CSL759: Cryptography and Computer Security

Ragesh Jaiswal

CSE, IIT Delhi

Administrative information

- Course webpage:
  - www.cse.iitd.ac.in/~rjaiswal/2013/csl759

- Evaluation components:
  - Minor 1 and 2 exams: 15% each
  - Homework (2 - 3): 20%
  - Project: 20%
  - Major exam: 20%

- Reference material:
  - Mihir Bellare's slides and notes (available on the web).
  - Introduction to Modern Cryptography (Katz and Lindell).
  - Foundations of Cryptography (Oded Goldreich).
  - Other notes/slides/practice material on the web.

Administrative information

- Pre-requisites:
  - Basic probability theory
  - Algorithms
  - Comfortable in reading/writing rigorous mathematical proofs
- Lecture Timing:
  - To be decided.

Introduction

Introduction

- Throughout most of history:
  - Cryptography = **art** of secret writing
  - Secure communication

M

$M = D_K(C)$

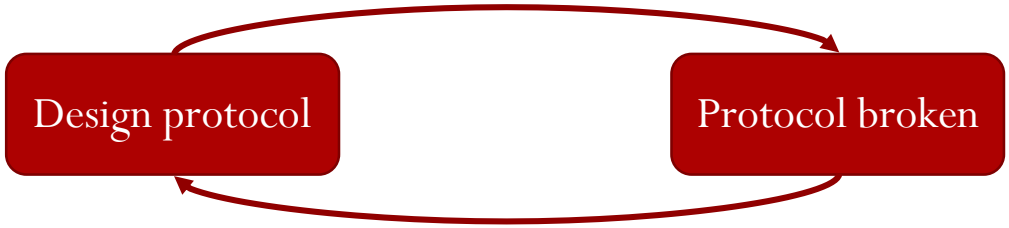
$C = E_K(M)$

K ⟷ K

Key exchange protocol

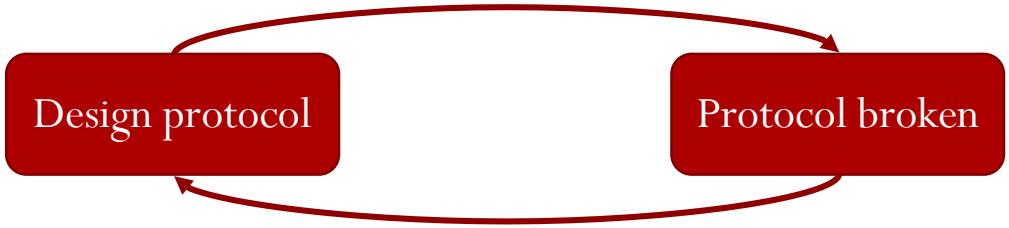
Introduction

- Early history (- early 70s):
  - Synonymous with secret communication.
  - Restricted to Military and Nobility.
  - More of *art* than rigorous science.



Introduction

- Early history (- early 70s):
  - Synonymous with secret communication.
  - Restricted to Military and Nobility.
  - More of *art* than rigorous science.



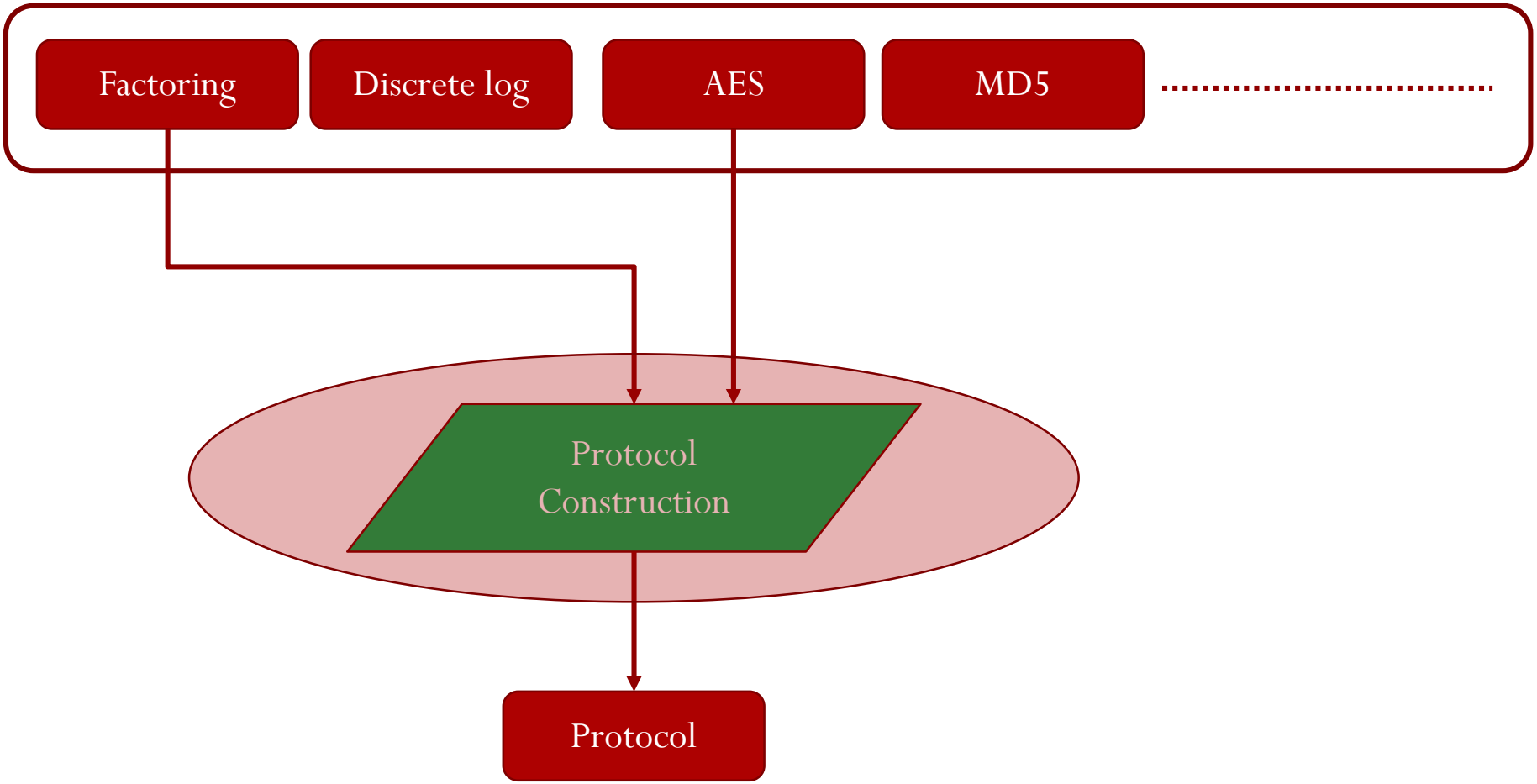
- Modern Cryptography:
  - Digital signatures, e-cash, secure computation, e-voting …
  - Touches most aspects of modern lifestyle.
  - Rigorous science:
    - *Reason about security of protocols*.

