# Security of Linear Secret-Sharing Schemes Against Mass Surveillance

Ruxandra F. Olimid

Crypto vs. Mass Surveillance:
The Uneasy Relationship Workshop 2016

November 14, 2016
Trondheim, Norway

Norwegian University of Science and Technology

**Security** of

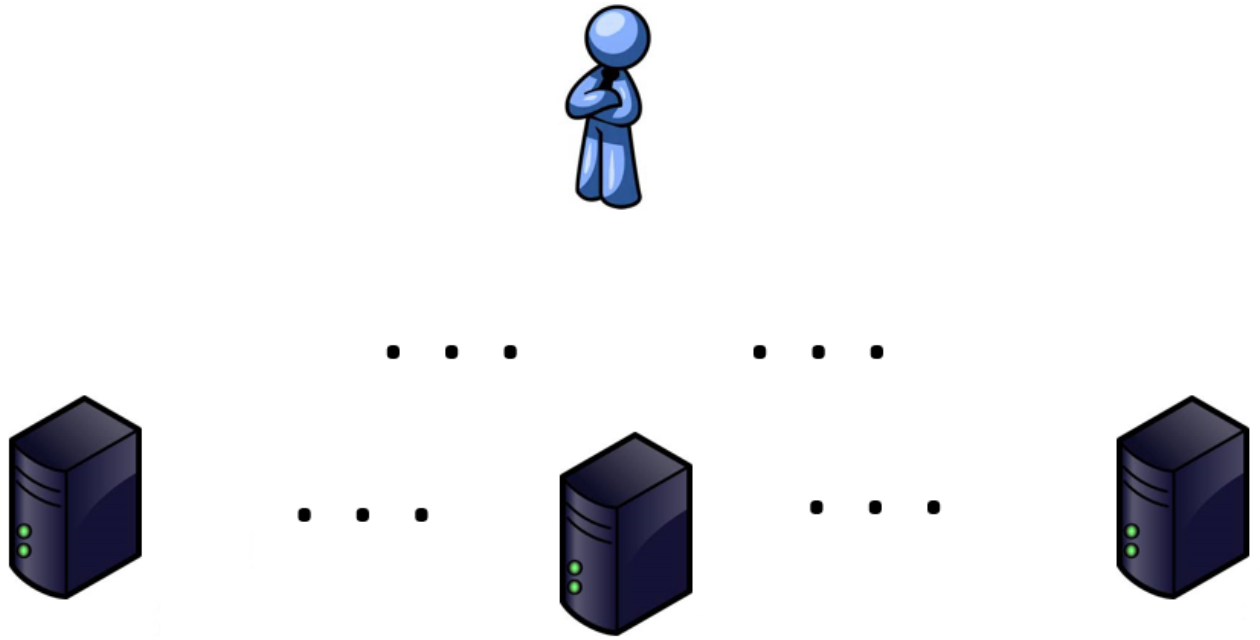
**Linear Secret-Sharing Schemes**

Against **Mass Surveillance**

# Secret Sharing Schemes (SSS)

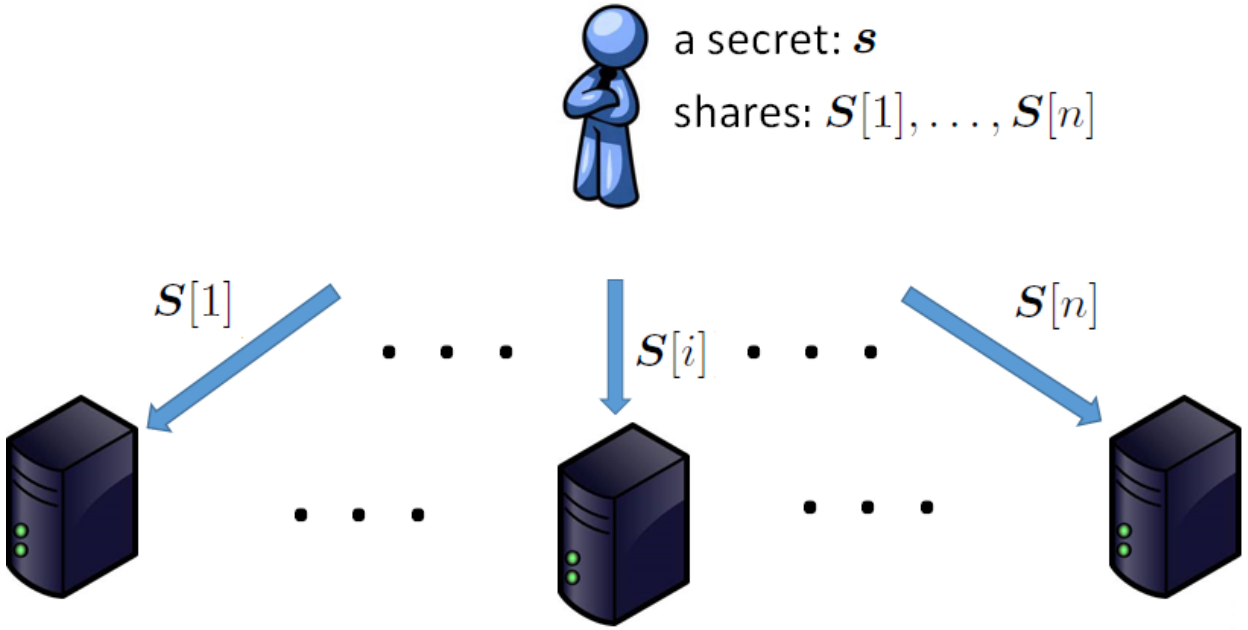
*Split a secret into shares such that the secret can be recovered only by using an authorised set of shares*

# Secret Sharing Schemes (SSS)



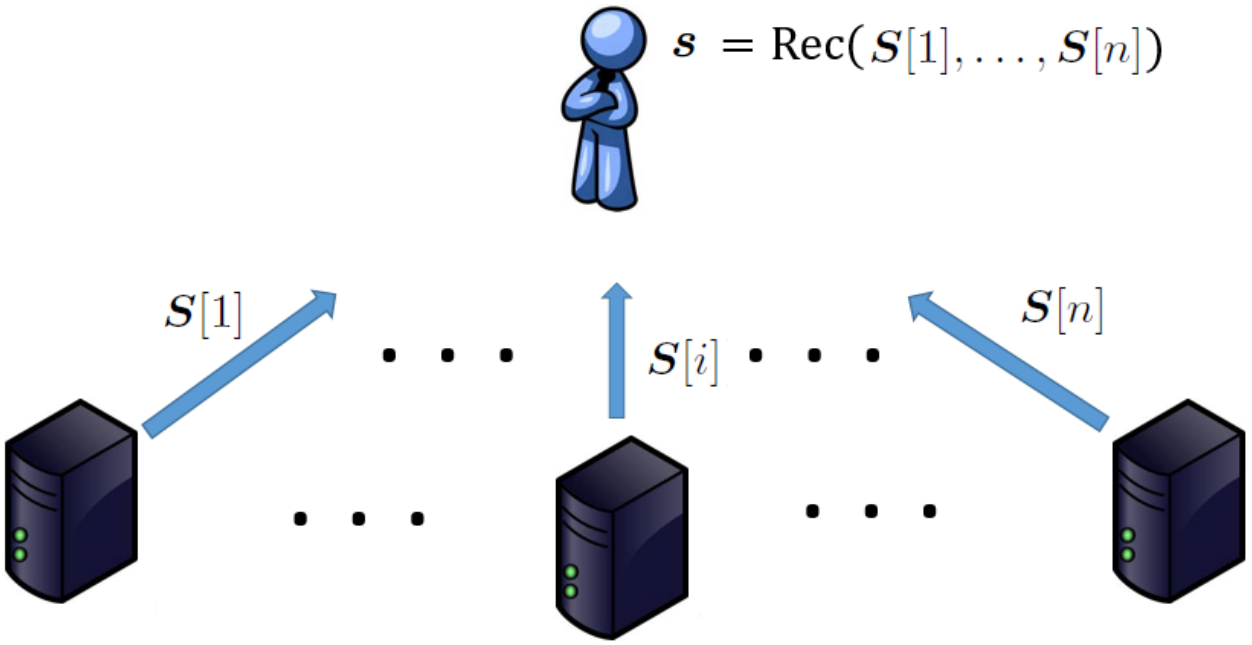
*Split a secret into shares such that the secret can be recovered only from authorised sets of shares*

# Secret Sharing Schemes (SSS)




*Split a secret into shares such that the secret can be recovered only from authorised sets of shares*

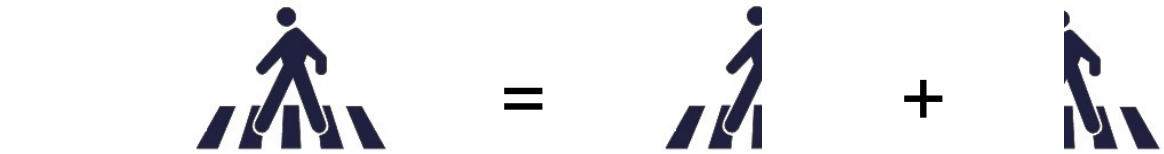
# Secret Sharing Schemes (SSS)

