability

additional checking data needed,
positive error probability: 2^{-k}

Known Schemes

- Rabin & Ben-Or (1989):
  - Overhead in share size: $\tilde{O}(k \cdot n)$ ☹
  - Computational complexity: $poly(k,n)$ ☺
- Cramer, Damgård & F (2001), based on Cabello, Padró & Sáez (1999):
  - Overhead in share size: $\tilde{O}(k+n)$ ☺ (lower bound: $\Omega(k)$)
  - Computational complexity: $exp(n)$ ☹
- Cevallos, F, Ostrovsky & Rabani (2012):
  - Overhead in share size: $\tilde{O}(k+n)$ ☺
  - Computational complexity: $poly(k,n)$ ☺

Further Outline

- Introduction
- The (simple) case $t < n/3$
- The Rabin & Ben-Or scheme
- The CDF 2001 scheme
- The CFOR 2012 scheme, and discussion of proof
- Conclusion

The (Simple) Case $n = 3t+1$

$s \in \mathbb{F}$

$f(X) = s + a_1 X + \ldots + a_t X^t \in \mathbb{F}[X]$

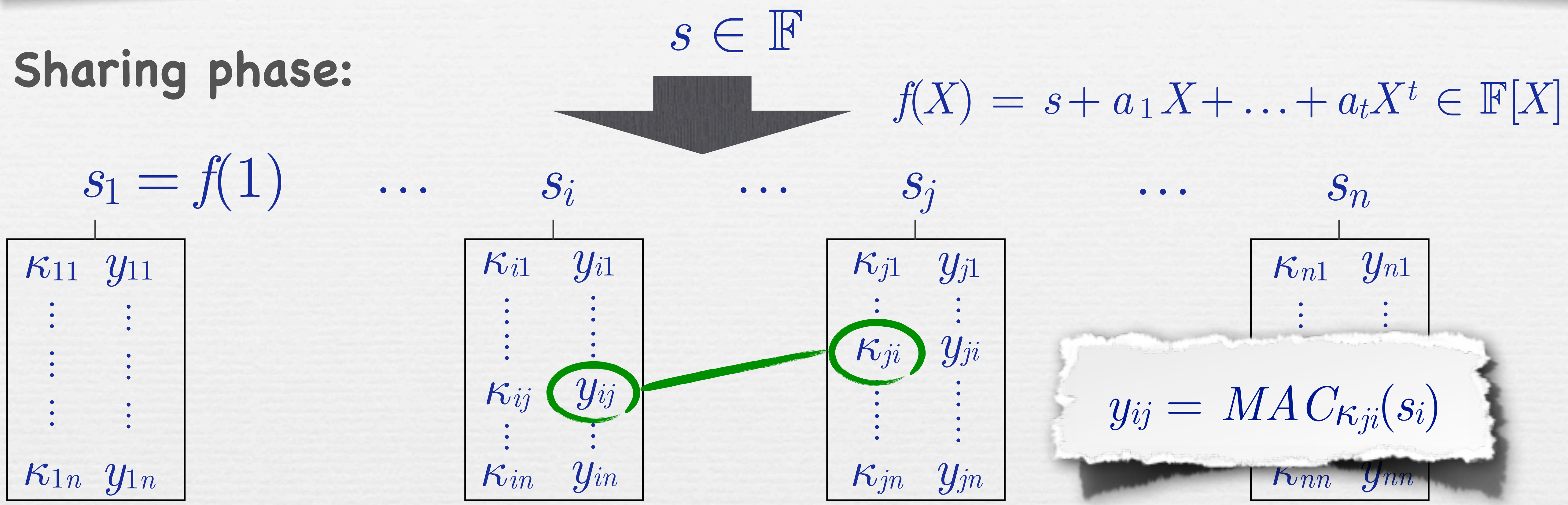
$s_1 = f(1)$ $\cdots$ s_{t+1}	s_{t+2} $\cdots$ s_{2t+1}	$\hat{s}_{n-t+1}$ $\cdots$ $\hat{s}_n$
$t+1$ correct shares -> determines f	$r=t$ redundant correct shares	$e=t$ faulty shares

Reed-Solomon decoding: If $e \leq r$ (satisfied here) then

- f is uniquely determined from $s_1, \ldots, \hat{s}_n$
- f can be efficiently computed (Berlekamp-Welch)

