COMP-330A
Probabilistic Computations and Cryptography

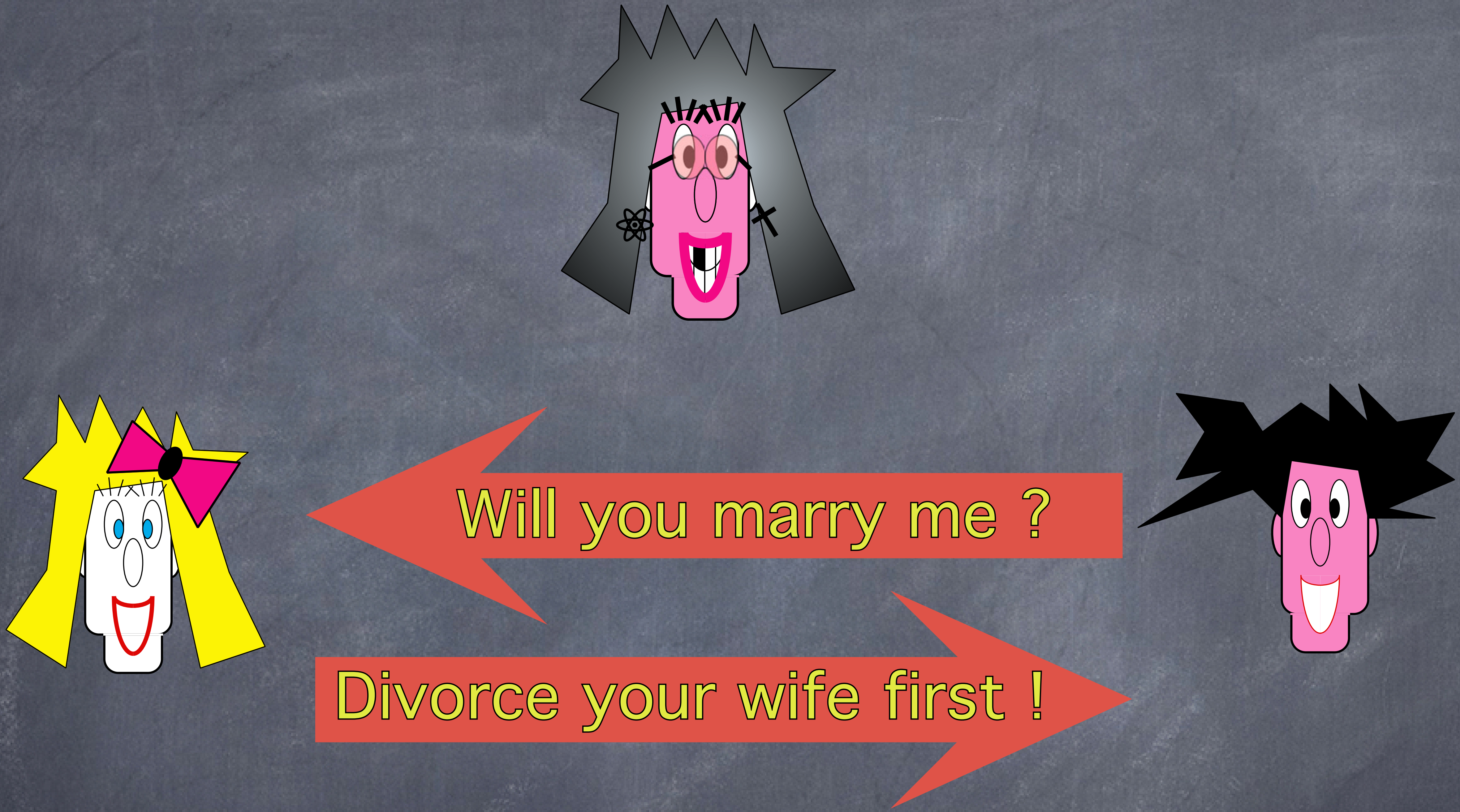
Prof. Claude Crépeau

School of Computer Science
McGill University

Classical

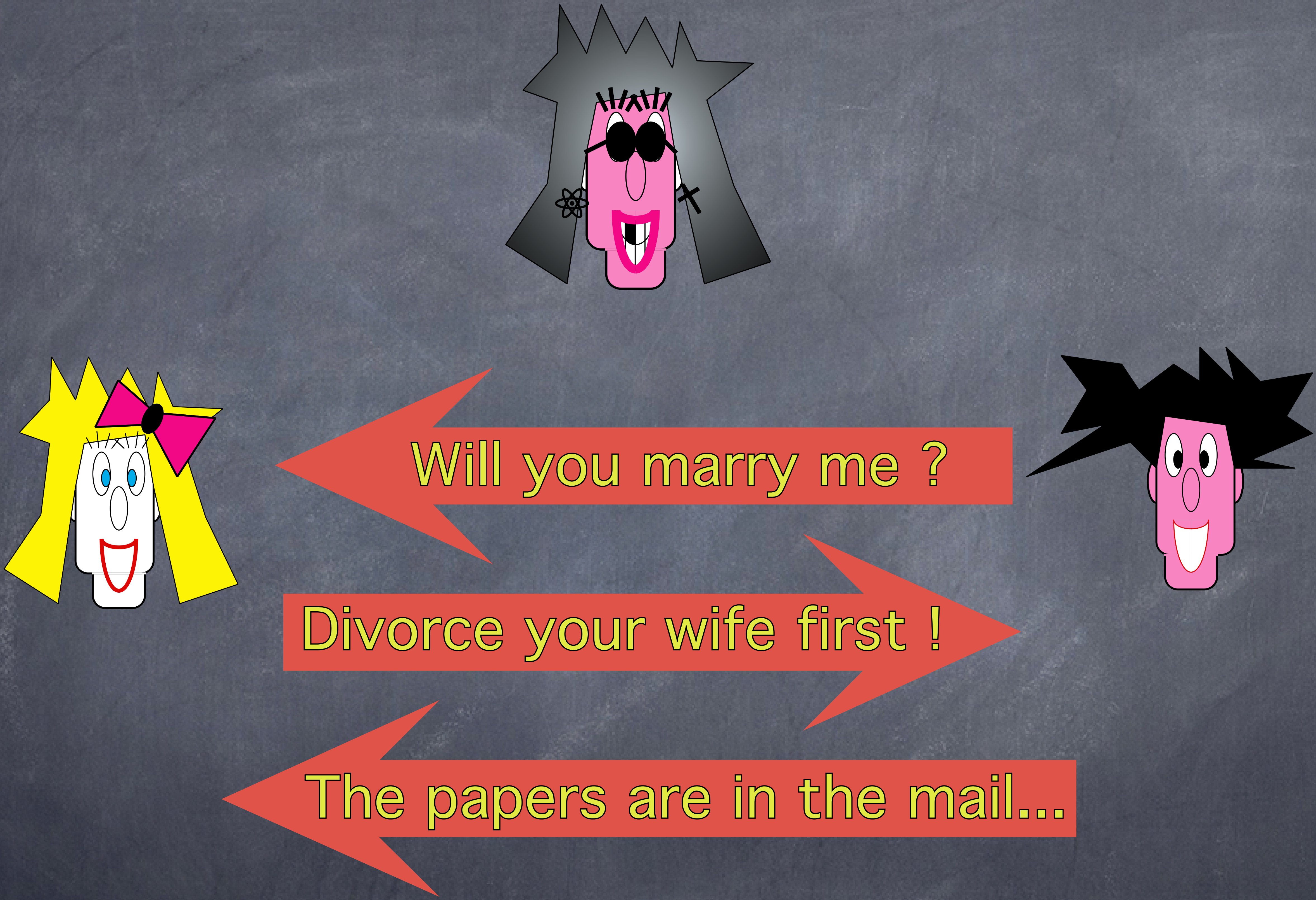
Cryptography

Will you marry me ?



Will you marry me ?

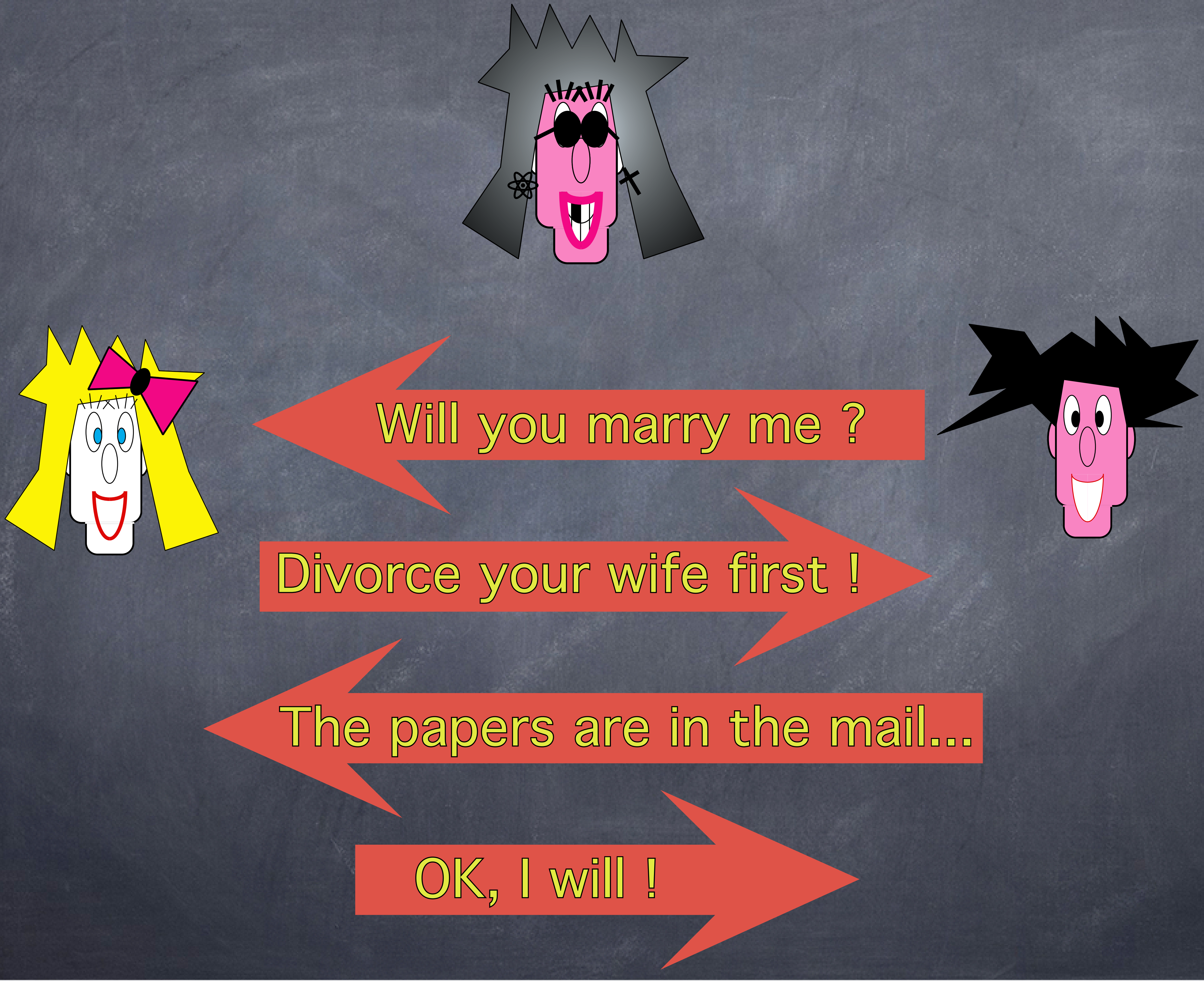
Divorce your wife first !



Will you marry me ?

Divorce your wife first !

The papers are in the mail...



Will you marry me ?