Introduction: Theme of this course

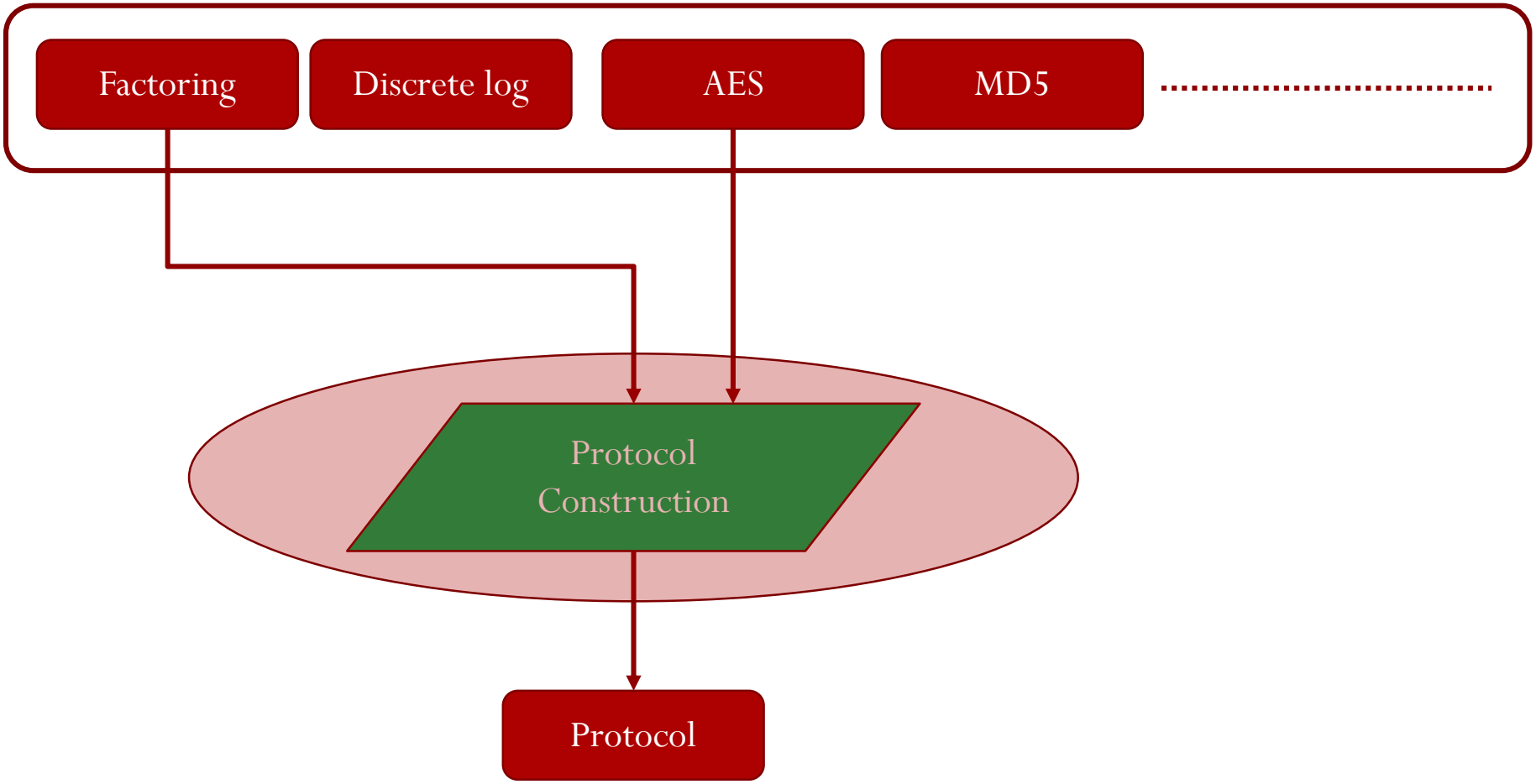
- Theme: Reason about security of protocols (**Provable security**)
  - Fix security goals and formalize the notion of security.
  - Construct a protocol.
  - Show that a successful attack as per the security notion results in a successful attack on an underlying problem that is believed to be hard to solve.

- What you should hope to learn in the course:
  - Learn basic cryptographic primitives and their interesting properties.
  - Reasoning about security of protocols.
  - Numerous applications/examples.