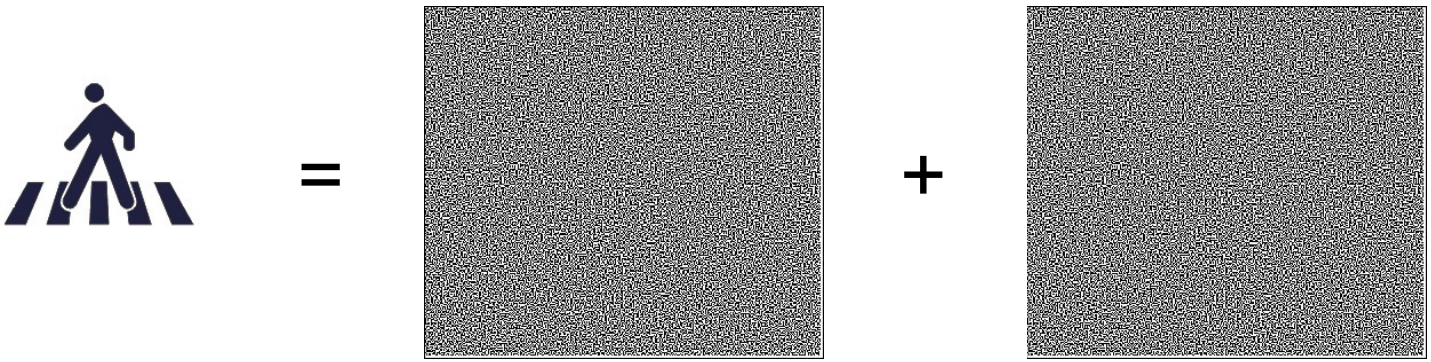



*Split a secret into shares such that the secret can be recovered only from authorised sets of shares*

# Visual SSS

*Split a secret into shares such that the secret can be recovered only from authorised sets of shares*





# All-or-Nothing SSS

*Split a secret into shares such that the secret can be recovered only from authorised sets of shares*

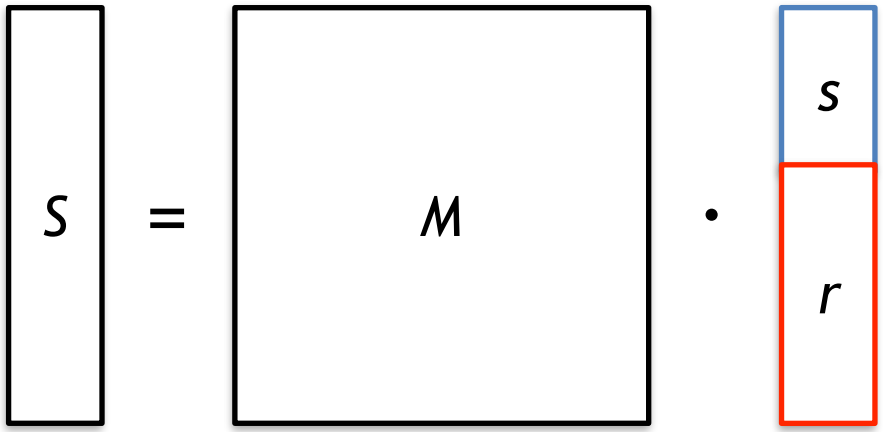
1000 1101 = 1011 0110 XOR 0011 1011

???? ???? = 1011 0110 XOR ???? ????

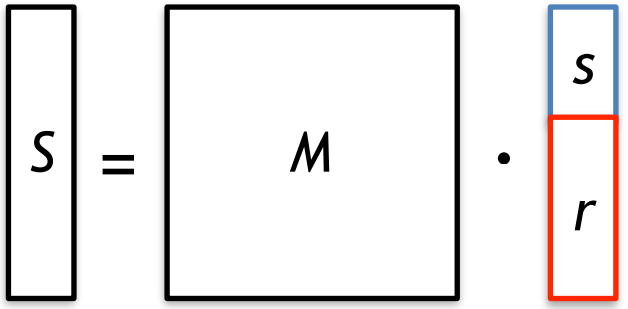
0??? ???? = 1011 0110 XOR 1??? ????

# Linear SSS

*Split a secret into shares such that the secret can be recovered only from authorised sets of shares*



# Linear SSS



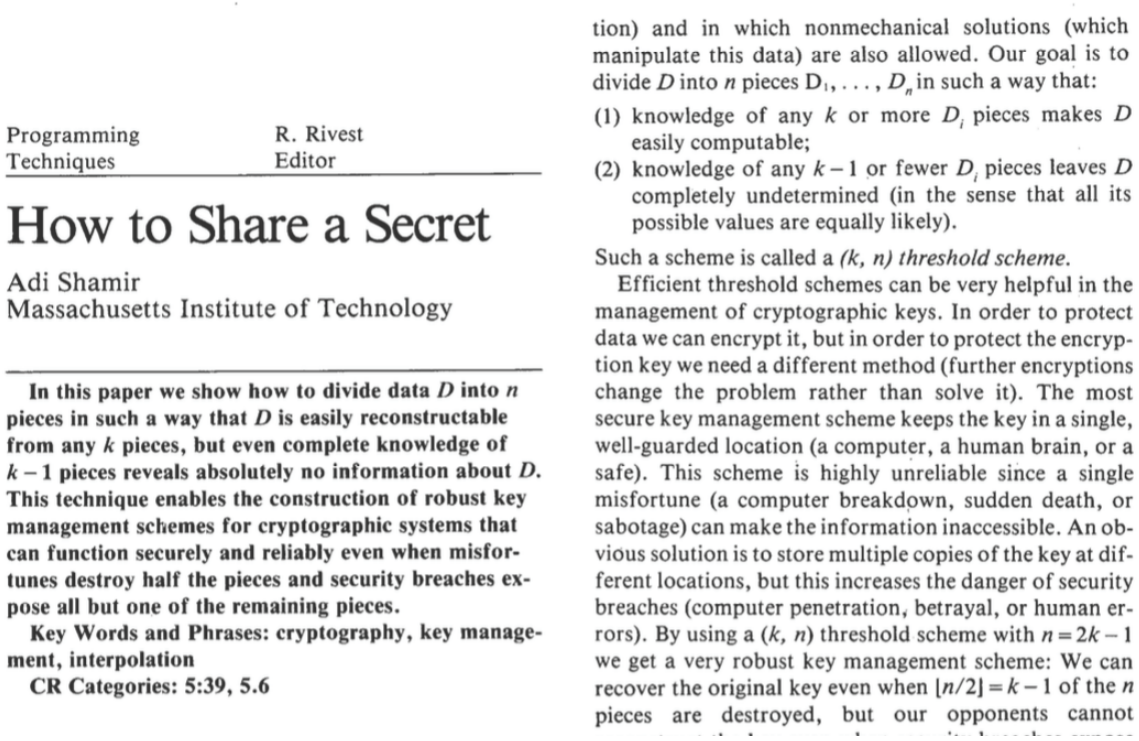

- $\Pi_{\mathbf{M}} = (\mathsf{Sh}_{\mathbf{M}}, \mathsf{Rec}_{\mathbf{M}})$ is linear, where:

$\mathsf{Sh}_{\mathbf{M}}(\mathbf{s})$
- $\mathbf{r} \twoheadleftarrow \mathbb{F}^d$
- $\mathbf{f}^T \leftarrow (\mathbf{s}, \mathbf{r})^T$
- $\mathbf{S} \leftarrow \mathbf{M} \cdot \mathbf{f}$
- **return** $\mathbf{S}$

$\mathsf{Rec}_{\mathbf{M}}(\mathbf{S}_B)$
- **if** $B$ *is qualified* **then**
  - $\mathbf{s} \leftarrow \mathbf{N}_B \cdot \mathbf{S}_B$
- **else**
  - $\mathbf{s} \leftarrow \perp$
- **return** $\mathbf{s}$

# Connection to Mass Surveillance?

Motivation: management of cryptographic keys



Programming Techniques — R. Rivest Editor

## How to Share a Secret

Adi Shamir
Massachusetts Institute of Technology

**In this paper we show how to divide data $D$ into $n$ pieces in such a way that $D$ is easily reconstructable from any $k$ pieces, but even complete knowledge of $k-1$ pieces reveals absolutely no information about $D$. This technique enables the construction of robust key management schemes for cryptographic systems that can function securely and reliably even when misfortunes destroy half the pieces and security breaches expose all but one of the remaining pieces.**

**Key Words and Phrases: cryptography, key management, interpolation**
**CR Categories: 5:39, 5.6**

tion) and in which nonmechanical solutions (which manipulate this data) are also allowed. Our goal is to divide $D$ into $n$ pieces $D_1, \ldots, D_n$ in such a way that:

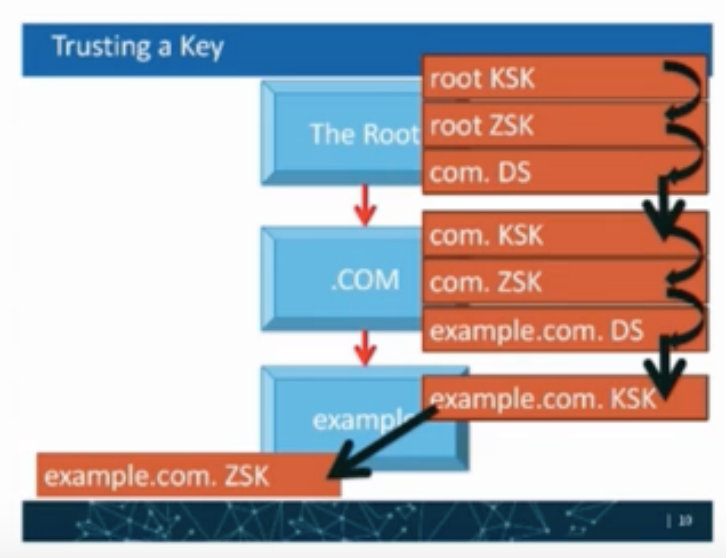
(1) knowledge of any $k$ or more $D_i$ pieces makes $D$ easily computable;
(2) knowledge of any $k-1$ or fewer $D_i$ pieces leaves $D$ completely undetermined (in the sense that all its possible values are equally likely).