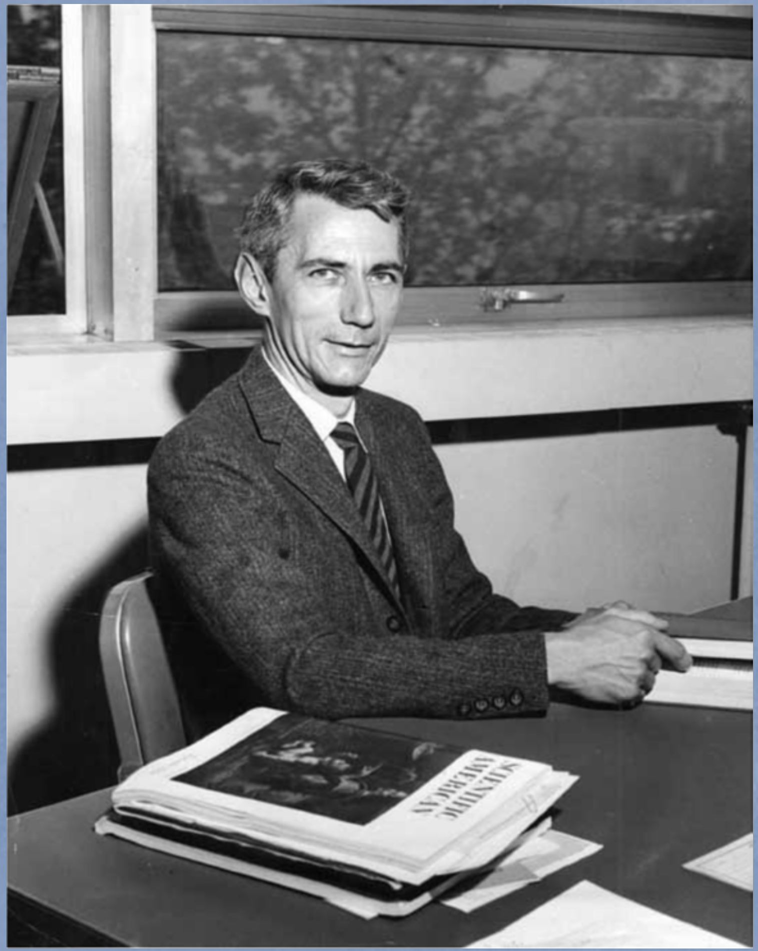
Divorce your wife first !

The papers are in the mail...

OK, I will !