Introduction: Provable security



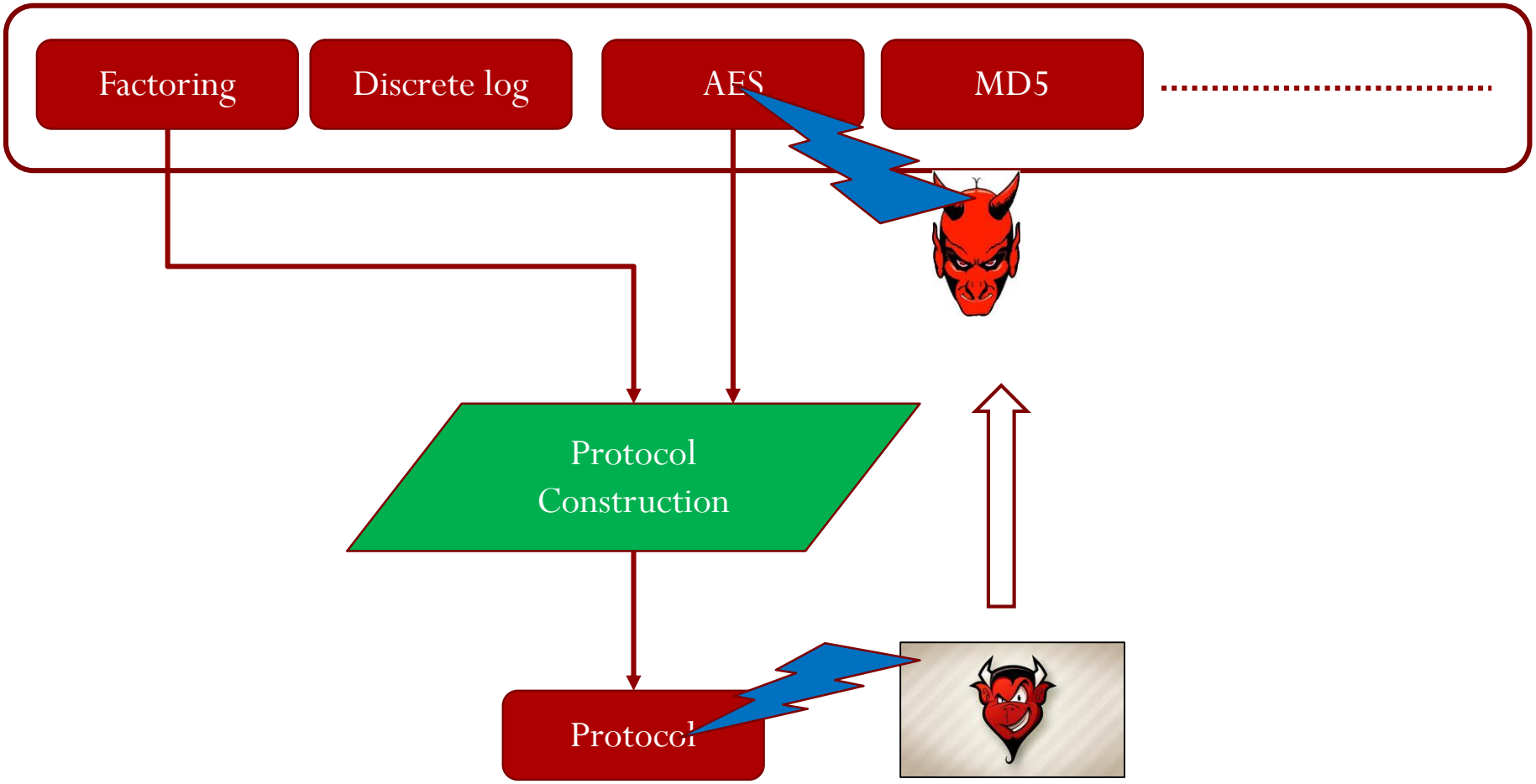
Introduction: Provable security



We would like to argue:

- If the basic primitive/problem is secure/hard, then the constructed protocol is "secure"

Introduction: Provable security



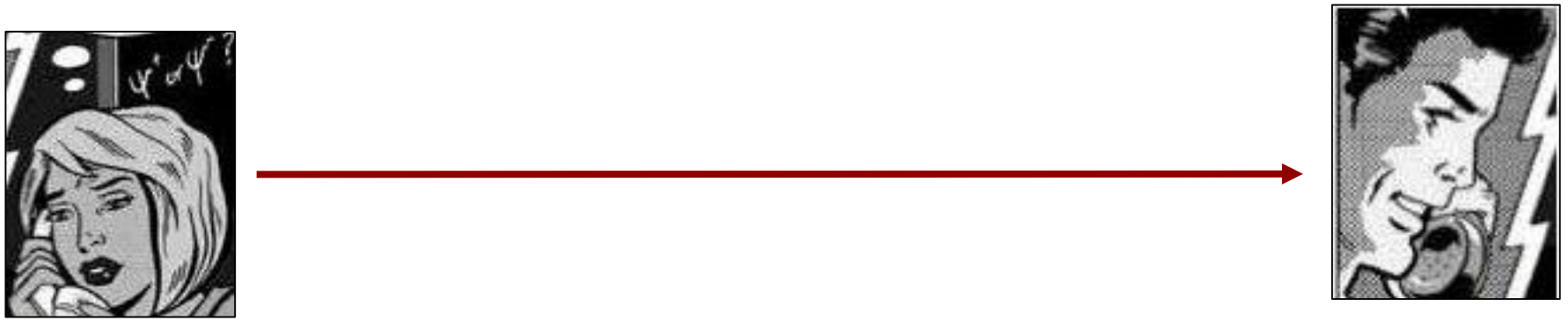
- :If there is an adversary that successfully attacks the protocol, then there is another adversary that successfully attacks/solves at least one of the basic primitives/problems.

Introduction

Secure communication

Introduction: Secure communication

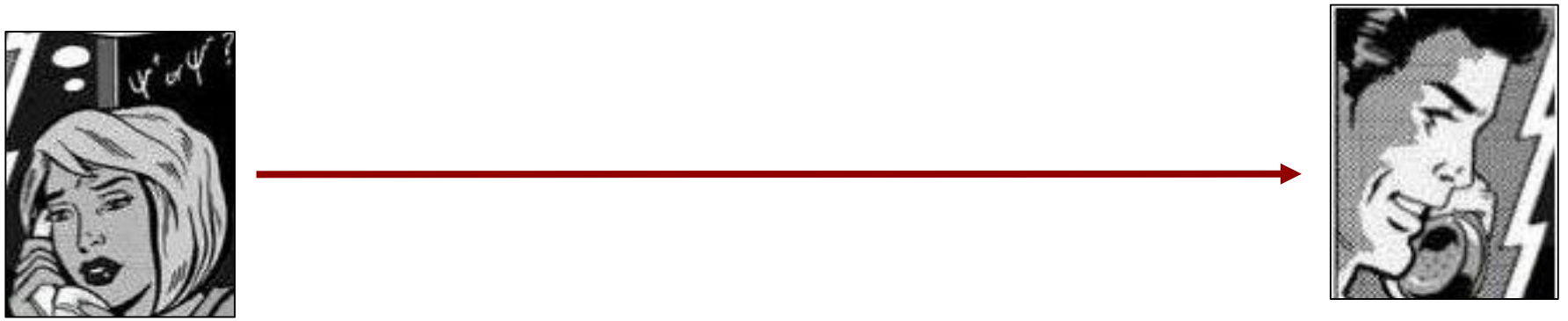
- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (Ceaser Cipher): Substitute each letter with the letter that is the αth letter after the letter in the sequence AB…Z
- Example ($\alpha = 2$): SEND TROOPS →

Introduction: Secure communication

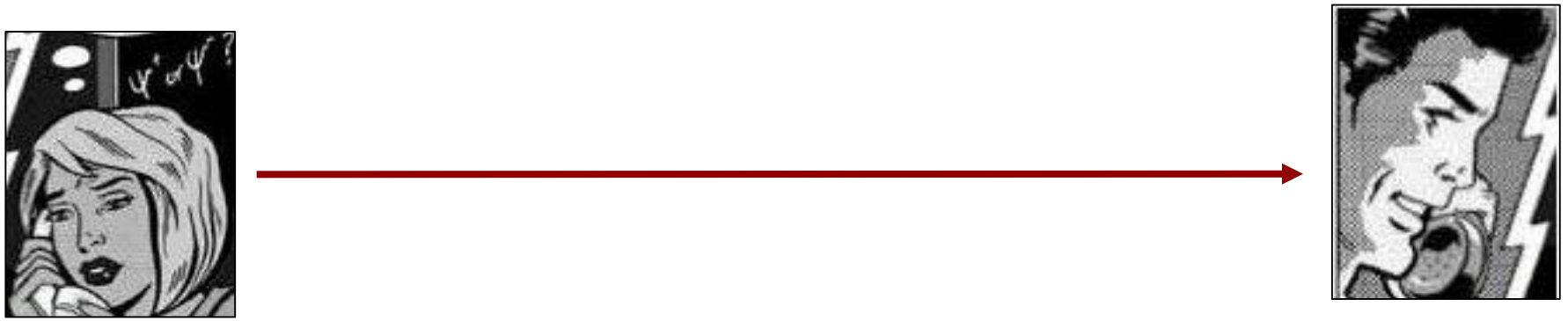
- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (Ceaser Cipher): Substitute each letter with the letter that is the αth letter after the letter in the sequence AB…Z
- Example ($\alpha = 2$): SENDTROOPS → UGPFVTQQRU

