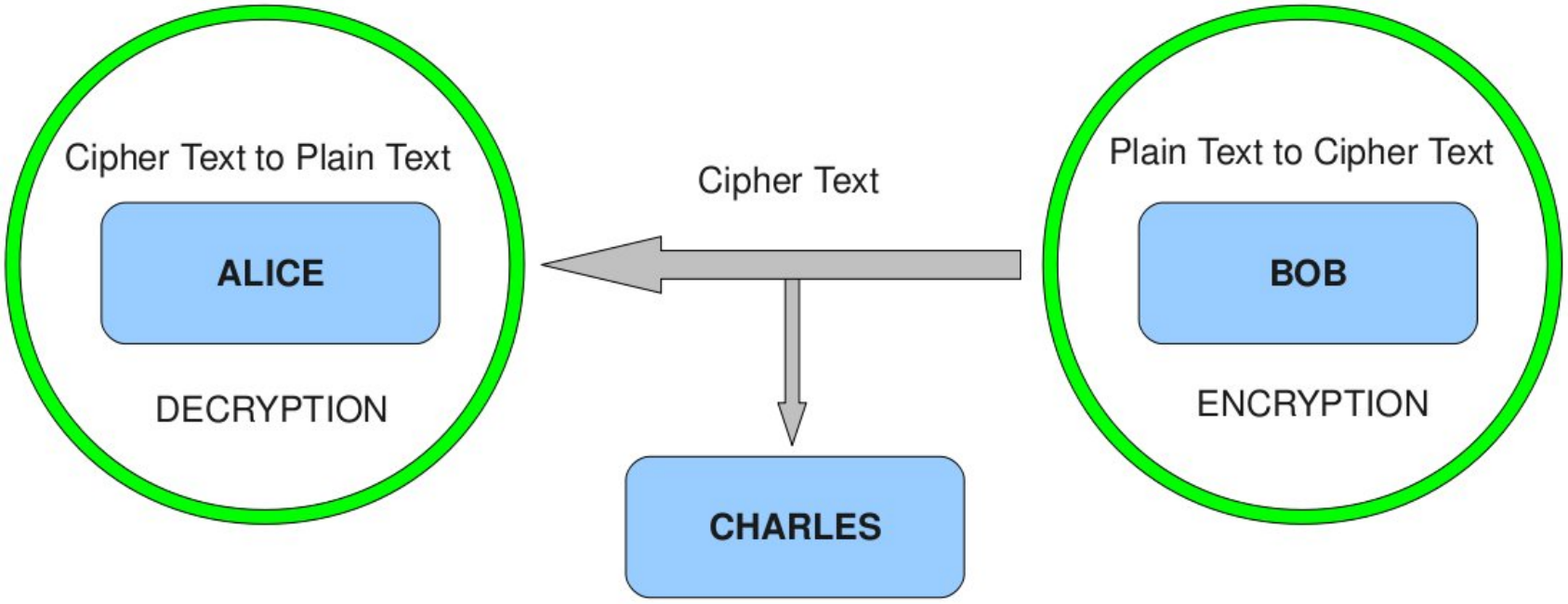
Some Problems in Cryptology

Bimal K. Roy

Indian Statistical Institute
203 Barrackpore Trunk Road
Kolkata 700 108, India

Cryptology – the art of secrecy



Encryption: $E_{k_1}(M) = C$ Decryption: $D_{k_2}(C) = M$

1. If k_1 and k_2 are known, all computations must be *easy*.
2. If k_1 and k_2 are unknown, then even if E, D, C are known, obtaining *any* information about M should be *difficult*!