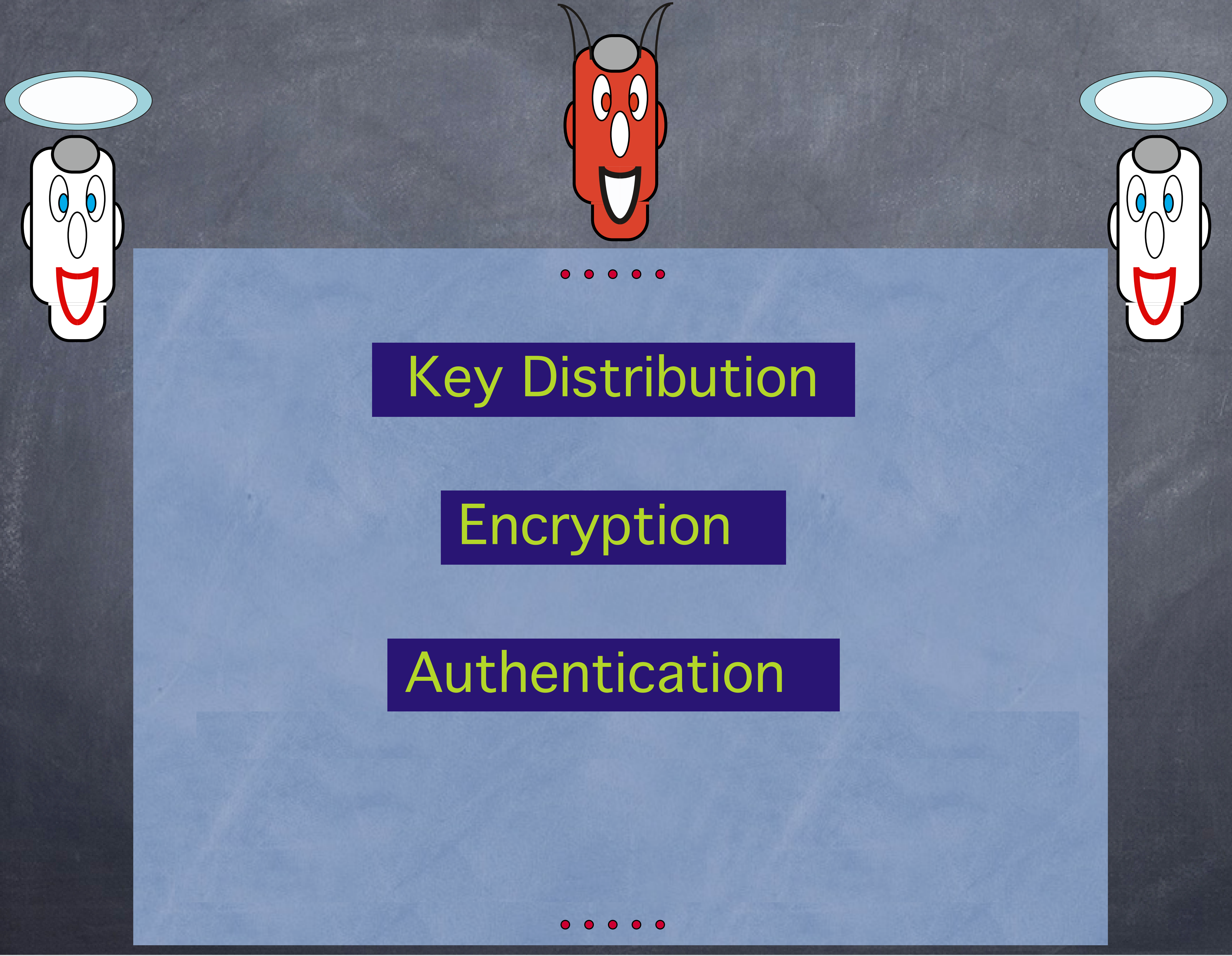
Information

Theoretical



Cryptography

Information Theoretical Cryptography

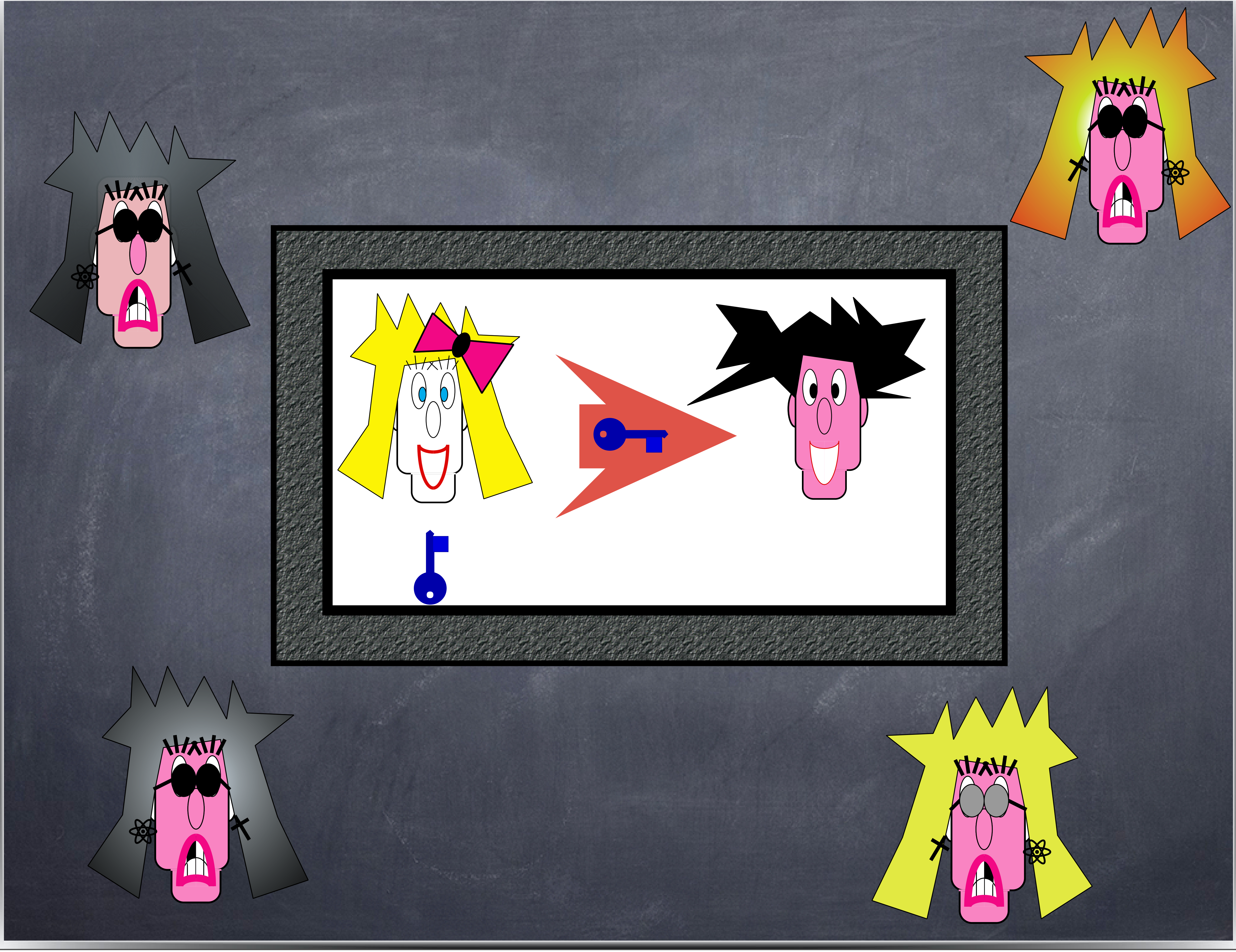


Key Distribution

Encryption

Authentication

Key Distribution



Encryption

8RdewtU5qkLa$es!T9@