Introduction: Secure communication

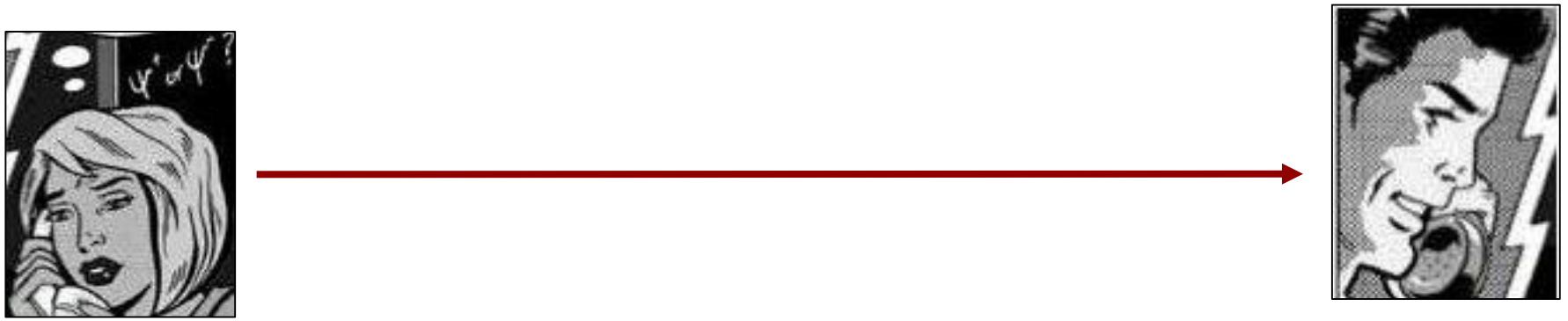
- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (Ceaser Cipher): Substitute each letter with the letter that is the αth letter after the letter in the sequence AB…Z
- Security was based on the fact that the encryption algorithm was a secret (Security through obscurity)

Introduction: Secure communication

- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.

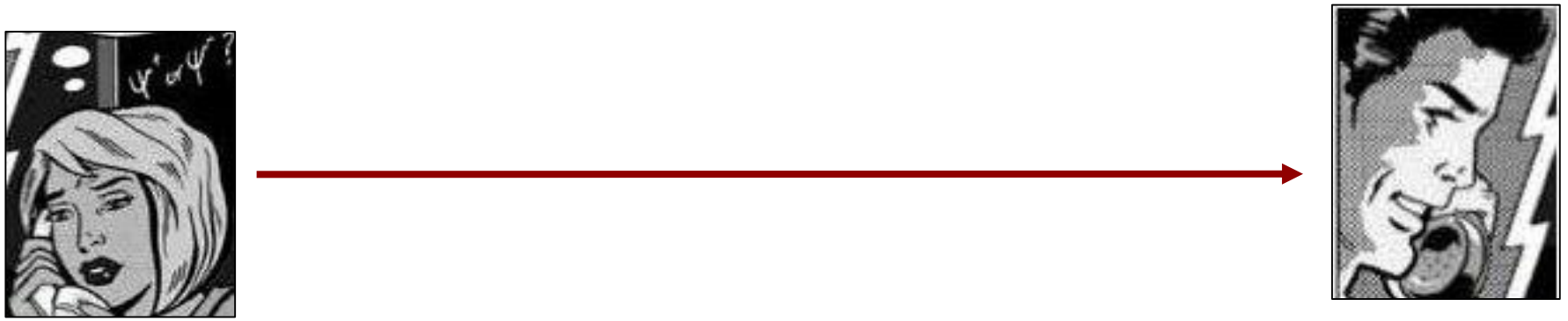


- Simple idea (Ceaser Cipher): Substitute each letter with the letter that is the αth letter after the letter in the sequence AB…Z
- ~~Security was based on the fact that the enc was a secret (Security through obscurity)~~

- Should be avoided at all cost!
- Algorithm should be public and security should come from secret keys.

Introduction: Secure communication

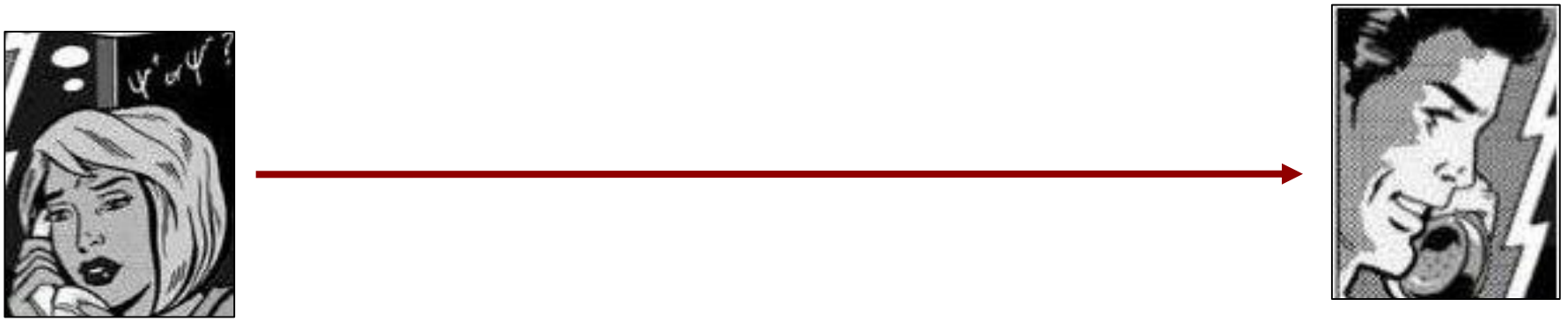
- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (Ceaser Cipher): Substitute each letter with the letter that is the αth letter after the letter in the sequence AB…Z
- Suppose we make the algorithm public and use the secret key as α. Can you break this protocol?