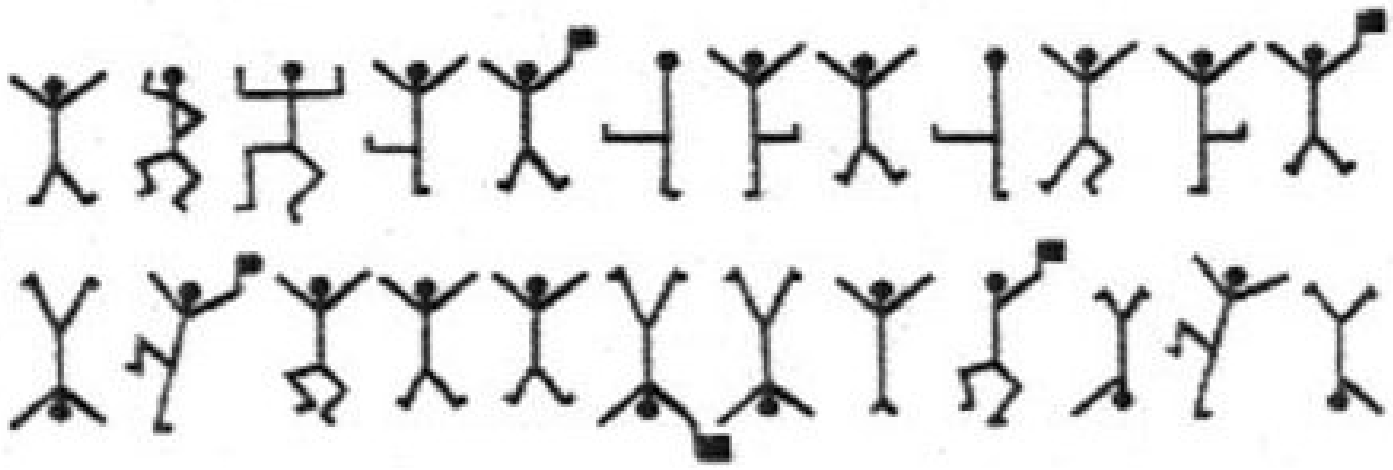
Secrecy without a key

400 BC, Greece

- Shave head of Spy
- Tattoo on Head
- Grow hair and travel

Secrecy with a key – Early days

Sherlock Holmes: The Adventure of the Dancing Men

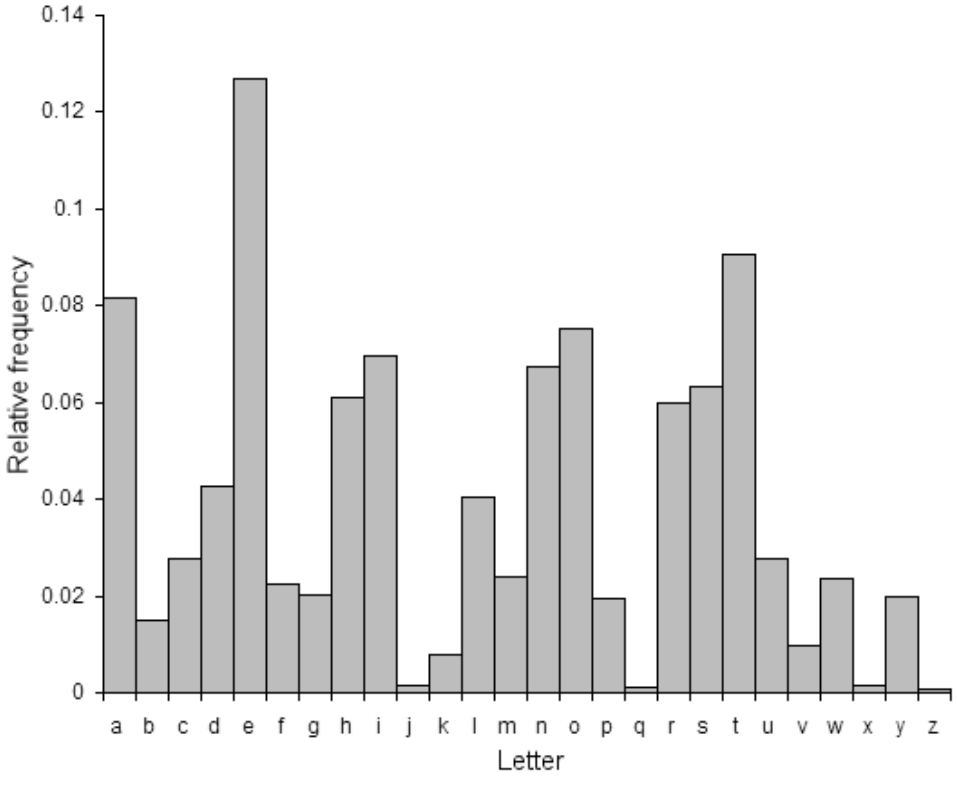


Substitution Cipher: Key is a *code book* for substituting letters in the plaintext alphabet with unique characters.