Decryption

Will you marry m es!T9@

Encryption

8RdewtU5qkLa$ y me ?

8RdewtU5qkLa$es!T9@

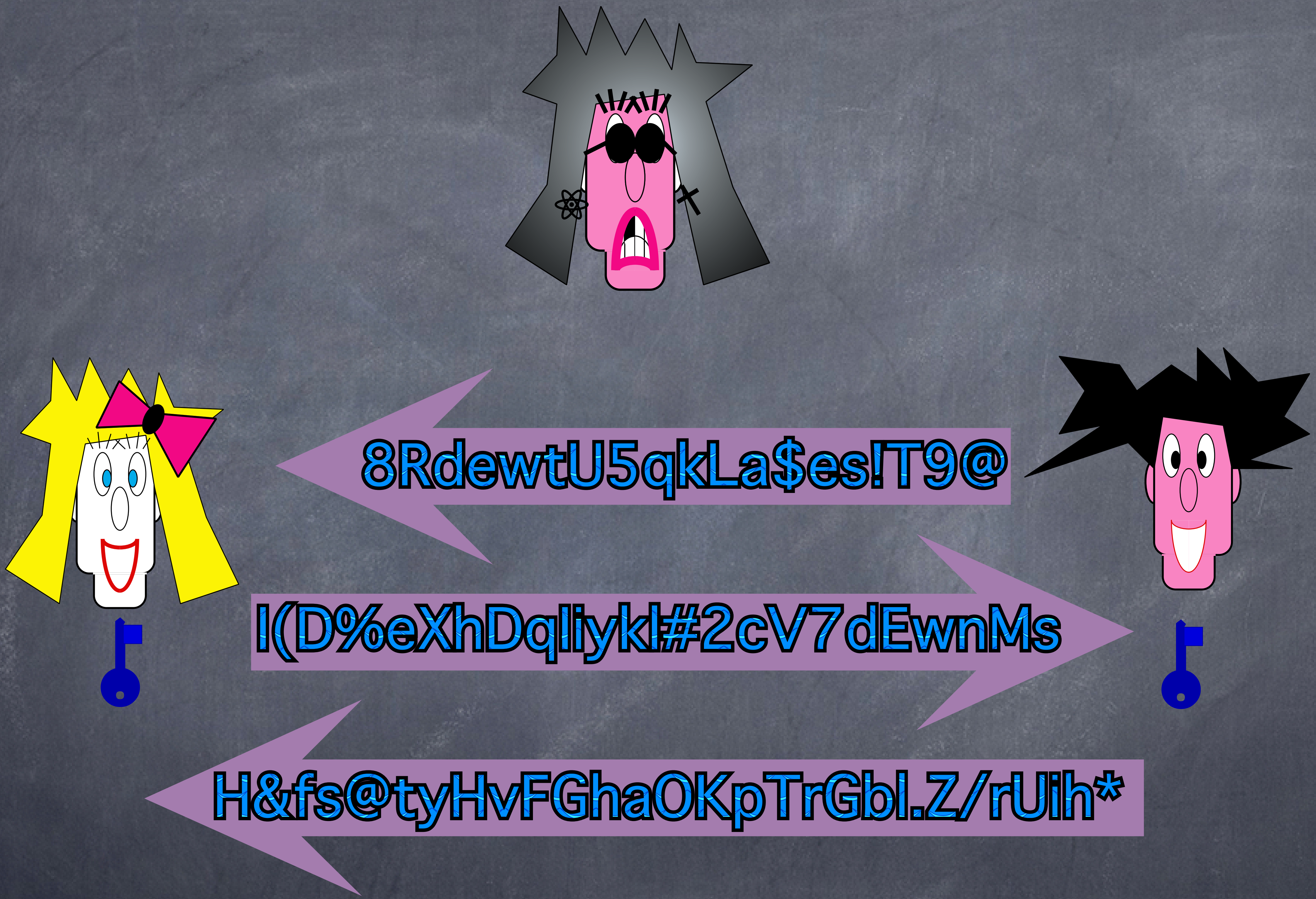
I(D%eXhDqliykl#2cV7dEwnMs

Encryption

Decryption

Divorce your wif 2cV7dEwnMs

I(D%eXhDqliykl#2cV7 ur wife first !