Introduction: Secure communication

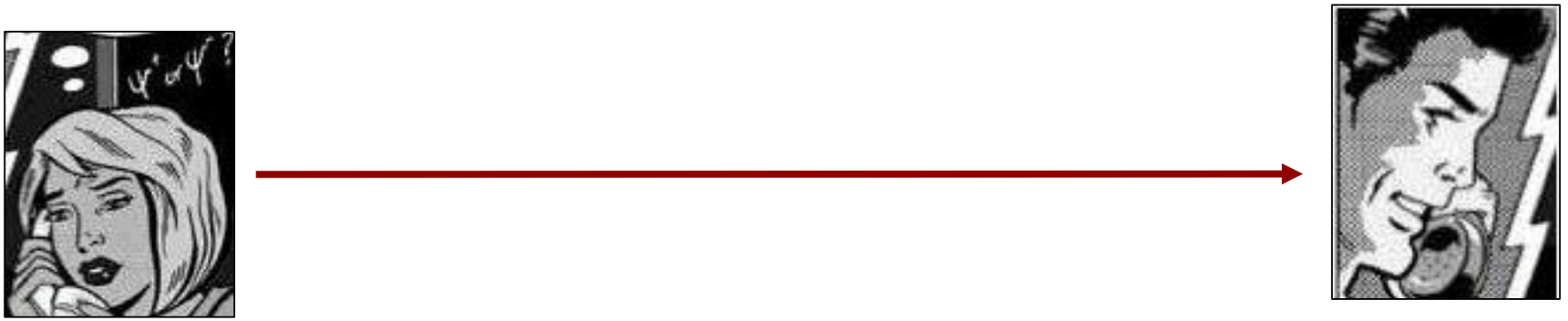
- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (Substitution Cipher): Let π be a permutation of the English letters. Substitute each letter α with the letter $\pi(\alpha)$. π acts as the secret key.
- Example: Let $\pi(A) = U, \pi(B) = T, \pi(C) = P$, ...then encryption of CAB is PUT.

Introduction: Secure communication

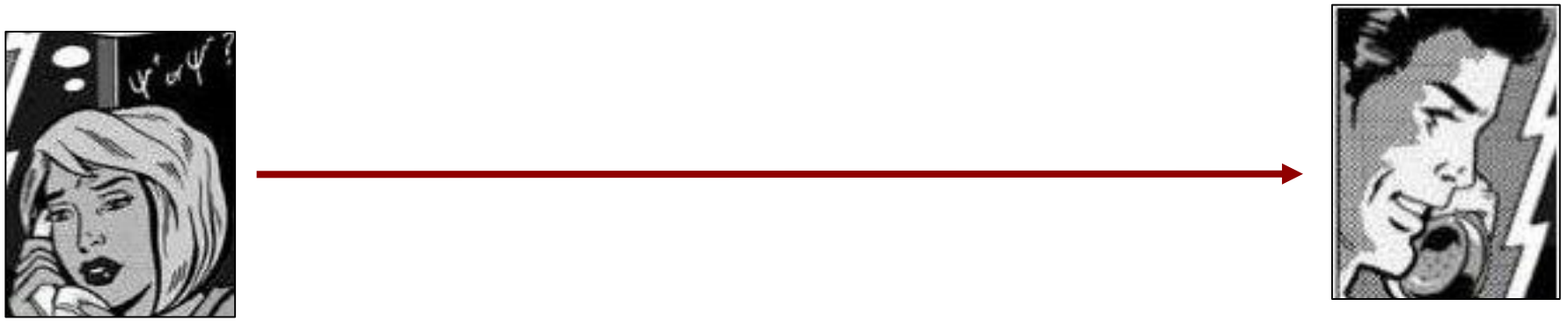
- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (Substitution Cipher): Let π be a permutation of the English letters. Substitute each letter α with the letter $\pi(\alpha)$. π acts as the secret key.
- Question: How much space you need to use to store the secret key?

Introduction: Secure communication

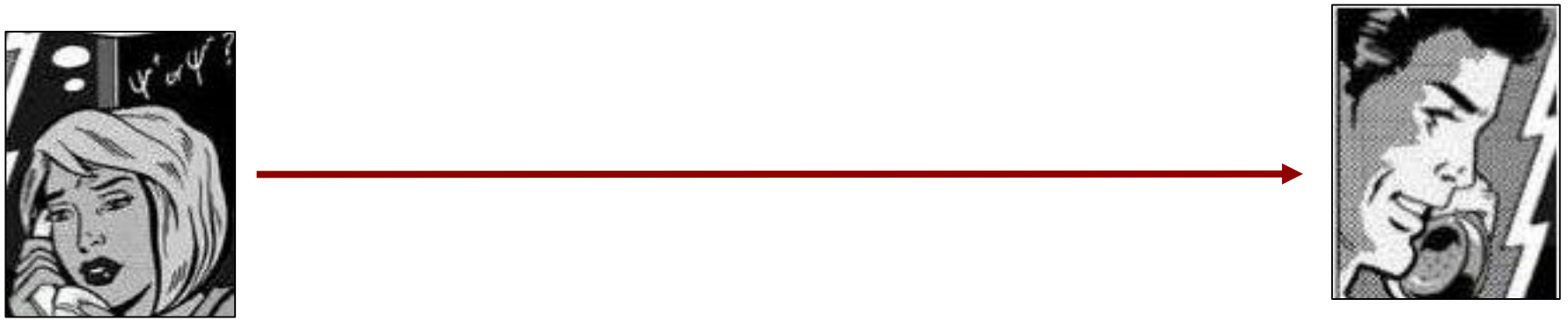
- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (Substitution Cipher): Let π be a permutation of the English letters. Substitute each letter α with the letter $\pi(\alpha)$. π acts as the secret key.
- Consider a brute-force attack where you try to guess the secret key. Is such an attack feasible?

Introduction: Secure communication

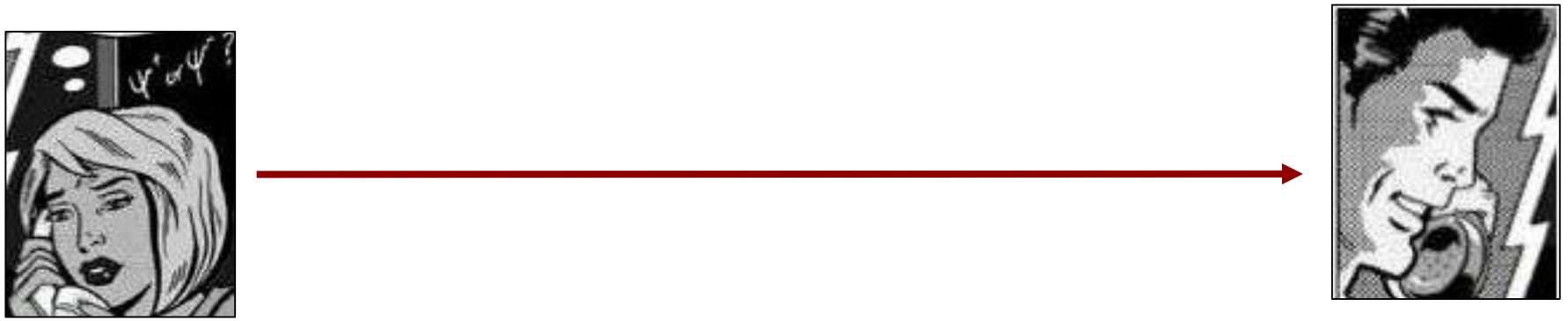
- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (Substitution Cipher): Let π be a permutation of the English letters. Substitute each letter α with the letter $\pi(\alpha)$.
- Can you break this scheme?