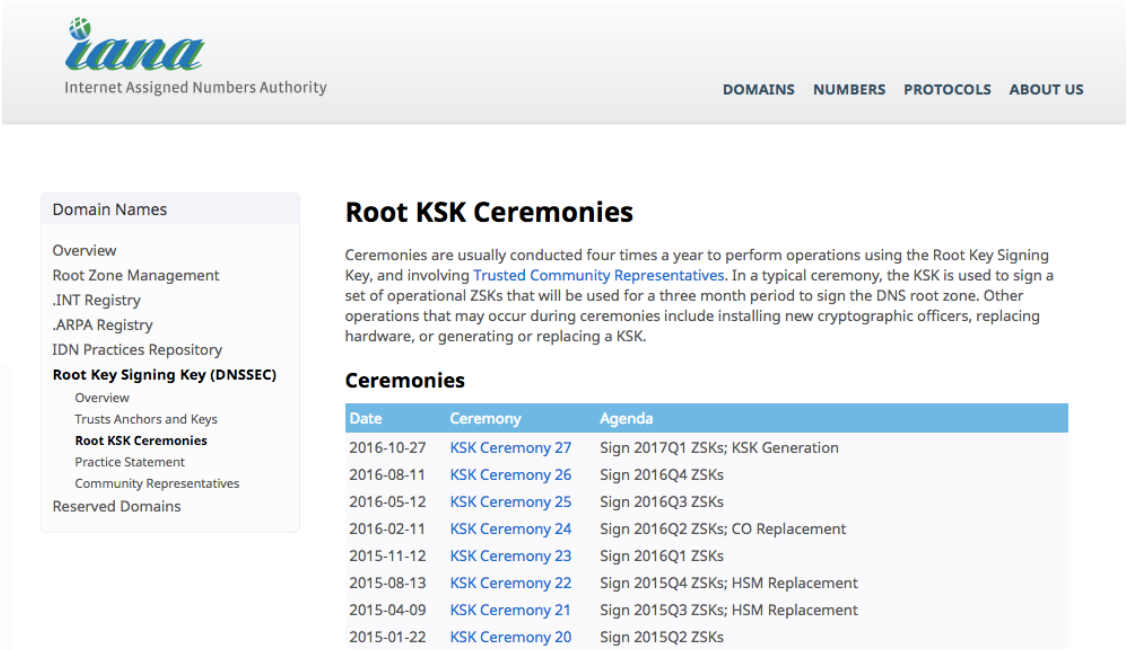
Such a scheme is called a *(k, n) threshold scheme.*

Efficient threshold schemes can be very helpful in the management of cryptographic keys. In order to protect data we can encrypt it, but in order to protect the encryption key we need a different method (further encryptions change the problem rather than solve it). The most secure key management scheme keeps the key in a single, well-guarded location (a computer, a human brain, or a safe). This scheme is highly unreliable since a single misfortune (a computer breakdown, sudden death, or sabotage) can make the information inaccessible. An obvious solution is to store multiple copies of the key at different locations, but this increases the danger of security breaches (computer penetration, betrayal, or human errors). By using a $(k, n)$ threshold scheme with $n = 2k-1$ we get a very robust key management scheme: We can recover the original key even when $\lfloor n/2 \rfloor = k-1$ of the $n$ pieces are destroyed, but our opponents cannot

[A.Shamir, *How to Share a Secret* (1979)]

# Real-Life Scenario: DNSSEC



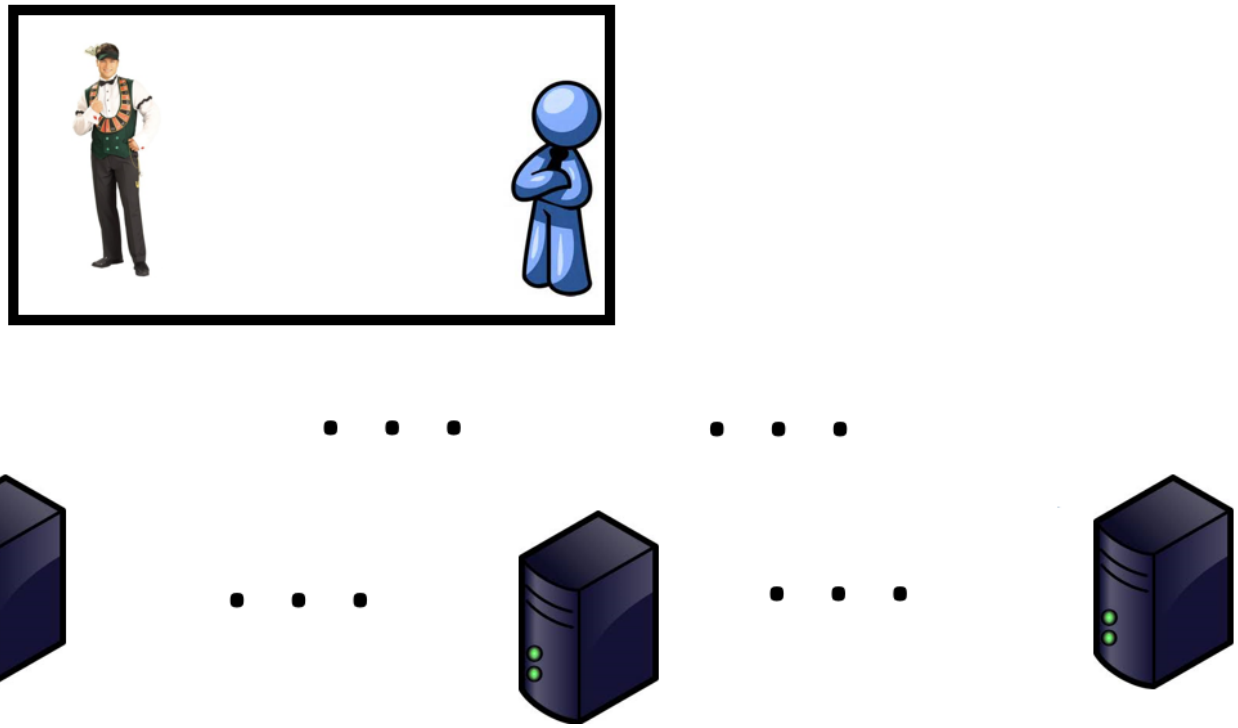




https://www.iana.org/dnssec/ceremonies

https://www.nanog.org/sites/default/files/1_Lewis_Rolling_the_Root_Zone_DNSSEC_Key_Signing_Key.pdf