8RdewtU5qkLa$es!T9@

I(D%eXhDqlliykl#2cV7dEwnMs

H&fs@tyHvFGhaOKpTrGbl.Z/rUih*

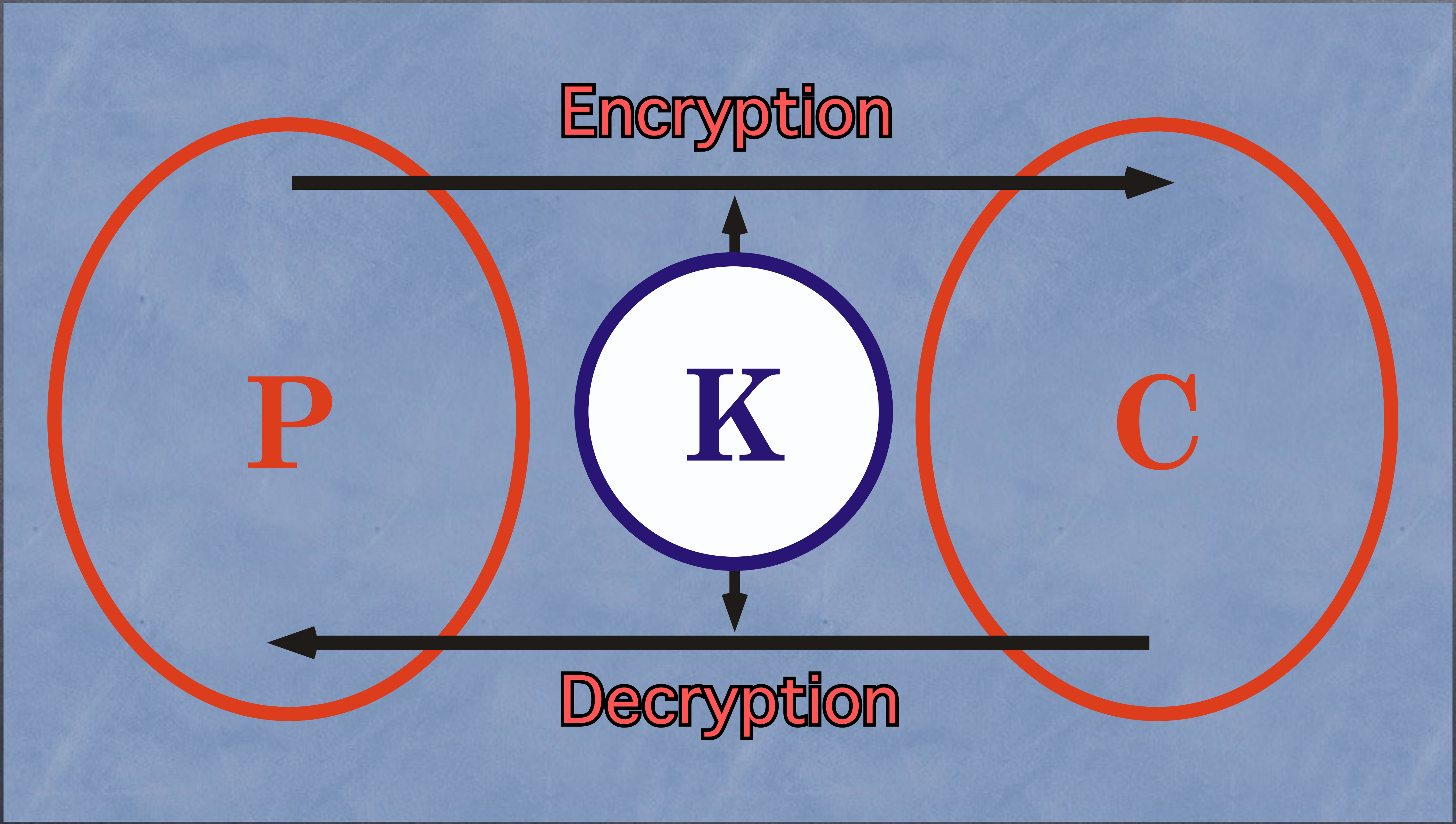
8RdewtU5qkLa$es!T9@

l(D%eXhDqliykl#2cV7dEwnMs

H&fs@tyHvFGhaOKpTrGbl.Z/rUih*

B7B3tdsjUila

Symmetric Encryption



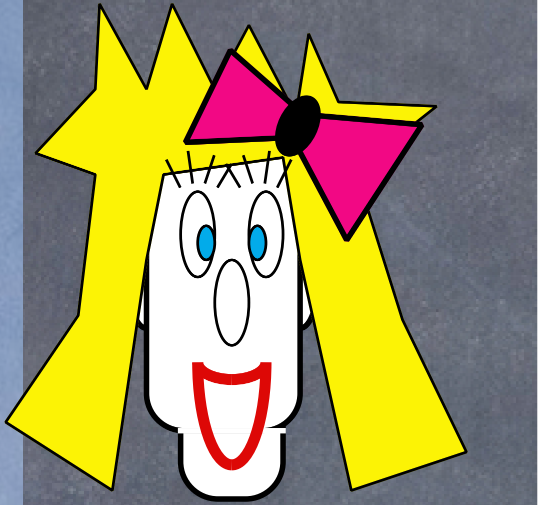
Information Theoretical Security

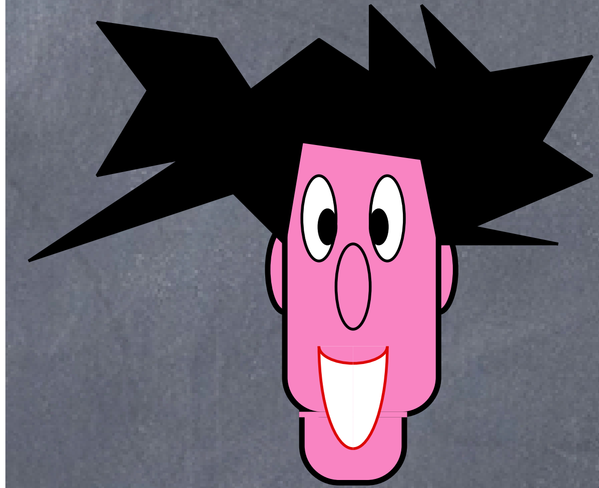
Symmetric Encryption

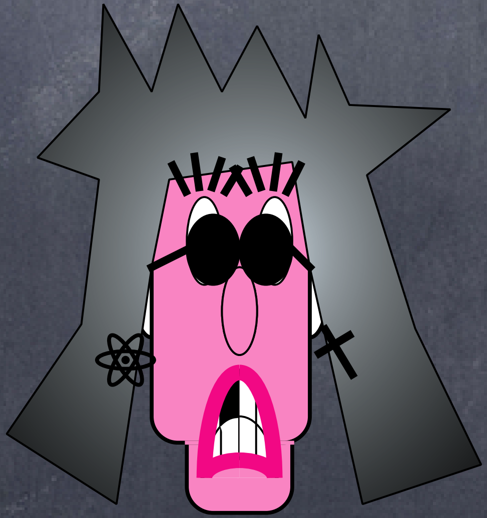
Ceasar’s Cipher

VERNAM's Cipher

m
1
0
1
0
0
1
0
0
1
1
1
1
1