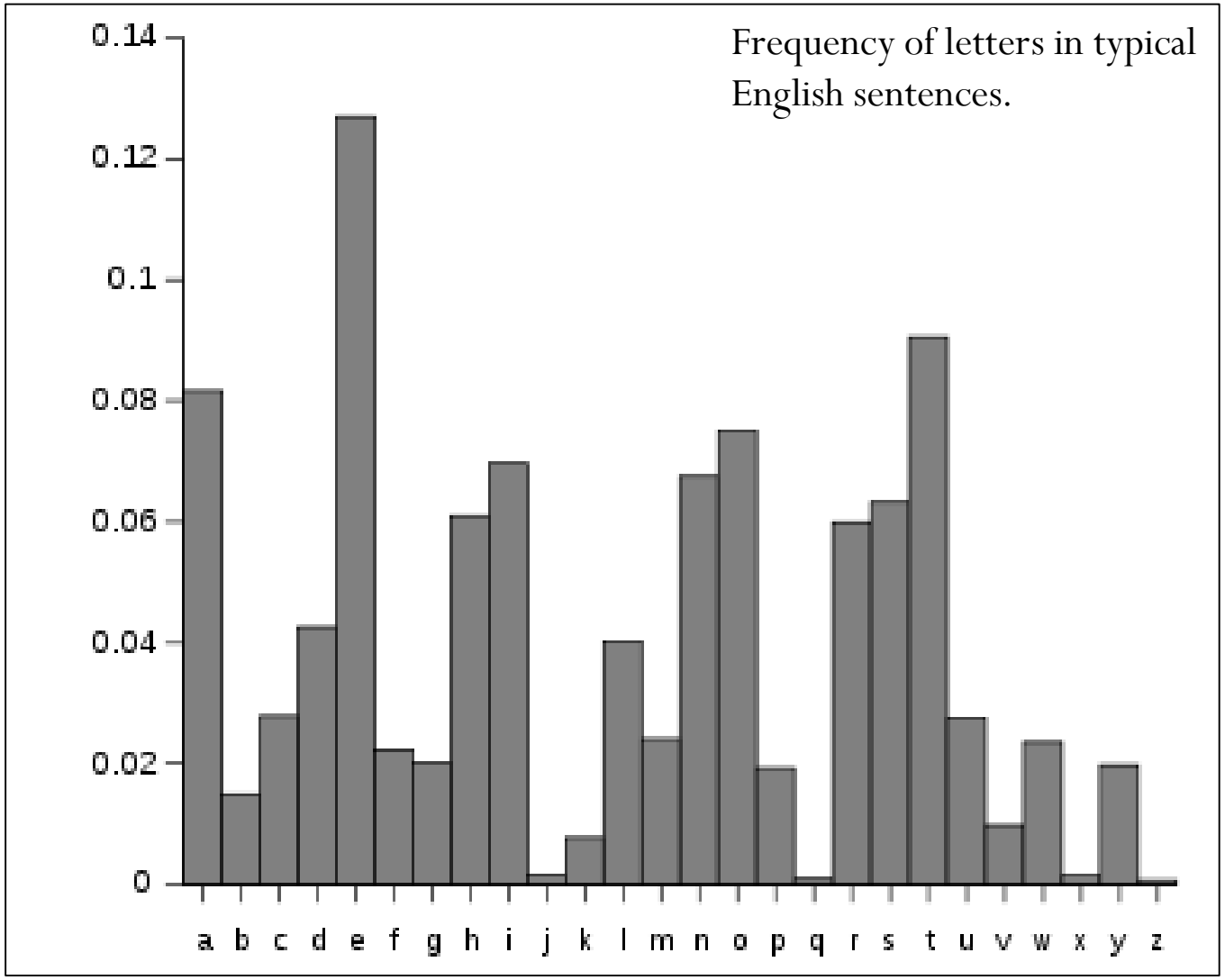
Introduction: Secure communication

- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



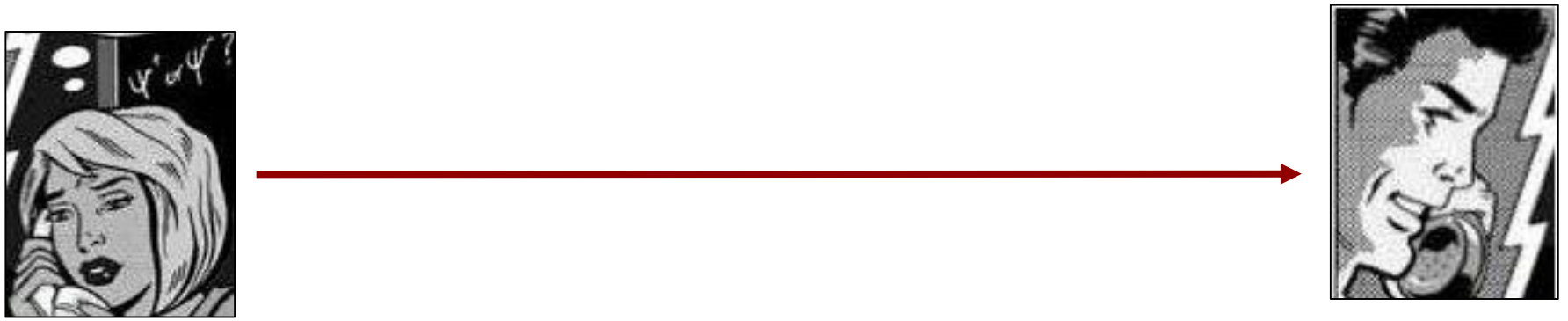
- Simple idea (Substitution Cipher): Let π be a permutation of the English letters. Substitute each letter α with the letter $\pi(\alpha)$.
- Attack idea: E's occur more frequently than X's