https://www.youtube.com/watch?v=1LLHPnxQm-M

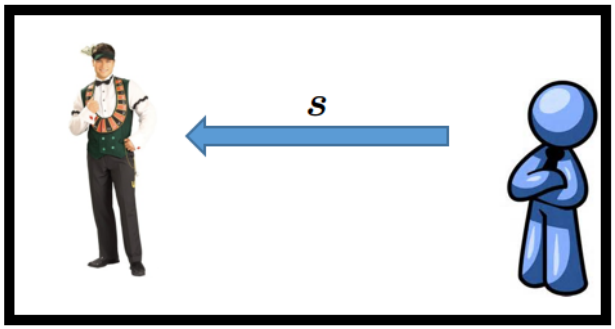
# Assumptions

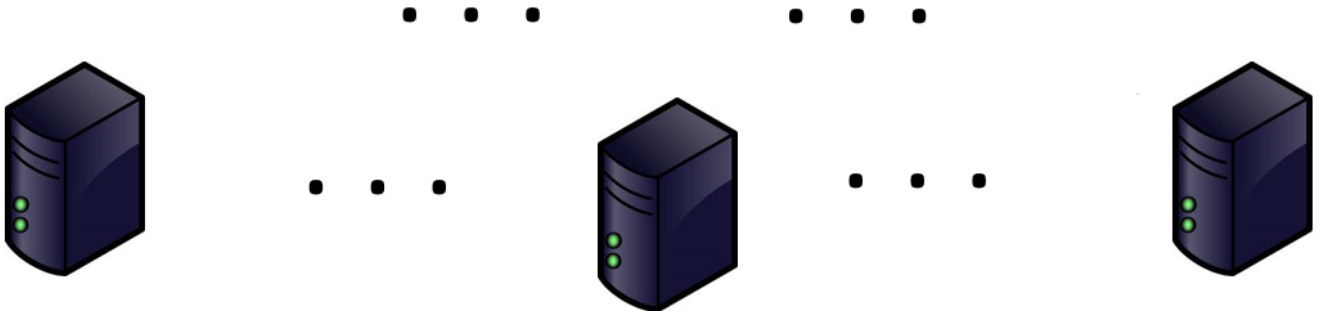


(1) decouple the user from the dealer

(2) the dealer only interacts with the user

# Assumptions

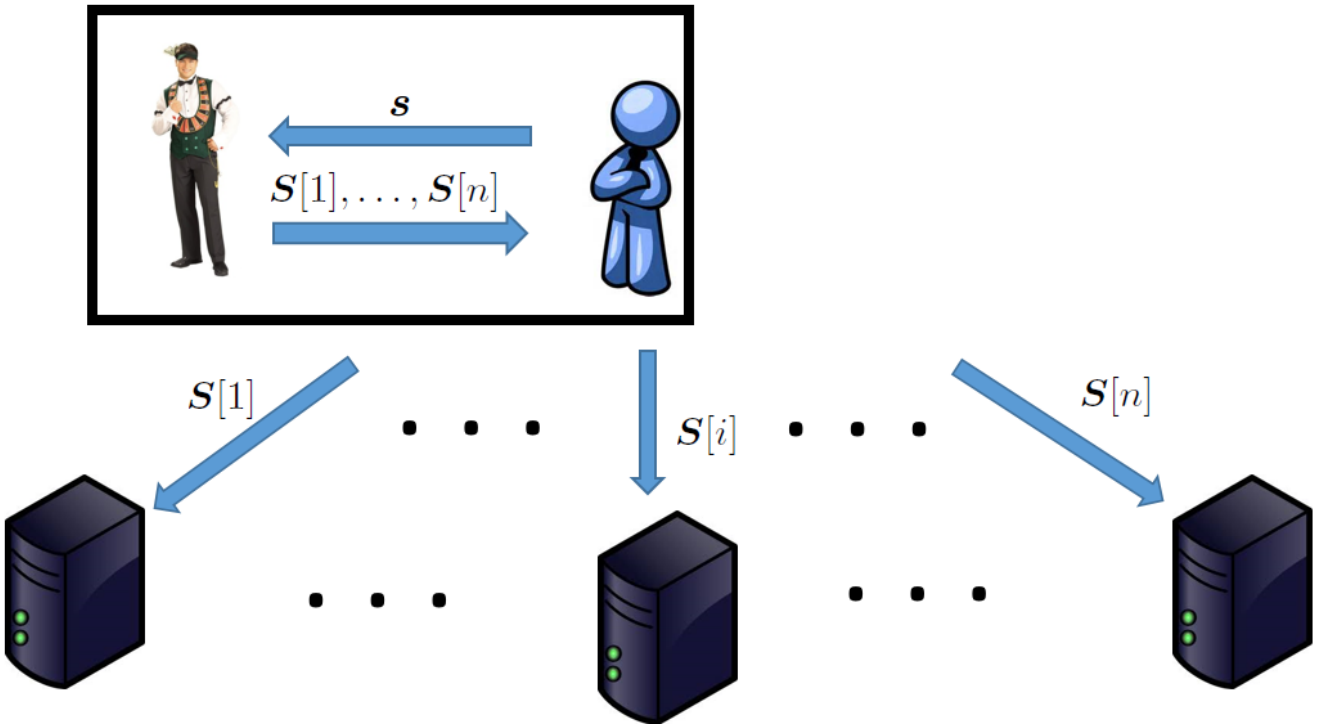


(1) decouple the user from the dealer

(2) the dealer only interacts with the user

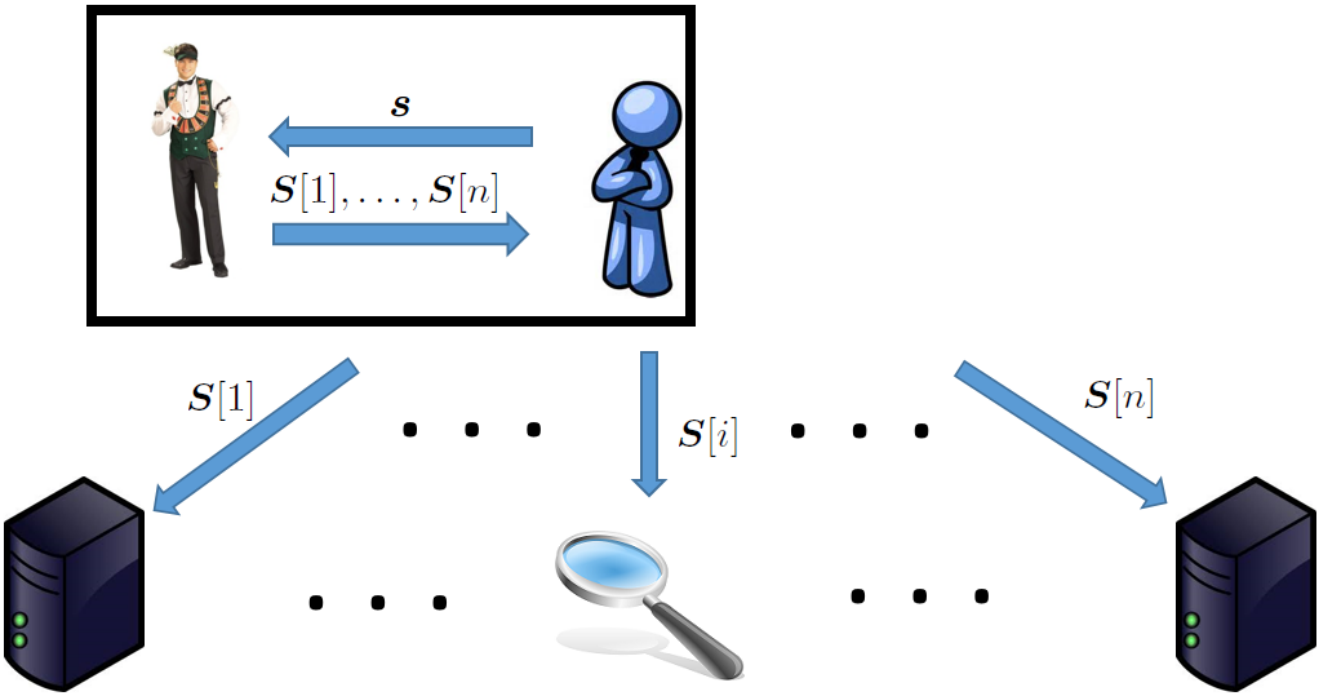
# Assumptions




(1) decouple the user from the dealer

(2) the dealer only interacts with the user

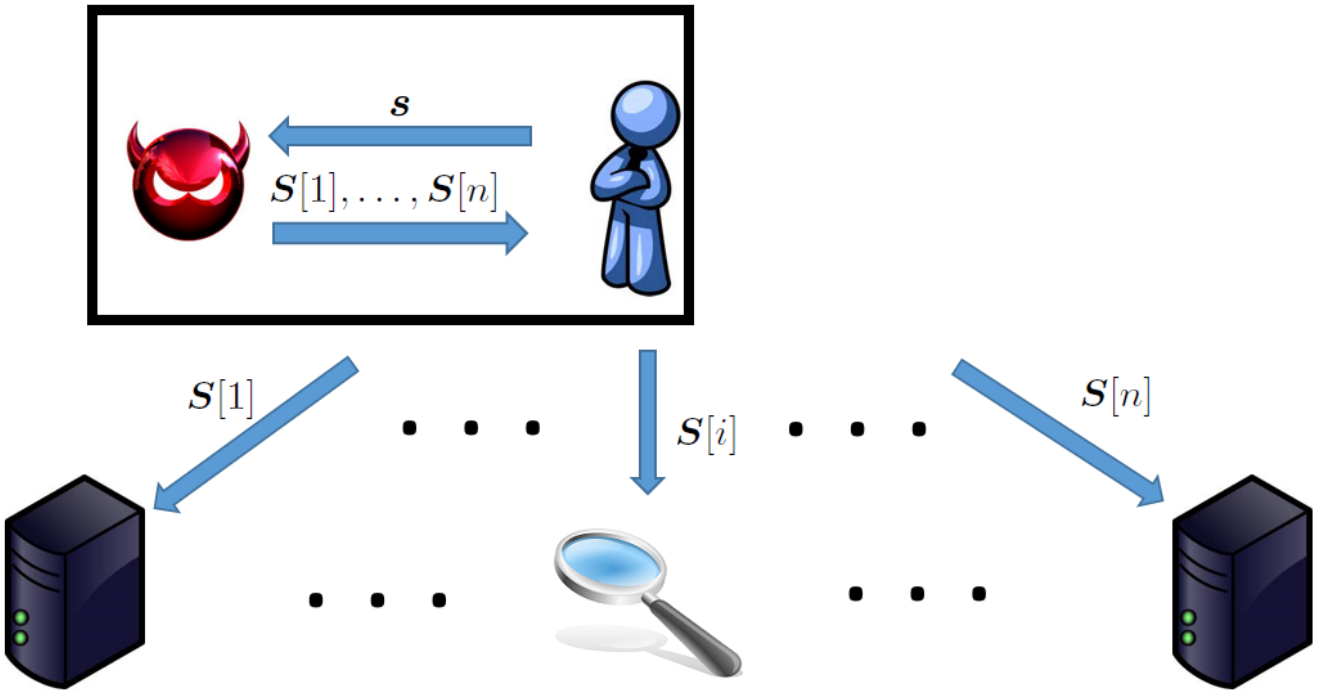
# Assumptions




(3) big brother controls some servers (not enough to reconstruct!)

(4) big brother might had previously interacted with the dealer

# Assumptions




(3) big brother controls some servers (not enough to reconstruct!)