Is this a secure scheme?

Statistical attack on Substitution Cipher

Statistical frequency analysis on a large volume of ciphertext reveals the plaintext if the alphabet has characteristic patterns.



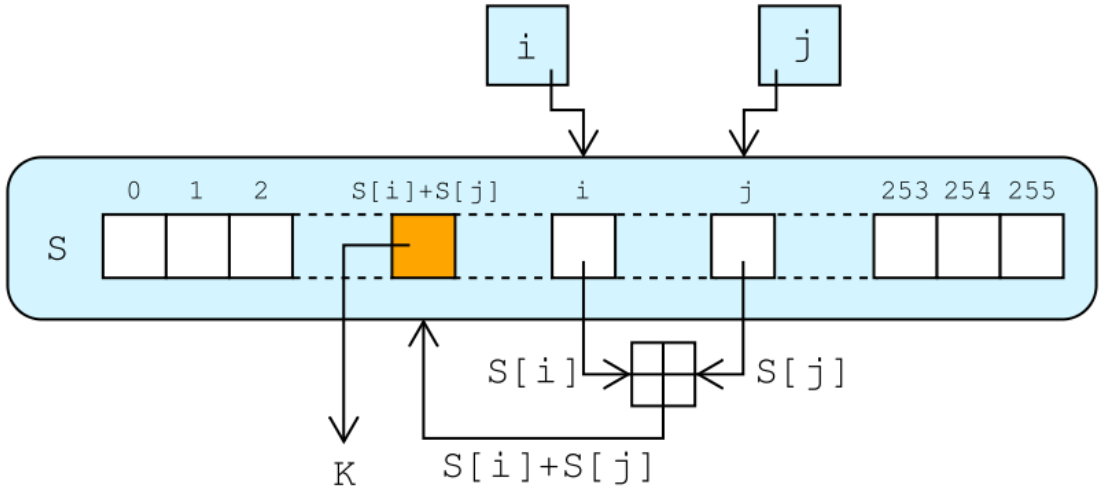
English

E = 12.7%
T = 9.1%
A = 8.2%
O = 7.5%
I = 7.0%
N = 6.7%

Secrecy with a key – Modern times

RC4: Rivest, 1987

ENC: $C = P \oplus K$
DEC: $P = C \oplus K$



Basic goal is to obtain a random stream of bytes K, by

1. creating a random permutation S of $\{0, \ldots, 255\}$,
2. and extracting random bytes from S thereafter

Does this really give a random stream of bytes?

Statistical attack on RC4

For a random stream of bytes (decimals 0 to 255), you expect the second output byte to be equal to 0 with probability $1/256$.

However, Mantin and Shamir proved: $P(z_2 = 0) \approx 2/256$

Broadcast attack: Suppose the same message M is sent to a lot of receivers, using RC4 with different keys each time. Thus,

$$C_i = M \oplus K_i = [m_1, m_2, m_3, \ldots] \oplus [z_{1i}, z_{2i}, z_{3i}, \ldots].$$

Second bytes of C_i are $[m_2 \oplus z_{2i}]$, where $P(z_{2i} = 0) \approx 2/256$

This reveals the message byte m_2 for enough ciphertexts!

Main tools for Cryptanalysis

Statistics

- Frequency analysis in case of Substitution Cipher
- Analysis of statistical bias in case of RC4

Combinatorics

- Combinatorial approach to find suitable paths in proving statistical biases in RC4, and other stream ciphers.

How do we safeguard our systems?