Introduction: Secure communication



Introduction: Secure communication

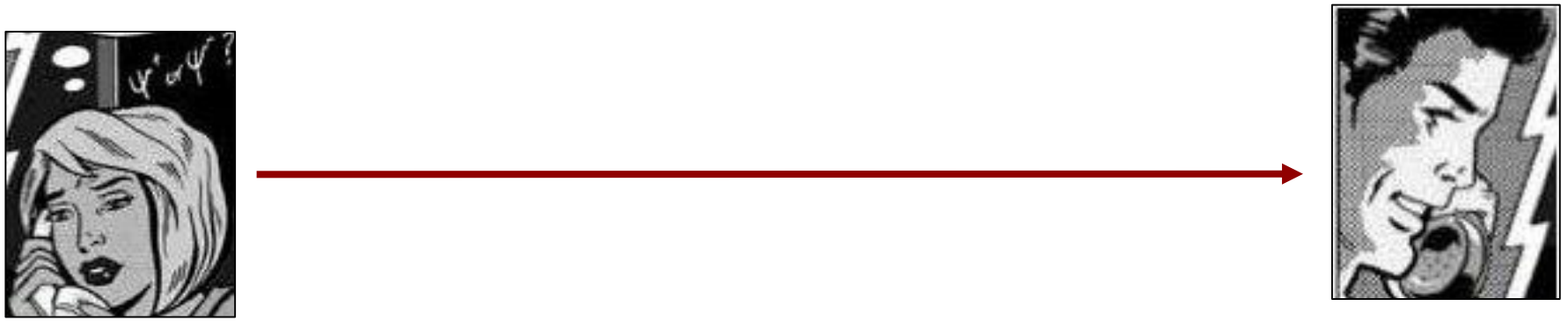
- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (Vignere Cipher): Let K be a short string. For any given message M, add repeated copies of K to M. K acts as the secret key.
- Example: Let $K = \text{AB}$ and $M = ATTACK$. Then the cipher text is $ATTACK + ABABAB = BVUCDM$.

Introduction: Secure communication

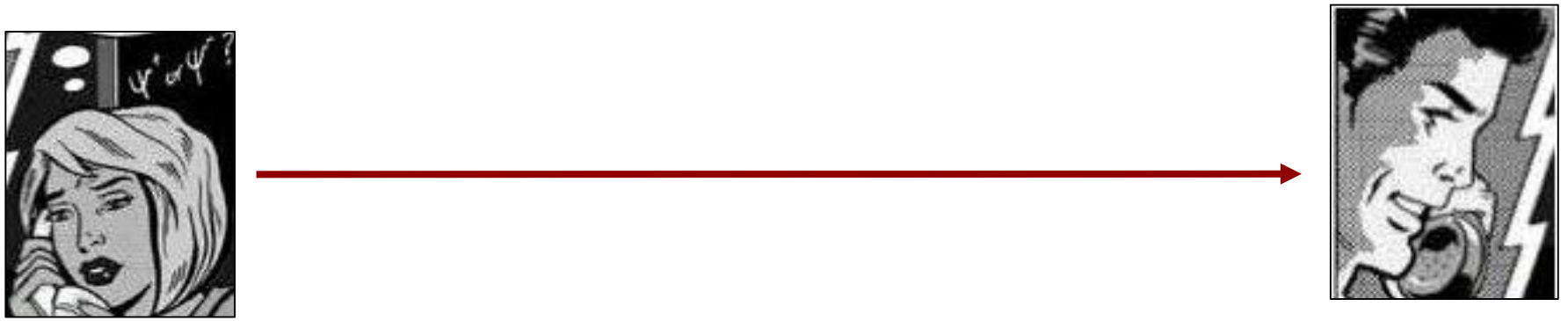
- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (Vignere Cipher): Let K be a short string. For any given message M, add repeated copies of K to M. K acts as the secret key.
- Can you break this scheme?

Introduction: Secure communication

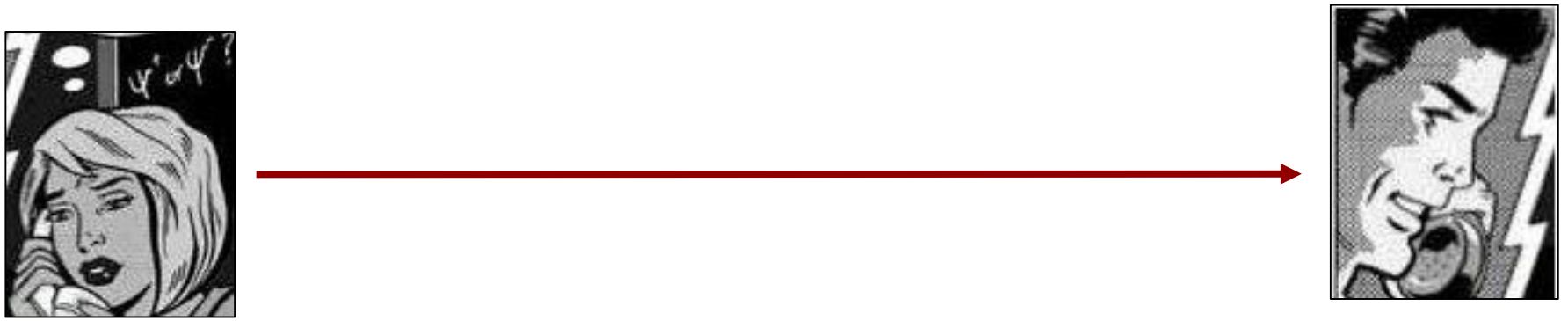
- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (One Time Pad(OTP)):Let the message M be an n binary string. Let K be an n bit binary string that is used as a secret key. Add M and K modulo 2 to get the ciphertext.
- Example: $M = 1101, K = 0101,$
  then $C = M + K\ (mod\ 2) = M \oplus K = 1000$

Introduction: Secure communication

- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.



- Simple idea (One Time Pad(OTP)):Let the message M be an n binary string. Let K be an n bit binary string that is used as a secret key. Add M and K modulo 2 to get the Ciphertext.
- Can you break this scheme?

Introduction: Secure communication

- Secure communication: Alice wants to talk to Bob without Eve (who has access to the channel) knowing the communication.
- Perfect Secrecy (Information Theoretic Security):
  - Let the message space be $\{0,1\}^n$.
  - For any two message M_0, M_1, and Ciphertext C
    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C]$$
    where the probability is over uniformly random K in the Keyspace.
  - Given the ciphertext, all messages are equally likely to be the secret message

Introduction: Secure communication

- Perfect Secrecy (Information Theoretic Security):
  - Let the message space be $\{0,1\}^n$.
  - For any two message M_0, M_1, and Ciphertext C
    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C]$$
    where the probability is over uniformly random K in the Keyspace.
- One Time Pad (OTP):
  - The Keyspace is $\{0,1\}^n$.
  - $E_K(M) = K \oplus M$
  - $D_K(C) = K \oplus C$
  - For any messages M_0, M_1 and ciphertext C:
    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C] = ??$$

Introduction: Secure communication

- Perfect Secrecy (Information Theoretic Security):
  - Let the message space be $\{0,1\}^n$.
  - For any two message M_0, M_1, and Ciphertext C
    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C]$$
    where the probability is over uniformly random K in the Keyspace.
- One Time Pad (OTP):
  - The Keyspace is $\{0,1\}^n$.
  - $E_K(M) = K \oplus M$
  - $D_K(C) = K \oplus C$
  - For any messages M_0, M_1 and ciphertext C:
    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C] = 1/2^n$$

Introduction: Secure communication

- Perfect Secrecy (Information Theoretic Security):
  - Let the message space be $\{0,1\}^n$.
  - For any two message M_0, M_1, and Ciphertext C
    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C]$$
    where the probability is over uniformly random K in the Keyspace.
- One Time Pad (OTP):
  - The Keyspace is $\{0,1\}^n$.
  - $E_K(M) = K \oplus M$
  - $D_K(C) = K \oplus C$
  - For any messages M_0, M_1 and ciphertext C:
    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C] = 1/2^n$$
  - Disadvantage: Key is as long as the message.