(4) big brother might had previously interacted with the dealer

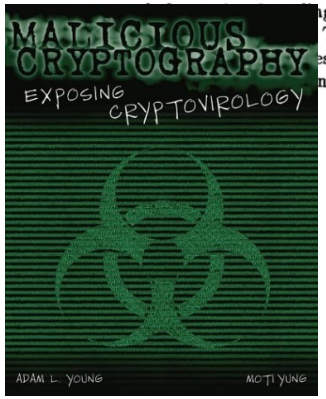
# Existing Work

**Kleptography:**
**Using Cryptography Against Cryptography**

Adam Young* and Moti Yung**

**Abstract.** The notion of a Secretly Embedded Trapdoor with Universal Protection (SETUP) has been recently introduced. In this paper we ...g information securely and subliminally from ... The SETUP mechanisms presented here, in ...es, leak secret key information **without** using ...nnel. This extends this area of threats, which

[EuroCrypt'97]




['04]

Encryption

Key Exchange

Signature Schemes

...

randomisation

**Security of Symmetric Encryption against Mass Surveillance**

Mihir Bellare[1], Kenneth G. Paterson[2], and Phillip Rogaway[3]

[1] Dept. of Computer Science and Engineering, University of California San Diego, USA. cseweb.ucsd.edu/~mihir
[2] Information Security Group, Royal Holloway, University of London, UK. www.isg.rhul.ac.uk/~kp
[3] Dept. of Computer Science, University of California Davis, USA. www.cs.ucdavis.edu/~rogaway

**Abstract.** Motivated by revelations concerning population-wide surveillance of encrypted communications, we formalize and investigate the resistance of symmetric encryption schemes to mass surveillance. The focus is on algorithm-substitution attacks (ASAs), where a subverted encryption algorithm replaces the real one. We assume that the goal of "big brother" is undetectable subversion, meaning that ciphertexts produced by the subverted encryption algorithm should reveal plaintexts to big brother yet be indistinguishable to users from those produced by the real encryption scheme. We formalize security notions to capture this goal and then offer both attacks and defenses. In the first category we show that successful (from the point of view of big brother) ASAs may be mounted on a large class of common symmetric encryption schemes. In the second category we show how to design symmetric encryption schemes that avoid such attacks and meet our notion of security. The lesson that emerges is the danger of choice: randomized, stateless schemes are subject to attack while deterministic, stateful ones are not.

[Crypto'14]



NTNU

# Security of Linear Secret-Sharing Schemes Against Mass Surveillance

- Based on the paper by -

Irene Giacomelli, Ruxandra F.Olimid , Samuel Ranellucci

Aarhus University, Denmark; University of Bucharest, Romania

*Special thanks to Samuel Ranellucci for kindly allowing me to build my presentation on top of the slides he had used for CANS`15.*