0
0
1



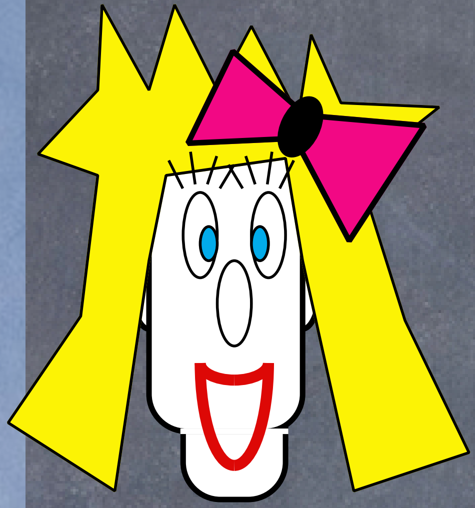


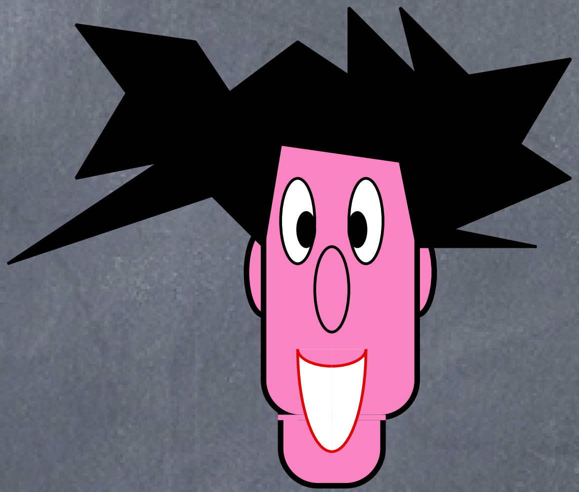


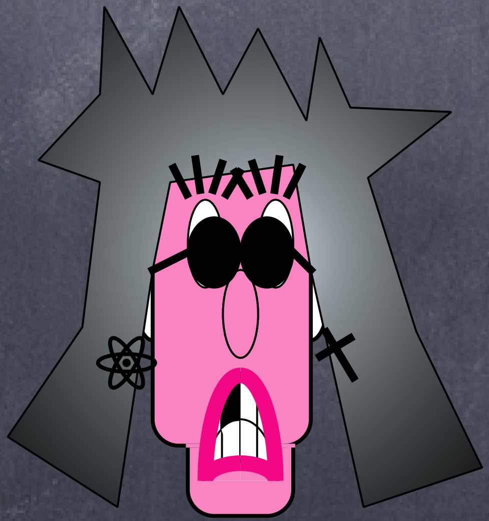
VERNAM's Cipher

m⊕k	
1	1
0	1
1	1
0	0
0	0
1	1
0	1
0	0
1	1
1	1
1	0
1	1
1	0
0	1
0	1
1	1

⊕



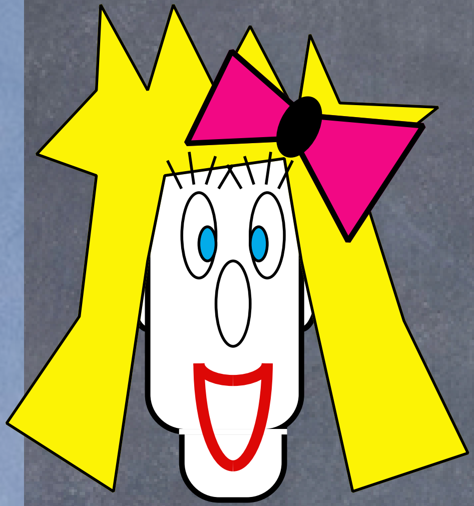


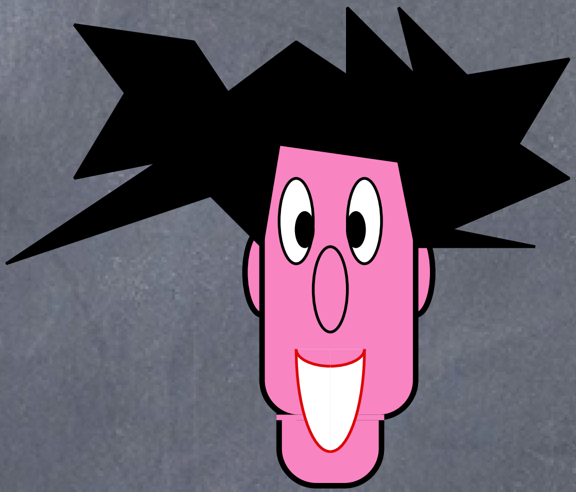


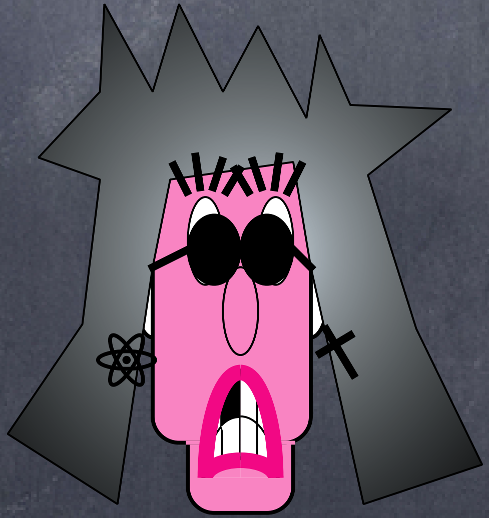
VERNAM’s Cipher

m⊕k= c		
1	1	0
0	1	1
1	1	0
0	0	0
0	0	0
1	1	0
0	1	1
0	0	0
1	1	0
1	1	0
1	0	1
1	1	0
1	0	1
0	1	1
0	1	1
1	1	0