Introduction: Secure communication

- Perfect Secrecy (Information Theoretic Security):
  - Let the message space be $\{0,1\}^n$.
  - For any two message M_0, M_1, and Ciphertext C
    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C]$$
    where the probability is over uniformly random K in the Keyspace.
- One Time Pad (OTP):
  - The Keyspace is $\{0,1\}^n$.
  - $E_K(M) = K \oplus M$
  - $D_K(C) = K \oplus C$
  - For any messages M_0, M_1 and ciphertext C:
    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C] = 1/2^n$$
  - Disadvantage: Key is as long as the message.
- Fact: If $|M| > |K|$, then no scheme is perfectly secure.

Introduction: Secure communication

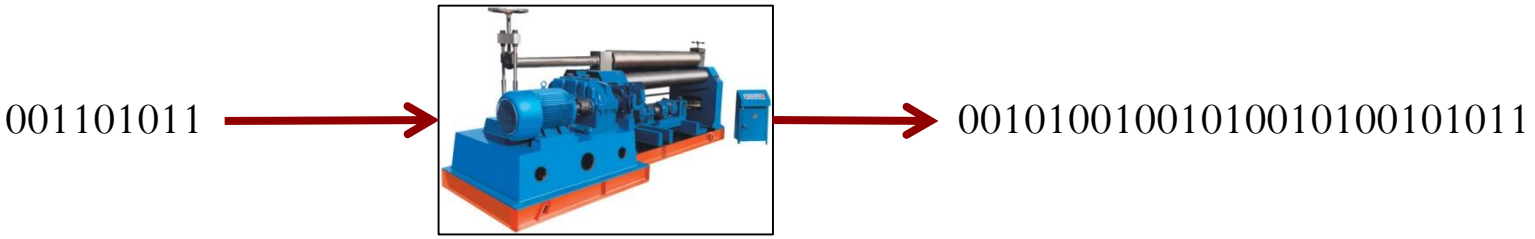
- Perfect Secrecy (Information Theoretic Security):
  - Let the message space be $\{0,1\}^n$.
  - For any two message M_0, M_1, and Ciphertext C
    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C]$$
    where the probability is over uniformly random K in the Keyspace.
- Fact: If $|M| > |K|$, then no scheme is perfectly secure.
- How do we get around this problem?

Introduction: Secure communication

- Perfect Secrecy (Information Theoretic Security):
  - Let the message space be $\{0,1\}^n$.
  - For any two message M_0, M_1, and Ciphertext C
    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C]$$
    where the probability is over uniformly random K in the Keyspace.
- Fact: If $|M| > |K|$, then no scheme is perfectly secure.
- How do we get around this problem?
  - Relax our notion of security: Instead of saying "it is impossible to break the scheme", we would like to say "it is *computationally infeasible* to break the scheme".

Introduction: Pseudorandom generator

- Suppose there was a *generator* that *stretches* random bits.

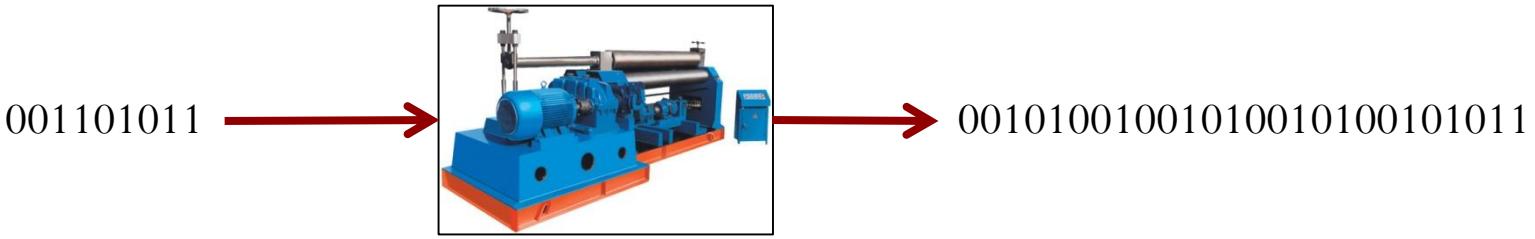


G

- Idea:
  - Choose a short key K randomly.
  - Obtain $K' = G(K)$.
  - Use K' as key for the one time pad.
- Issue: ?

Introduction: Pseudorandom generator

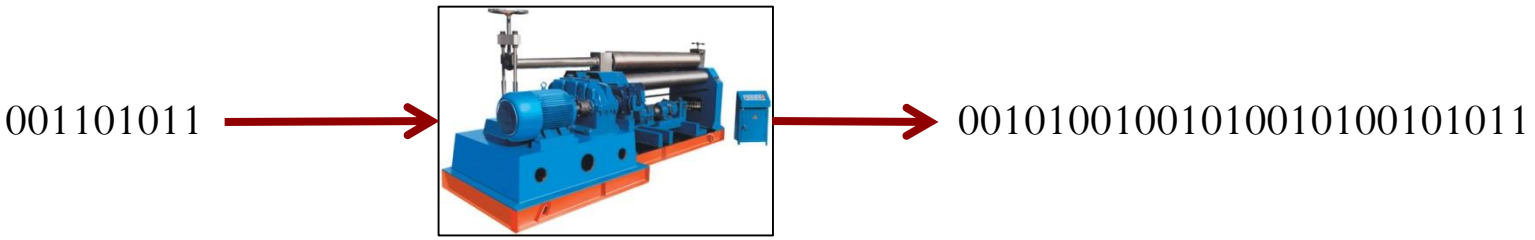
- Suppose there was a *generator* that *stretches* random bits.



  - Idea:
    - Choose a short key K randomly.
    - Obtain $K' = G(K)$.
    - Use K' as key for the one time pad.
  - Issue:
    - Such a generator is not possible!
    - Any such generator produces a longer string but the string is not *random*.

Introduction: Pseudorandom generator

- Suppose there was a *generator* that *stretches* random bits.



G

- Idea:
  - Choose a short key K randomly.
  - Obtain $K' = G(K)$.
  - Use K' as key for the one time pad.
- Issue:
  - Such a generator is not possible!
  - Any such generator produces a longer string but the string is not *random*.
- What if we can argue that the output of the generator is *computationally indistinguishable* from truly random string.

End