Strong systems

- Provable security: Build strong modes of operations and protocols using strong primitives which are based on reasonable and sound security assumptions.

Strong primitives

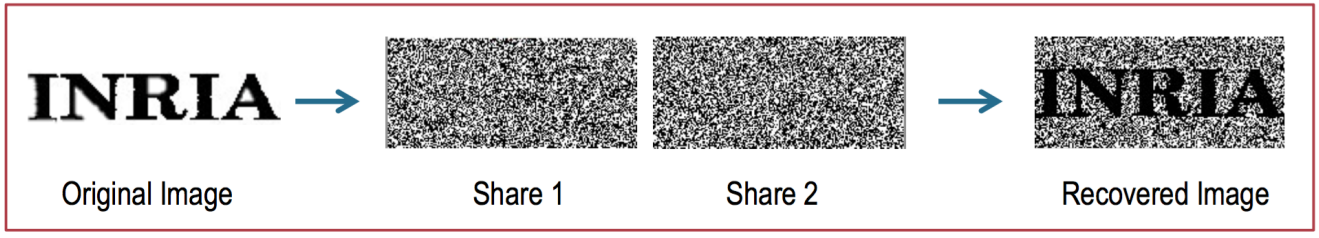
- Stream Ciphers: Pseudo-random bit generator (PRBG)
- Block Ciphers: Pseudo-random permutation (PRP)

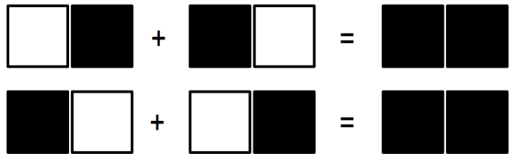
The security notion is to make the randomness of the stream and block ciphers *indistinguishable* from that of an unbiased coin tossed independently over arbitrarily many instances.

Visual Cryptography

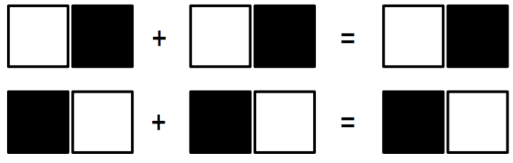
Conceptualised by Naor and Shamir, in 1994

- Secret sharing scheme with n participants, 1 secret image
- Secret image to be split into n shadow images called shares
- Certain qualified subsets of participants can recover the secret
- Other forbidden sets of participants have no information





Sharing for a single Black Pixel



Sharing for a single White Pixel

Problem Statement

Construct a (m, n) Visual Cryptography Scheme (VCS) such that

- There are n participants and 1 secret image
- Secret image to be split into n shadow images called shares
- Any m-subset of participants can recover the secret
- No t-subset of participants can recover the image if $t < m$

In particular, we will construct a $(2, n)$-VCS in this talk.

Metric: Relative Contrast

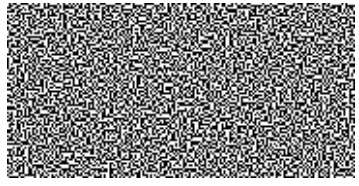
> *If* $(2, n)$*-VCS has basis matrices* S^0, S^1 *and pixel expansion* m*, then* relative contrast *for participants in subset* X *is given by* $\alpha_X(m) = \frac{1}{m}(w(S^1_X) - w(S^0_X))$.

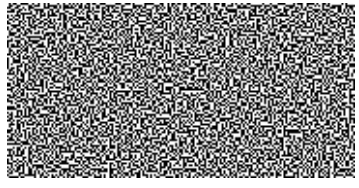
PBIBD applied to VCS

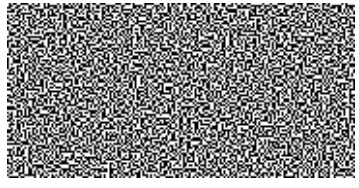
Visual outcome of $(6, 4, 2, 3, 0, 1)$-PBIBD to $(2, 6)$-VCS

Secret image: 

One Share

Share 1: 

Share 2: 

Share 6: 