# Parties

User

Big Brother

Server

Dealer

Subverted dealer

Detector

# Goals

## User

wants to **hide** secrets from big brother

wants to **detect** if big brother is trying to learn the secret

might use a **detector** 

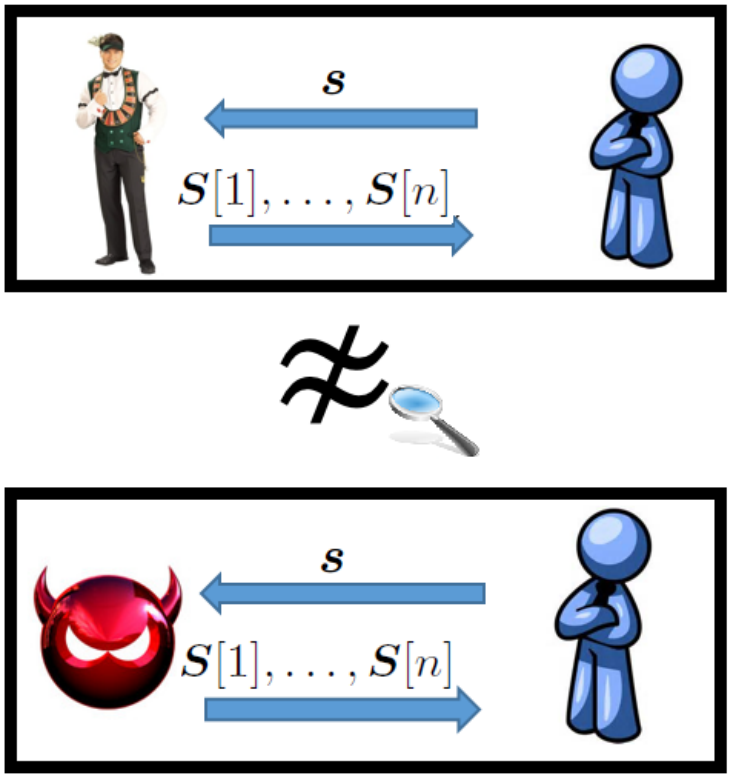
## Big Brother

wants to **learn** the user`s secret

wants to **hide** that he is trying to learn the secret

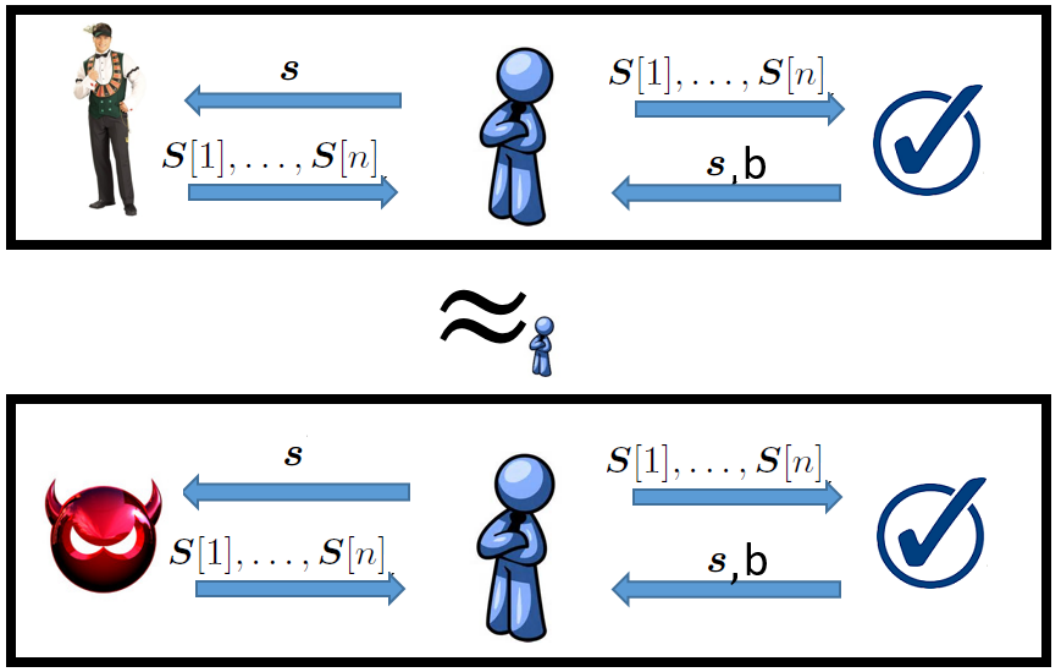
might previously **subvert** the dealer

# Successful Subversion




Surveillance

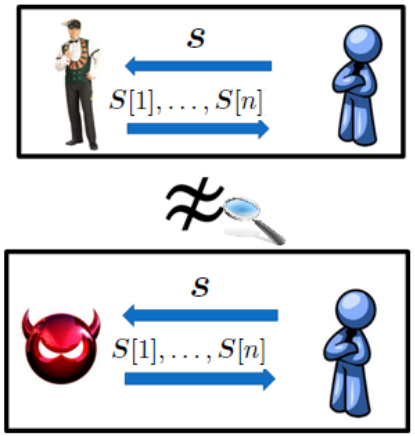
# Successful Subversion



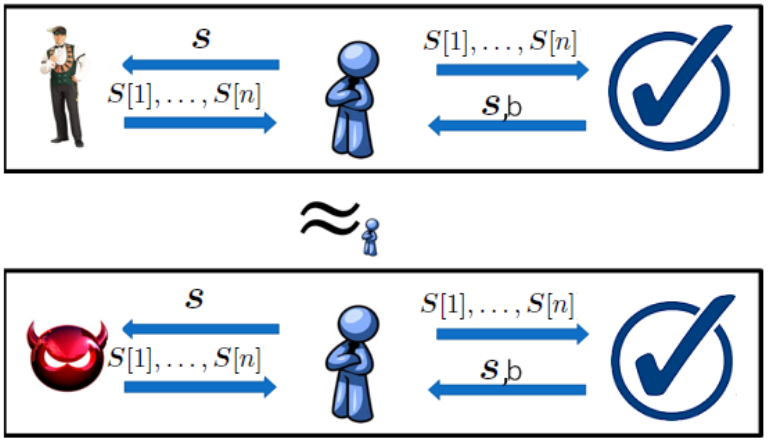

Undetectability

# Successful Subversion

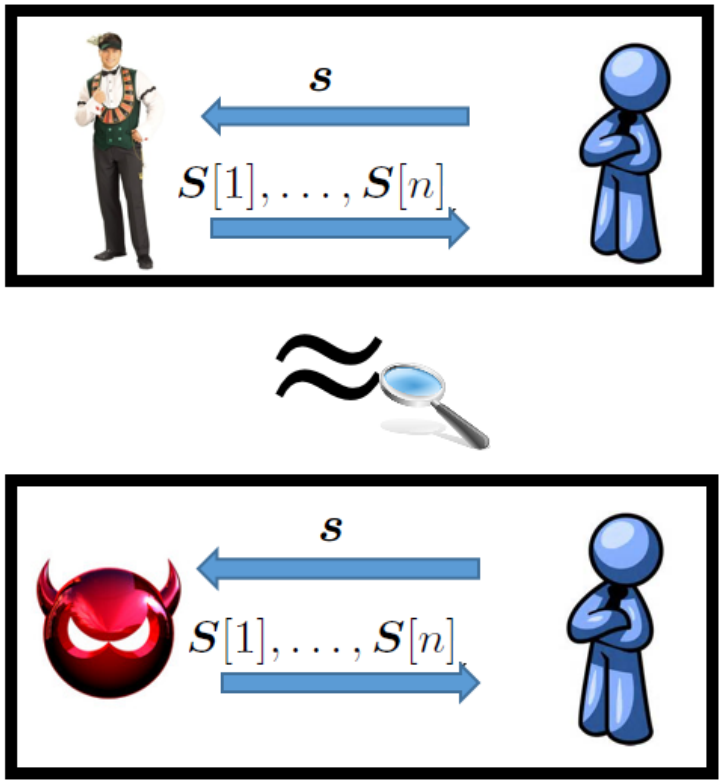
$\exists$  $\forall$ 




AND

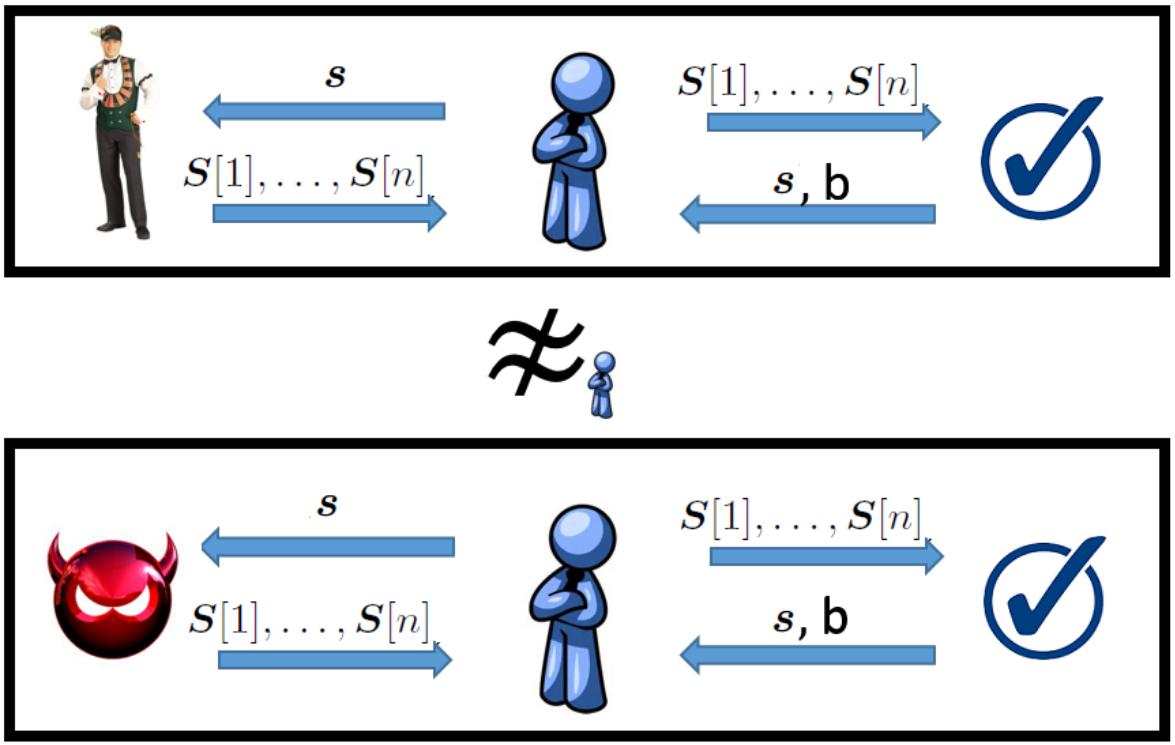


# Successful Resilience



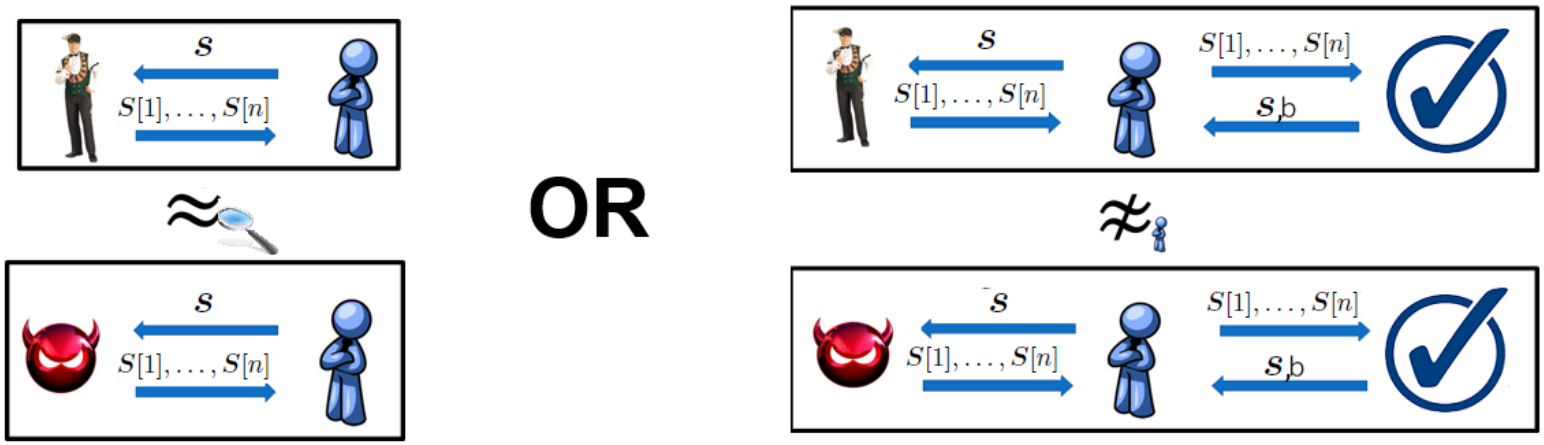

No surveillance

# Successful Resilience




Detectable subversion

# Successful Resilience

# Results

**Theorem**

*For any* $\Pi = (\mathrm{Sh}, \mathrm{Rec})$ *LSSS, there exists an undetectable subversion* $\widetilde{\Pi} = (\widetilde{\mathrm{Sh}}, \widetilde{Rec})$ *such that:*

- *big brother learns* $lsb(\mathbf{s}[1])$ *with probability* 1
- *if* $\gamma - l \geq 2$ *(this assures* $t \geq 2$*), big brother learns the first* $t - 1$ *components of* $\mathbf{s}$ *with probability* 1

Notations:

$\mathbf{s}$ (with $|\mathbf{s}| = l$): the secret

$\gamma$: largest cardinality of a minimal qualified set

$T$ (with $|T| = t$): the set of servers big brother controls

# Shares Replacement Attack

Subverted dealer:

- generates t shares using big brother`s PK such that:
  - big brother uses SK to reconstruct (part of) ***s*** from the ***t*** corrupted shares (surveillance)
  - the ***t*** shares are indistinguishable from shares generated by a honest dealer (undetectability)
- fixes the above shares and extends to the full set of shares

# Shares Replacement Attack (t>1)

$\widetilde{\mathrm{Sh}}(\mathbf{s}, \mathtt{ID}, \mathtt{PK}, \mathcal{T})$
$T \leftarrow \mathcal{T}$
$\mathbf{S}_T \twoheadleftarrow \mathcal{F}_\Pi(\mathbf{s}, T)$
$\mathbf{S} \leftarrow \widehat{\mathrm{Sh}}(\mathbf{s}, \mathbf{S}_T)$
**return** $\mathbf{S}$

$\mathcal{F}_\Pi(\mathbf{s}, T)$
$x \twoheadleftarrow \mathbb{F}$
$\mathbf{S}_T[i_1] \leftarrow \mathcal{E}(\mathtt{PK}, x)$
$\mathbf{S}' \leftarrow \mathrm{PRG}(x)$
**for** $j = 2 \ldots t$ **do**
$\mathbf{S}_T[i_j] \leftarrow \mathbf{s}[j-1] + \mathbf{S}'[j-1]$
**return** $\mathbf{S}_T$

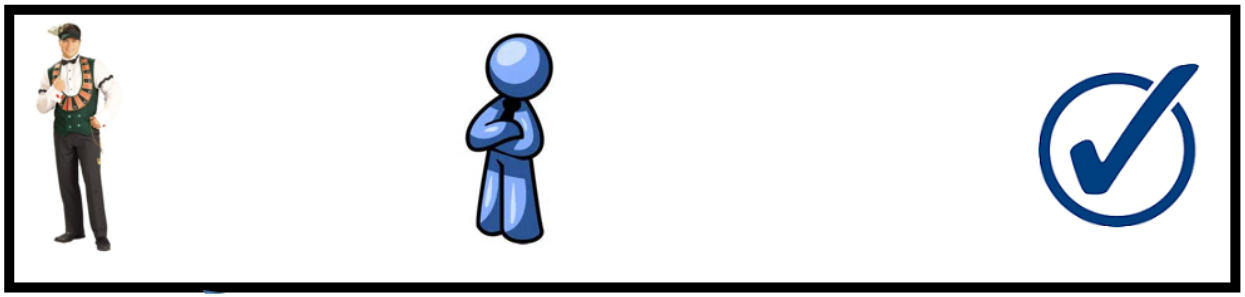
$\widetilde{\mathrm{Rec}}(\mathbf{S}_T, \mathtt{ID}, \mathtt{SK})$
$x \leftarrow \mathcal{D}(\mathtt{SK}, \mathbf{S}[i_1])$
$\mathbf{S}' \leftarrow \mathrm{PRG}(x)$
**for** $j = 2 \ldots t$ **do**
$\mathbf{s}[j-1] \leftarrow \mathbf{S}_T[i_j] - \mathbf{S}'[j-1]$
**return** $(\mathbf{s}[1], \ldots, \mathbf{s}[t-1])$