⊕ =



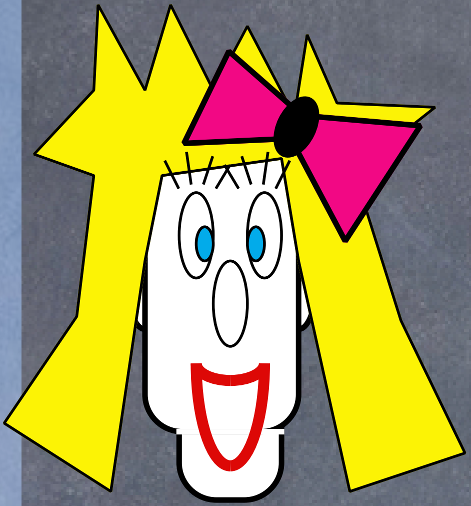


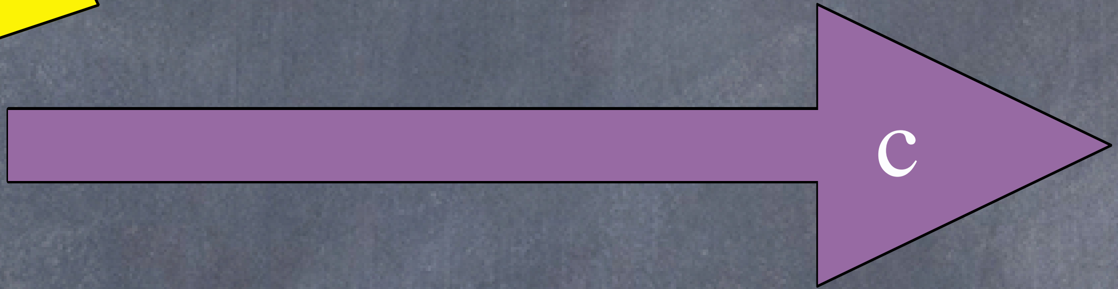


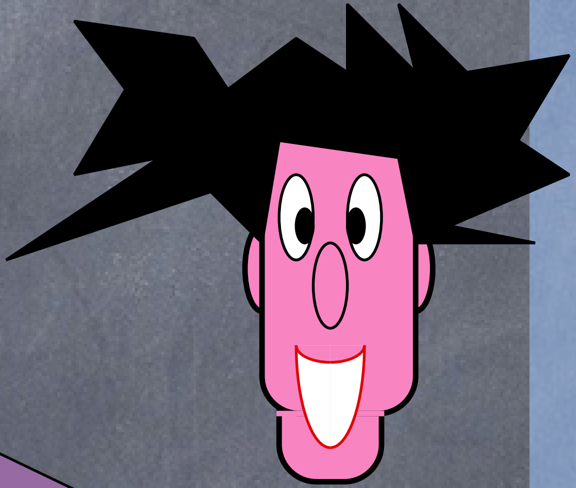
VERNAM’s Cipher

m⊕k=		c
1	1	0
0	1	1
1	1	0
0	0	0
0	0	0
1	1	0
0	1	1
0	0	0
1	1	0
1	1	0
1	0	1
1	1	0
1	0	1
0	1	1
0	1	1
1	1	0