Two Shares

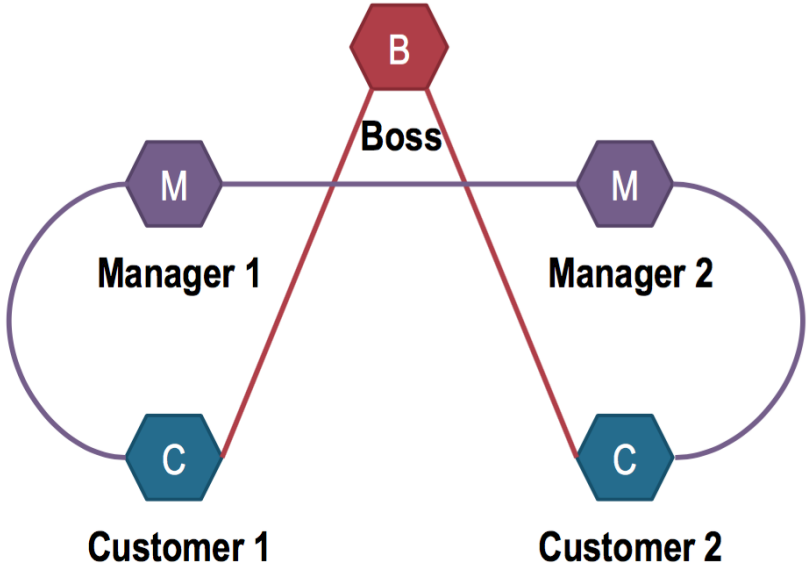
Shares 1 & 6: 

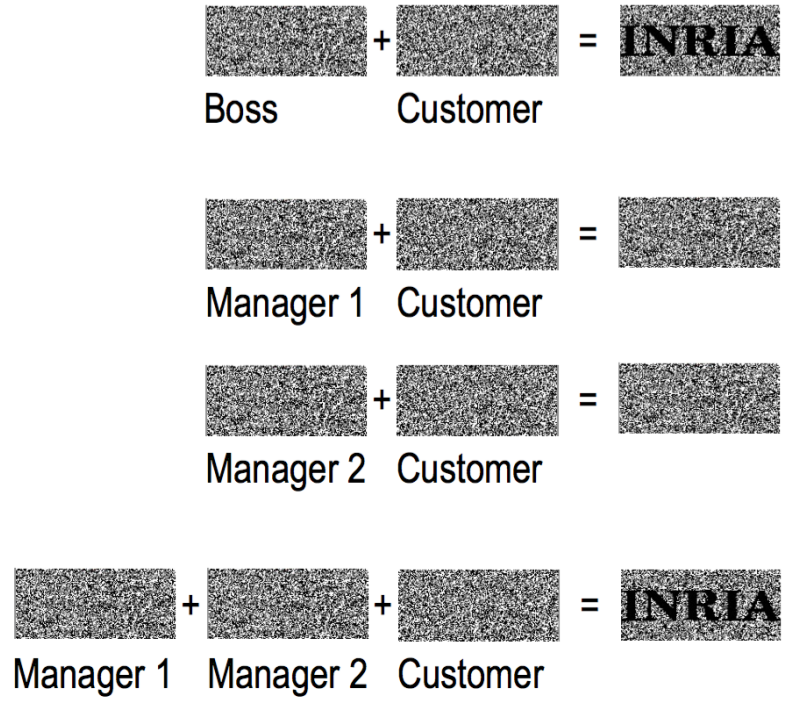
Shares 1 & 2: 

Relative contrast is
$\frac{1}{2}$ for 1 & 6 and $\frac{1}{4}$ for 1 & 2

VCS for Access Control

Secret is revealed *only* by the approved sets.
Example : {Boss + Customer} or {Both Managers + Customer}





Data Obfuscation

- Owner of a large database lends it for public use. The user is allowed to run restricted set of queries on data items.
- Owner's goal is to prevent the user from deriving any further information from the database, than what is derivable from the allowed set of restricted queries.