# Subversion Resilience

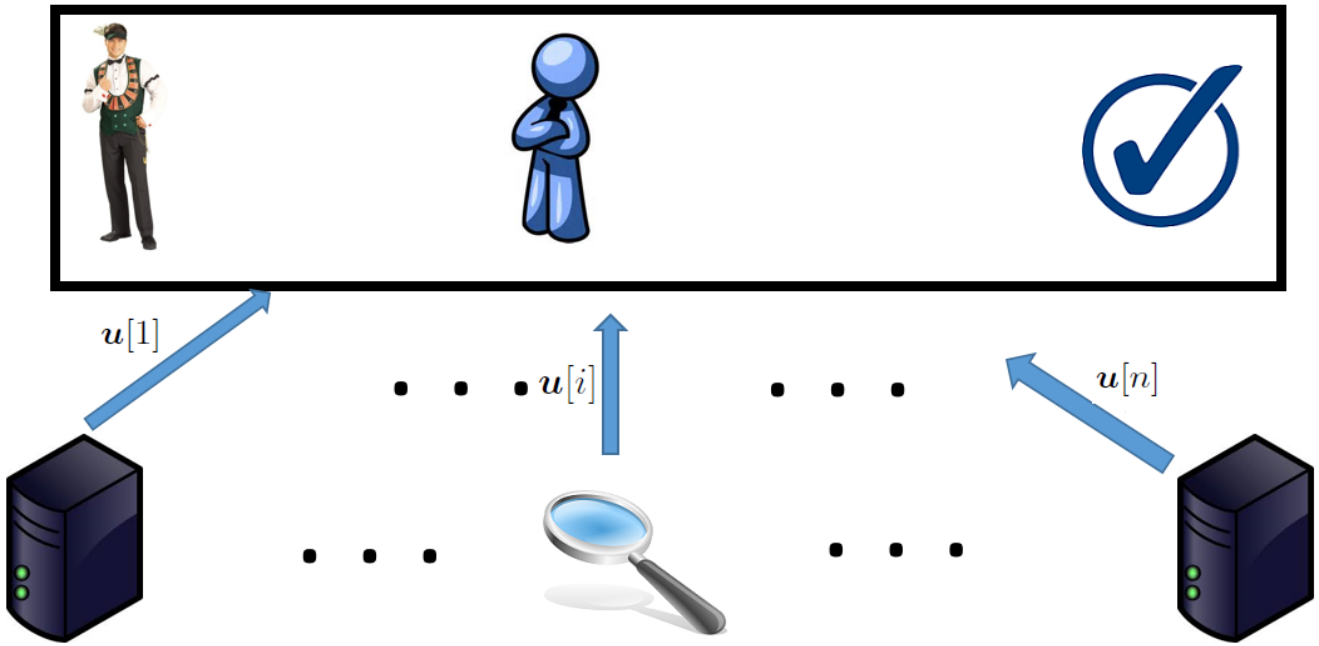
**Theorem**

*For any* $\Pi = (\mathsf{Sh}, \mathsf{Rec})$ *LSSS, there exists* $\Pi^* = (\mathsf{Sh}^*, \mathsf{Rec}^*)$ *a multi-input LSSS* $\widetilde{\Pi} = (\widetilde{\mathsf{Sh}}, \widetilde{\mathsf{Rec}})$ *that is subversion resilient.*