The Rabin & Ben-Or Scheme ($n = 2t+1$)

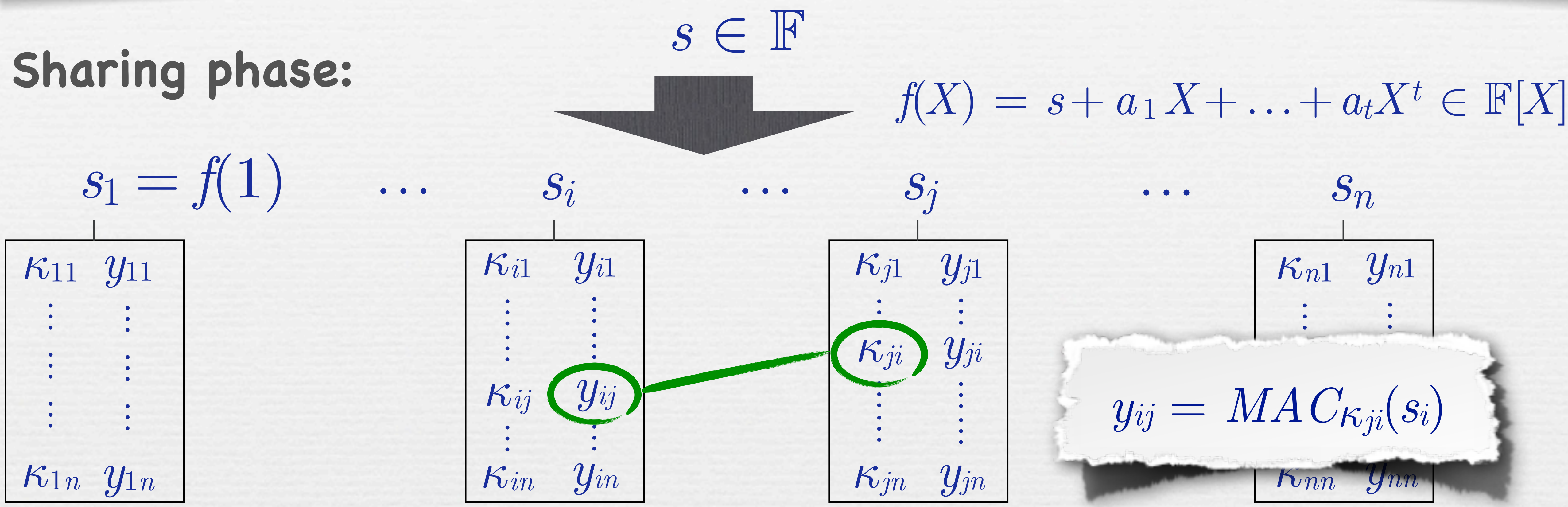
Sharing phase:



- MAC security: for any $\hat{s}_i \neq s_i$ and $\hat{y}_{ij}$: $P[\hat{y}_{ij} = MAC_{\kappa_{ji}}(\hat{s}_i)] \leq \varepsilon$.
- Example: $\kappa_{ij} = (\alpha_{ij}, \beta_{ij}) \in \mathbb{F}^2$ and $y_{ij} = MAC_{\kappa_{ji}}(s_i) = \alpha_{ij} \cdot s_i + \beta_{ij}$.
- For error probability $\varepsilon \leq 2^{-k}$:
  - bit size $|\kappa_{ij}|, |y_{ij}| \geq k$
  - overhead per share (above Shamir share): $\Omega(k \cdot n)$

The Rabin & Ben-Or Scheme ($n = 2t+1$)

Sharing phase:



Reconstruction phase:

1. For every share s_i:

   accept s_i iff it is consistent with keys of $\geq t+1$ players,

   (meaning $\#\{j \mid y_{ij} = MAC_{\kappa_{ji}}(s_i)\} \geq t+1$)

2. Reconstruct s using the accepted shares s_i.

The CDF 2001 Scheme

Sharing phase:

$s \in \mathbb{F}$, $r \in \mathbb{F}$ and $p = s \cdot r \in \mathbb{F}$

$s_1 = f(1)$	$\dots$	s_i	$\dots$	s_n
$r_1 = g(1)$	$\dots$	r_i	$\dots$	r_n
$p_1 = h(1)$	$\dots$	p_i	$\dots$	p_n

Reconstruction phase:

For every $A \subset \{1,\dots,n\}$ with $|A| = t+1$:

- reconstruct s', r' and p' from $(s_i)_{i \in A}$, $(r_i)_{i \in A}$ and $(p_i)_{i \in A}$
- if $s' \cdot r' = p'$ then output s' and halt

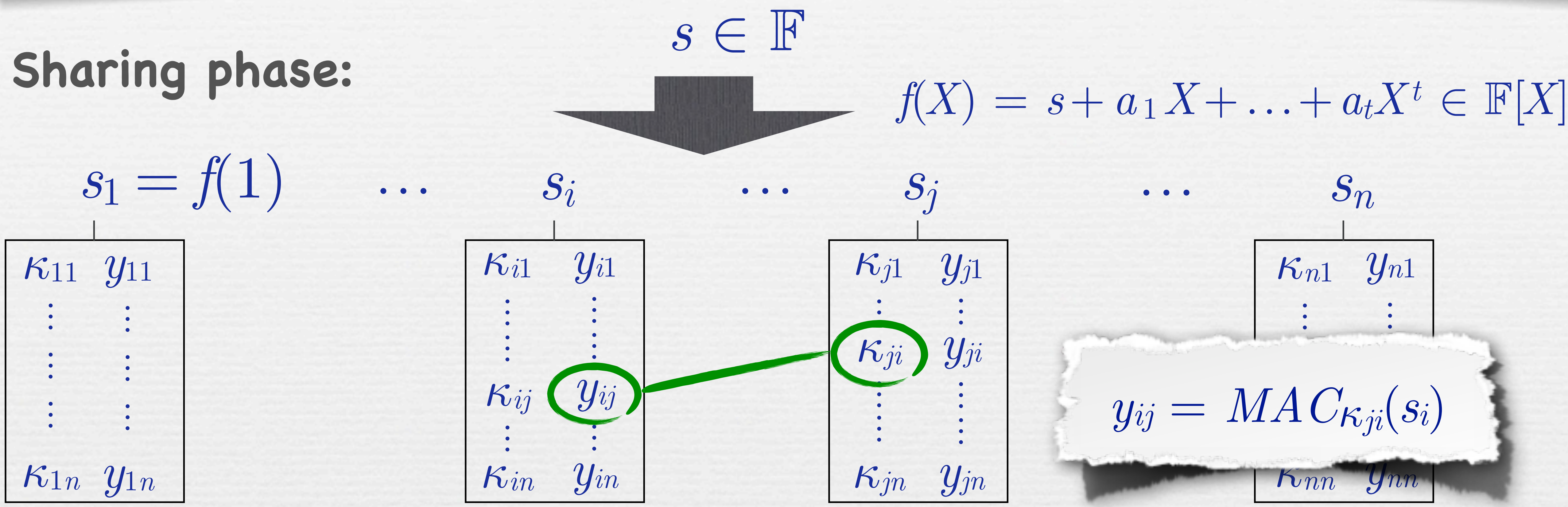
Note: Running time is exponential in n

Further Outline

- Introduction
- The (simple) case $t < n/3$
- The Rabin & Ben-Or scheme
- The CDF 2001 scheme
- The CFOR 2012 scheme, and discussion of proof
- Conclusion

The CFOR 2012 Scheme

Sharing phase:



- Use **small** tags and keys $|\kappa_{ij}|, |y_{ij}| = \tilde{O}(k/n+1)$ (instead of $O(k)$)
- Gives: overhead per share: $n \cdot \tilde{O}(k/n+1) = \tilde{O}(k+n)$
- Problem:
  - MAC has weak security
  - incorrect shares may be consistent with some honest players
  - Rabin & Ben-Or reconstruction fails