Data Obfuscation is a type of data masking where some useful information about the complete dataset remains even after hiding the individual sensitive information.

Data Obfuscation

The problem:

- User requires the original database to test applications.
- Owner requires privacy of certain columns (attributes).

Potential solution:

- Encrypt data of the private columns. It requires a short (128 bit, say) random key which remains secret with the owner.

Problem with traditional encryption modes is that they are not format preserving. For example, AADHAAR number 4580 5000 8000 encrypts to **** under 256-bit AES ECB mode. Thus, if the user application accessing the AADHAAR field has check and validation for 12-digit AADHAAR number, it simply fails.

Data Obfuscation

Format Preserving Encryption

- Mode of encryption where format of ciphertext is same as that of the plaintext. That is, the encryption behaves as a permutation on the domain of the plaintext.
- Example : 12-digit AADHAAR number maps to 12-digit AADHAAR number, or 16-digit credit card number maps to 16-digit credit card number.

Objectives of Data Obfuscation

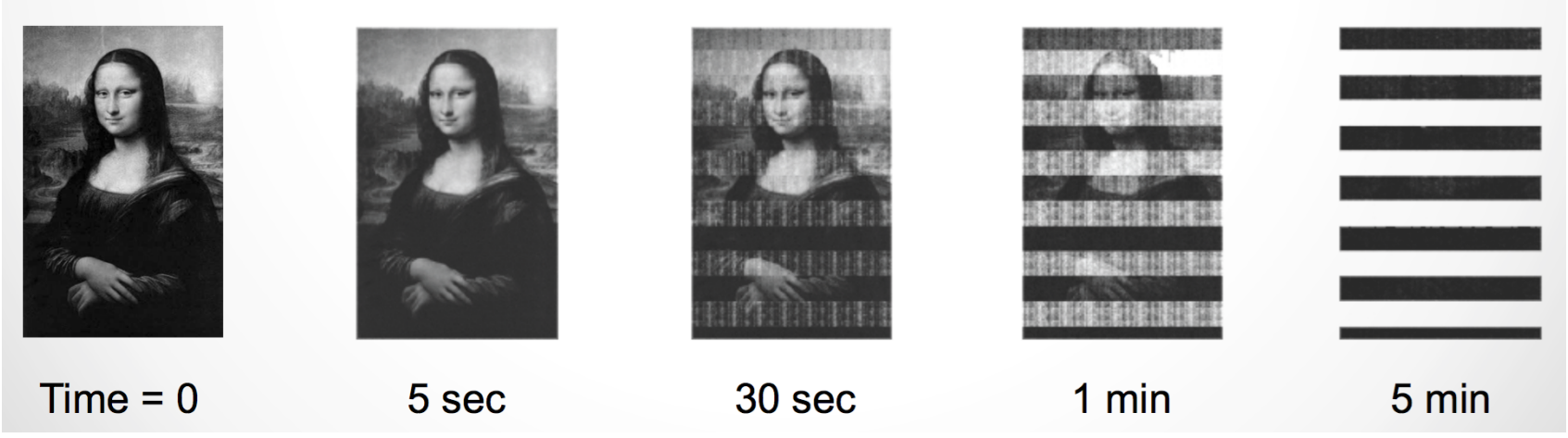
- Minimize risk of disclosure while providing access to the data.
- Maximize the analytical usefulness of the accessible data.

To understand cryptographic systems better, one needs to understand that operational platform of the algorithms

Here is where Engineering comes into the picture.

ColdBoot attack on RSA

Data remanence is a huge problem in cryptographic applications.
Example : Think of a Computer Memory that erases, but slowly.



Any form of residual cryptographic data may be sensitive!

ColdBoot attack on RSA

Idea of the attack

- RSA cryptosystem uses modulus $N = pq$ where the security depends on the hardness of factoring N.
- PKCS#1 standard for RSA mandates the storage of p, q and other RSA secret keys in the memory during operation.
- A clever attacker can retrieve partial information about the RSA secret keys from a decaying computer memory.

If you get about 30% bits of the primes p, q, you can factorize N.

Thank You