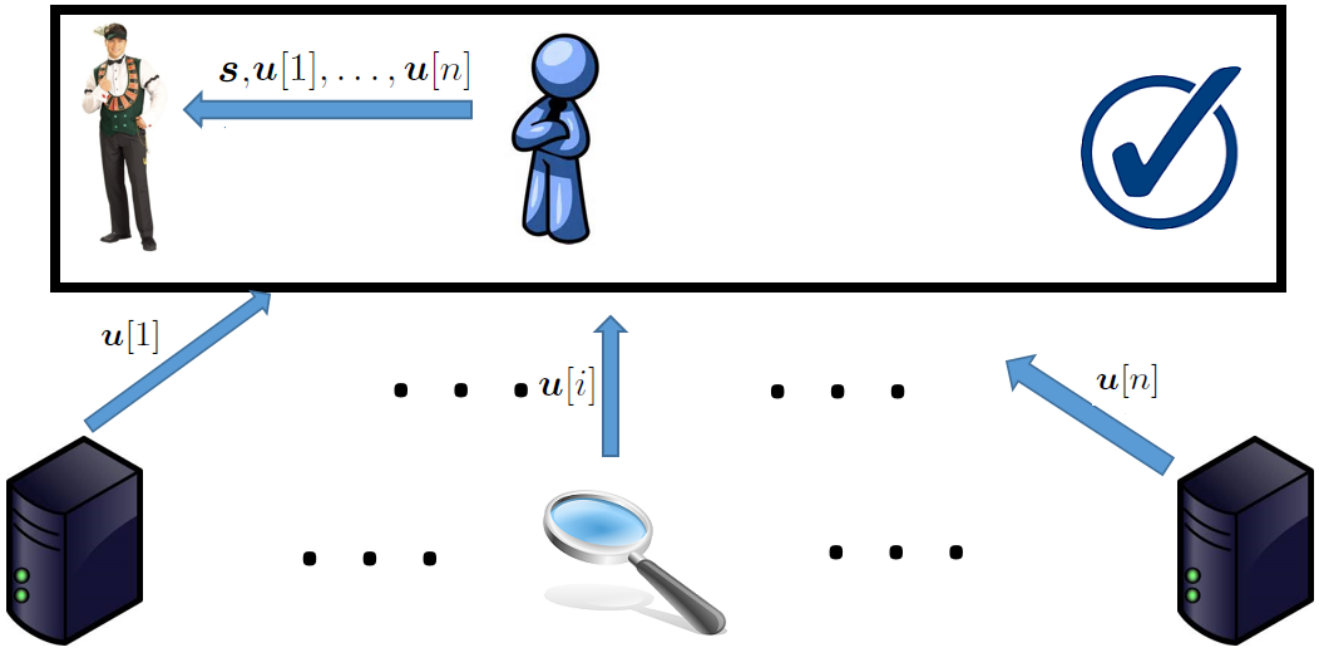
# Subversion Resilience



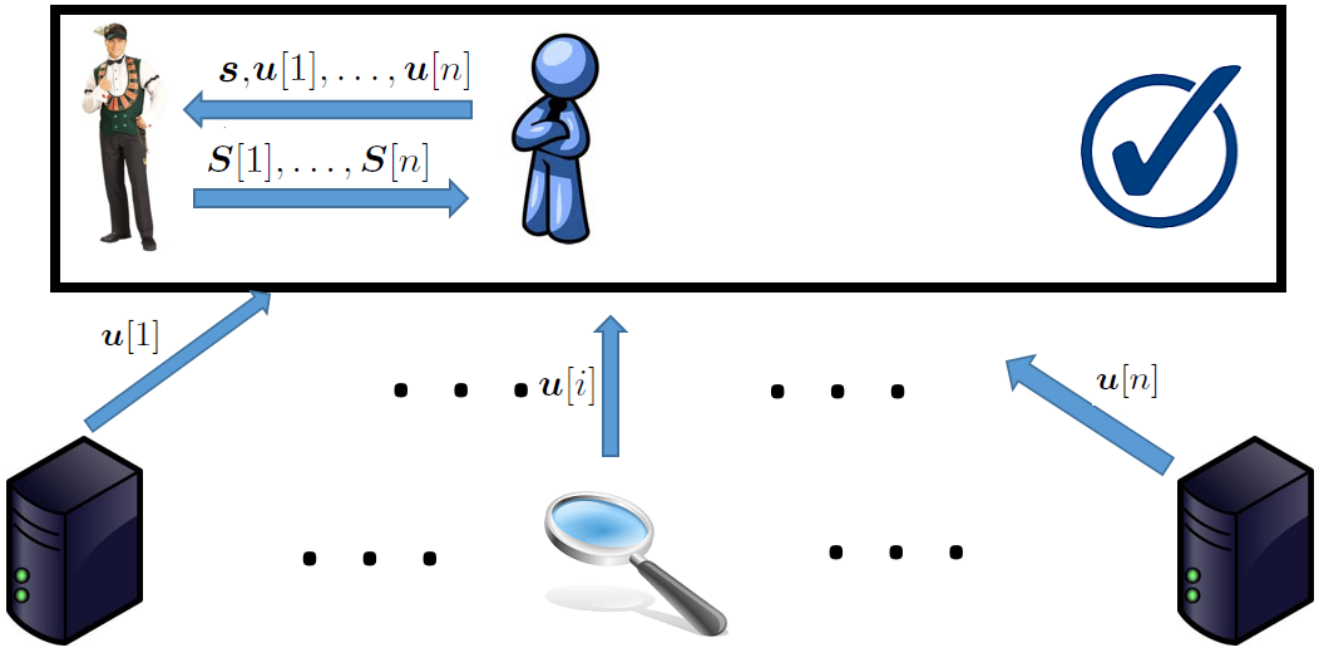
# Subversion Resilience



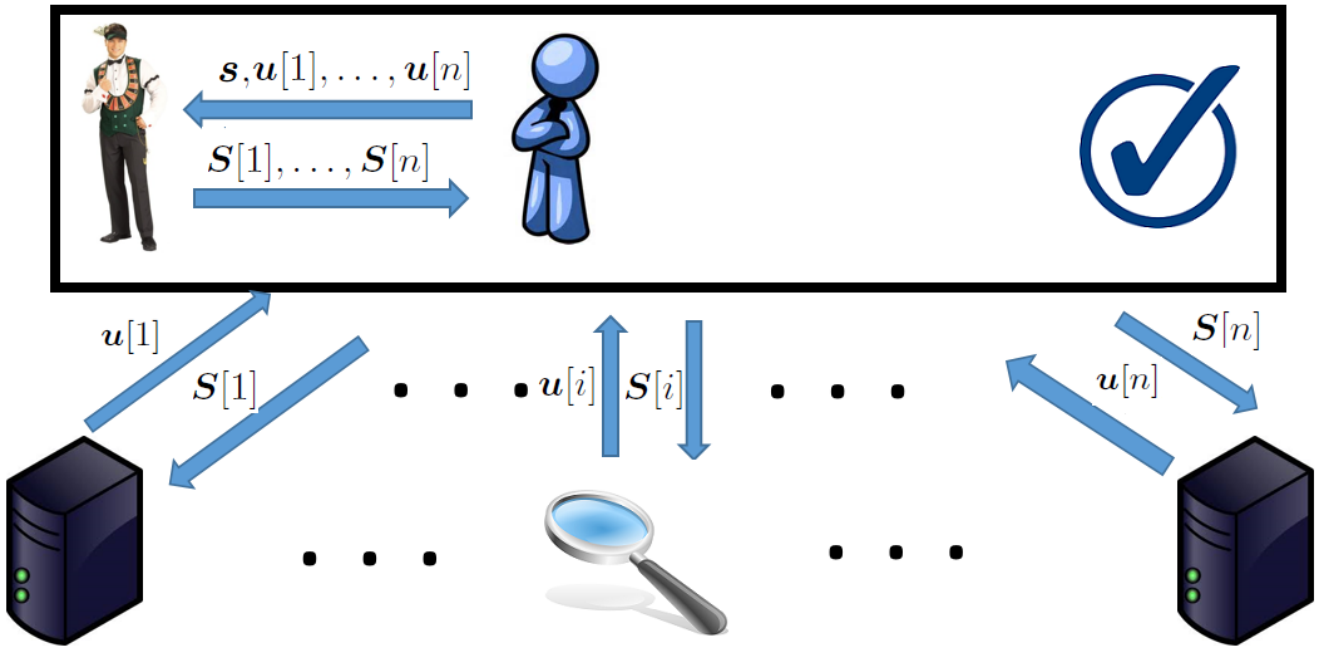
# Subversion Resilience



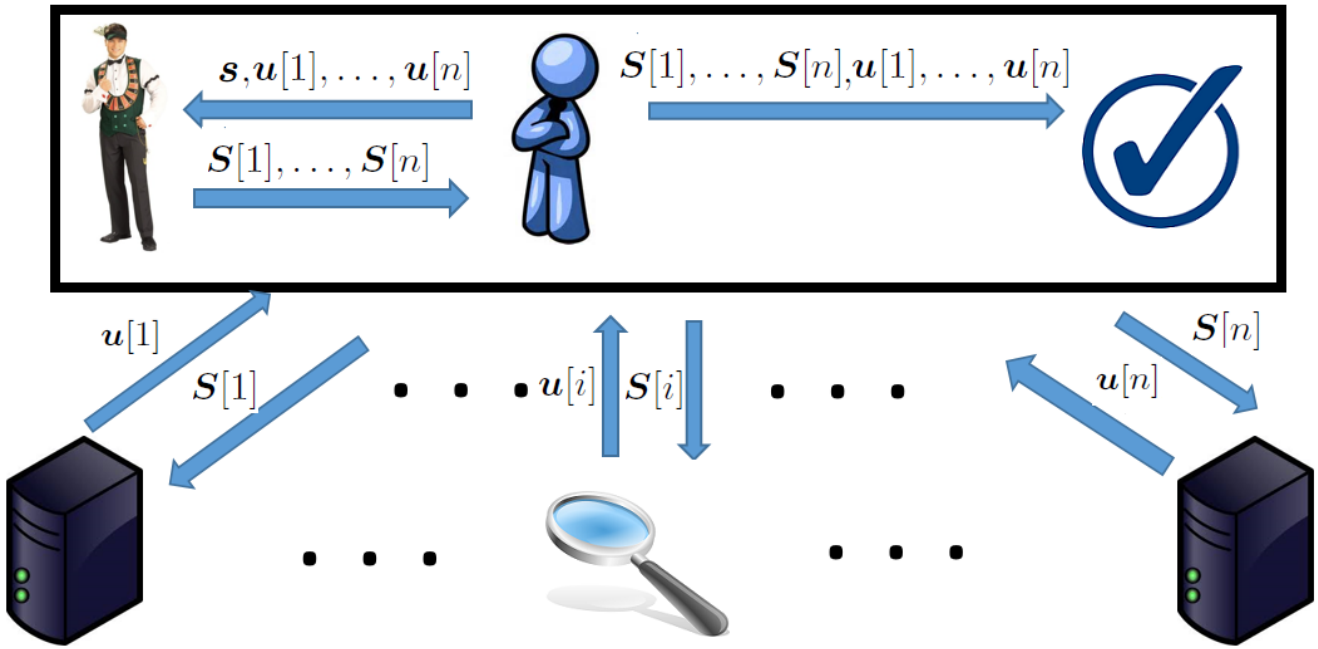
# Subversion Resilience



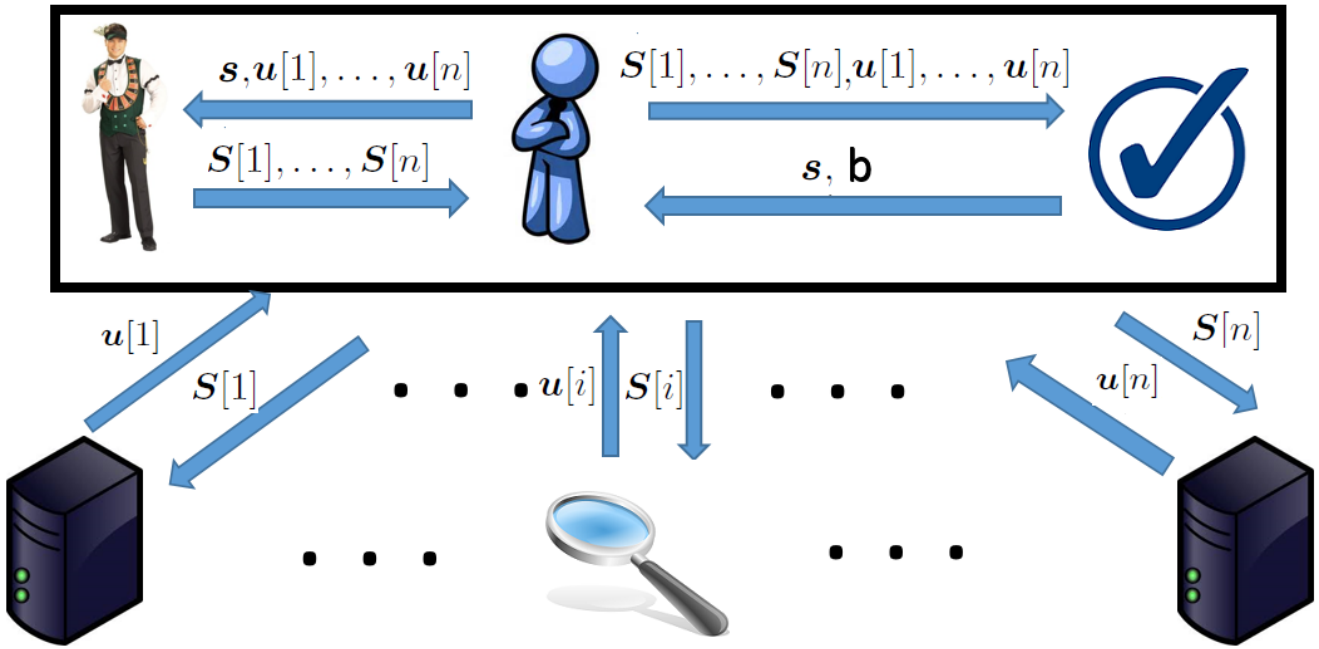
# Subversion Resilience



# Subversion Resilience



# Subversion Resilience



# Subversion Resilience

$\mathrm{Sh}(\mathbf{s}, \mathbf{u})$
- $\mathbf{r} \leftarrow \mathrm{PRG}(\mathbf{u}[1] \oplus \cdots \oplus \mathbf{u}[n])$
- $\mathbf{f}^T \leftarrow (\mathbf{s}, \mathbf{r})^T$
- $\mathbf{S} \leftarrow \mathbf{M} \cdot \mathbf{f}$
- **return** $\mathbf{S}$

$\mathrm{Rec}(\mathbf{S}_B)$
- **if** $B$ *is qualified* **then**
  - $\mathbf{s} \leftarrow \mathbf{N}_B \cdot \mathbf{S}_B$
- **else**
  - $\mathbf{s} \leftarrow \perp$
- **return** $\mathbf{s}$

# Thank you!