The CFOR 2012 Scheme

Sharing phase:

$$s \in \mathbb{F}$$

$$f(X) = s + a_1 X + \ldots + a_t X^t \in \mathbb{F}[X]$$

$$s_1 = f(1) \quad \ldots \quad s_i \quad \ldots \quad s_j \quad \ldots \quad s_n$$

$$\begin{array}{cc} \kappa_{11} & y_{11} \\ \vdots & \vdots \\ \kappa_{1n} & y_{1n} \end{array} \qquad \begin{array}{cc} \kappa_{i1} & y_{i1} \\ \vdots & \vdots \\ \kappa_{ij} & y_{ij} \\ \vdots & \vdots \\ \kappa_{in} & y_{in} \end{array} \qquad \begin{array}{cc} \kappa_{j1} & y_{j1} \\ \vdots & \vdots \\ \kappa_{ji} & y_{ji} \\ \vdots & \vdots \\ \kappa_{jn} & y_{jn} \end{array} \qquad \begin{array}{cc} \kappa_{n1} & y_{n1} \\ \vdots & \vdots \\ \kappa_{nn} & y_{nn} \end{array}$$

$$y_{ij} = MAC_{\kappa_{ji}}(s_i)$$

- Use **small** tags and keys $|\kappa_{ij}|, |y_{ij}| = \tilde{O}(k/n+1)$ (instead of $O(k)$)
- Gives: overhead per share: $n \cdot \tilde{O}(k/n+1) = \tilde{O}(k+n)$
- Problem
  - MAC
  - incorrect shares may be consistent with some honest players
  - Rabin & Ben-Or reconstruction fails

Need: better reconstruction procedure

Improving the Reconstruct Procedure

- Example: Say that
  - s_1 is consistent with $\{1,\ldots,n\}$ -> accept s_1
  - s_2 is consistent with $\{1,\ldots,t+1\}$ -> accept s_2
  - s_3 is consistent with $\{2,\ldots,t+1\}$ -> reject s_3
  - ...
- Rabin & Ben-Or reconstruction: accepts s_1, s_2 etc.
- In our new reconstruction:
  - Notice: s_2 is consistent with $\leq t$ **honest** players (as 3 is dishonest)
    => s_2 stems from dishonest player
  - Will reject s_2

Improving the Reconstruct Procedure

- Example: Say that
  - s_1 is consistent with $\{1,\ldots,n\}$ -> accept s_1
  - s_2 is consistent with $\{1,\ldots,t+1\}$ -> accept s_2
  - s_3 …
  - …
- Rabi…

Our new reconstruction:

Whenever we reject a share, we reconsider the so-far accepted shares.

- In ou…

Plus: Reed-Solomon decoding.

  - Notice: s_2 is consistent with $\leq t$ honest players (as 3 is dishonest)
    => s_2 stems from dishonest player
  - Will reject s_2

The New Reconstruction Procedure

```
(Init)  Set  Good := {1,...,n}

(Loop)  For every i ∈ Good:

          if #{j ∈ Good | y_ij = MAC_κji(s_i)} ≤ t then
            - set  Good := Good ∖ {i}
            - redo (Loop)

(Dec)   Set  s := Reed-Solomon({s_i}_{i∈Good})
```

Main Theorem. If MAC is ε-secure then our scheme is δ-robust with

$$\delta \leq e\cdot((t+1)\cdot\varepsilon)^{(t+1)/2} \qquad \text{(where } e = \exp(1)\text{).}$$

Corollary. Using MAC with $|\kappa_{ij}|,|y_{ij}| = O(k/n + \log n)$ gives $\delta \leq 2^{-\Omega(k)}$ and overhead in share size $\tilde{O}(k+n)$.

What Makes the Proof Tricky

1. **Optimal strategy for dishonest players is unclear**

- In Rabin & Ben-Or: an incorrect share for every dishonest player
- Here: some dishonest players may hand in correct shares
- Such a passive dishonest player:
  - stays "alive"
  - can support bad shares
- **The more** such passive dishonest players:
  - **the easier it gets** for bad shares to survive
  - **the more** bad shares have to survive to fool RS decoding (# bad shares > # correct shares of dishonest players)
- Optimal trade-off: unclear

What Makes the Proof Tricky

2. Circular dependencies

- Whether $\hat{s}_i$ gets accepted **depends** on whether $\hat{s}_j$ gets accepted …
- … and vice versa
- Cannot analyze individual bad shares
- If we try, we run into a circularity

Summary

- First robust secret sharing scheme for $n = 2t{+}1$, with
  - small overhead $\tilde{O}(k{+}n)$ in share size
  - efficient sharing and reconstruction procedures
- Scheme is simple and natural adaptation of Rabin & Ben-Or
- Proof is non-standard and non-trivial

- Open problem:
  - Scheme with overhead $O(k)$ (= proven lower bound)
- Note:
  - CDF and CFOR have a $\Omega(n)$ gap (for different reasons)
  - Not known if this is inherent or not.