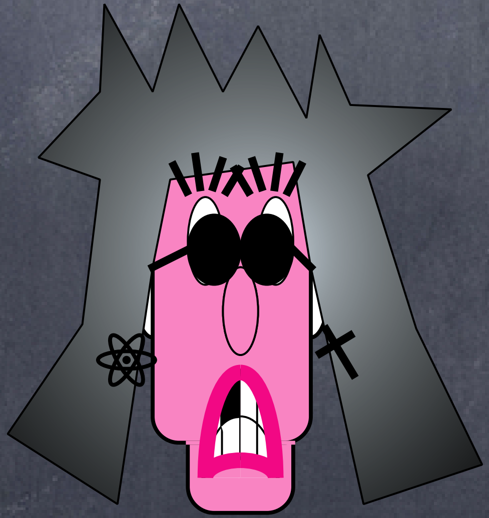
$\oplus =$





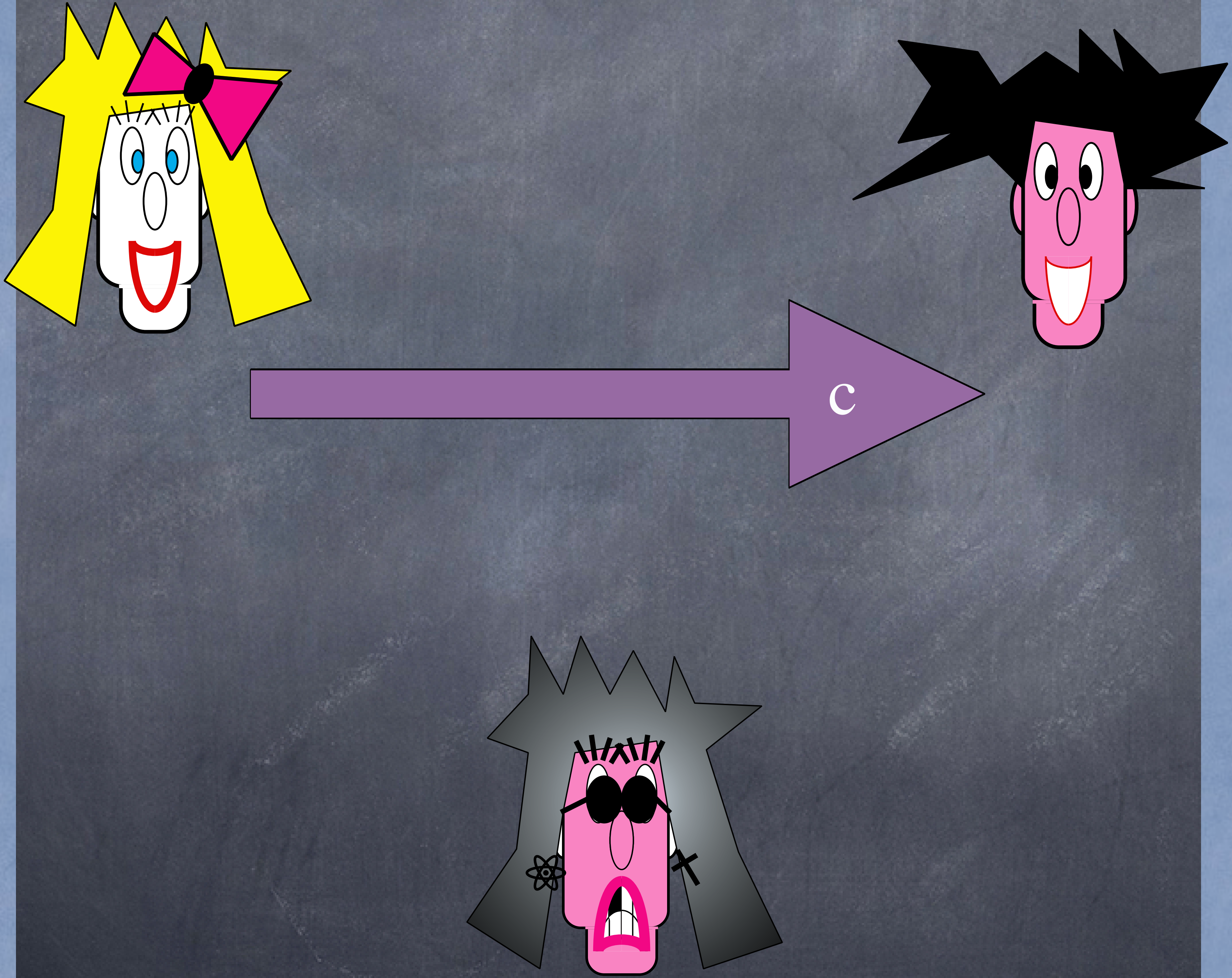


c
0
1
0
0
0
0
1
0
0
0
1
0
1
1
1
0



VERNAM's Cipher

$m \oplus$	$k =$	c
1	1	0
0	1	1
1	1	0
0	0	0
0	0	0
1	1	0
0	1	1
0	$\oplus$ 0	$=$ 0
1	1	0
1	1	0
1	0	1
1	1	0
1	0	1
0	1	1
0	1	1
1	1	0

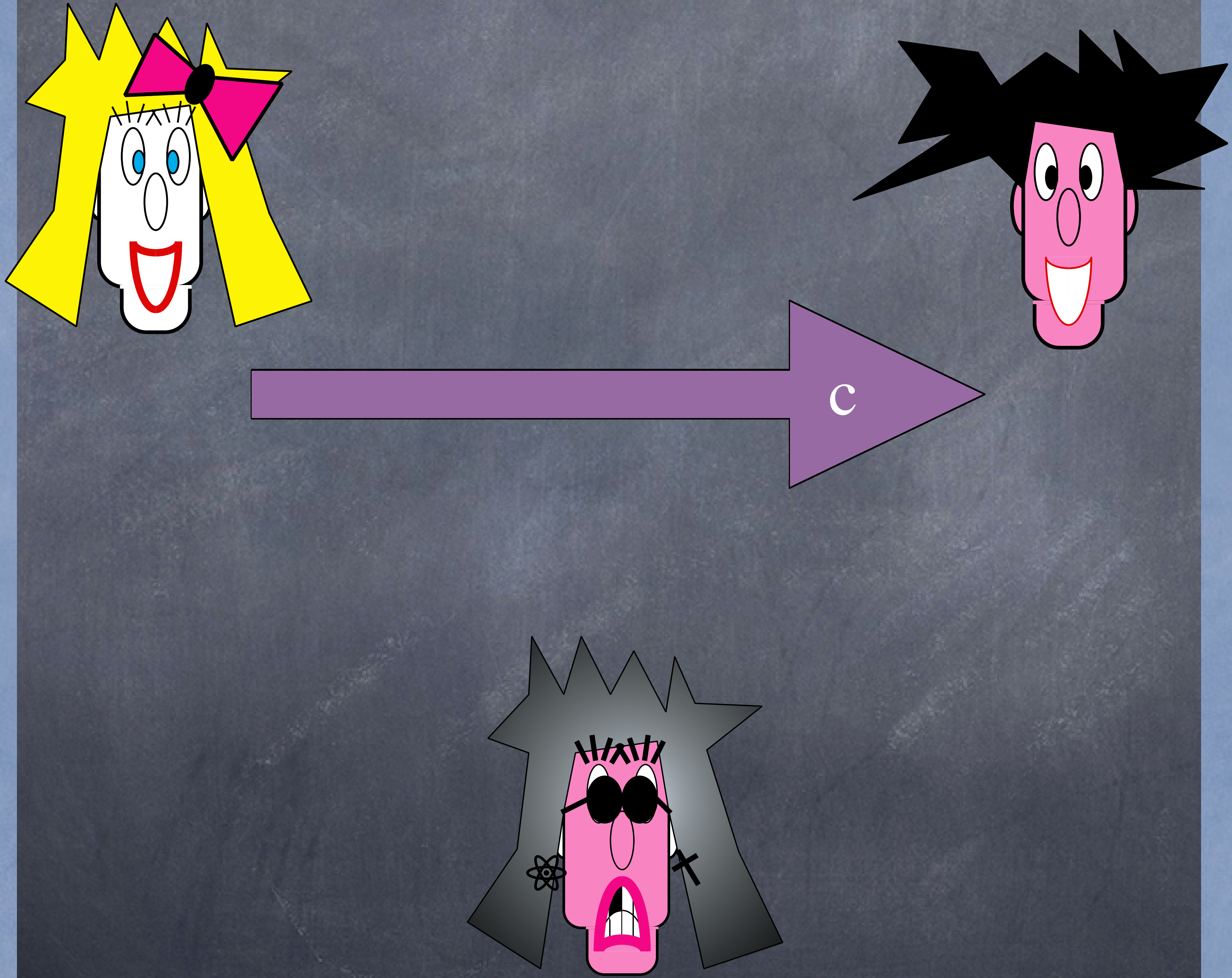


$c \oplus$	k
0	1
1	1
0	1
0	0
0	0
0	1
1	1
0	$\oplus$ 0
0	1
0	1
1	0
0	1
1	0
1	1
1	1
0	1

VERNAM's Cipher

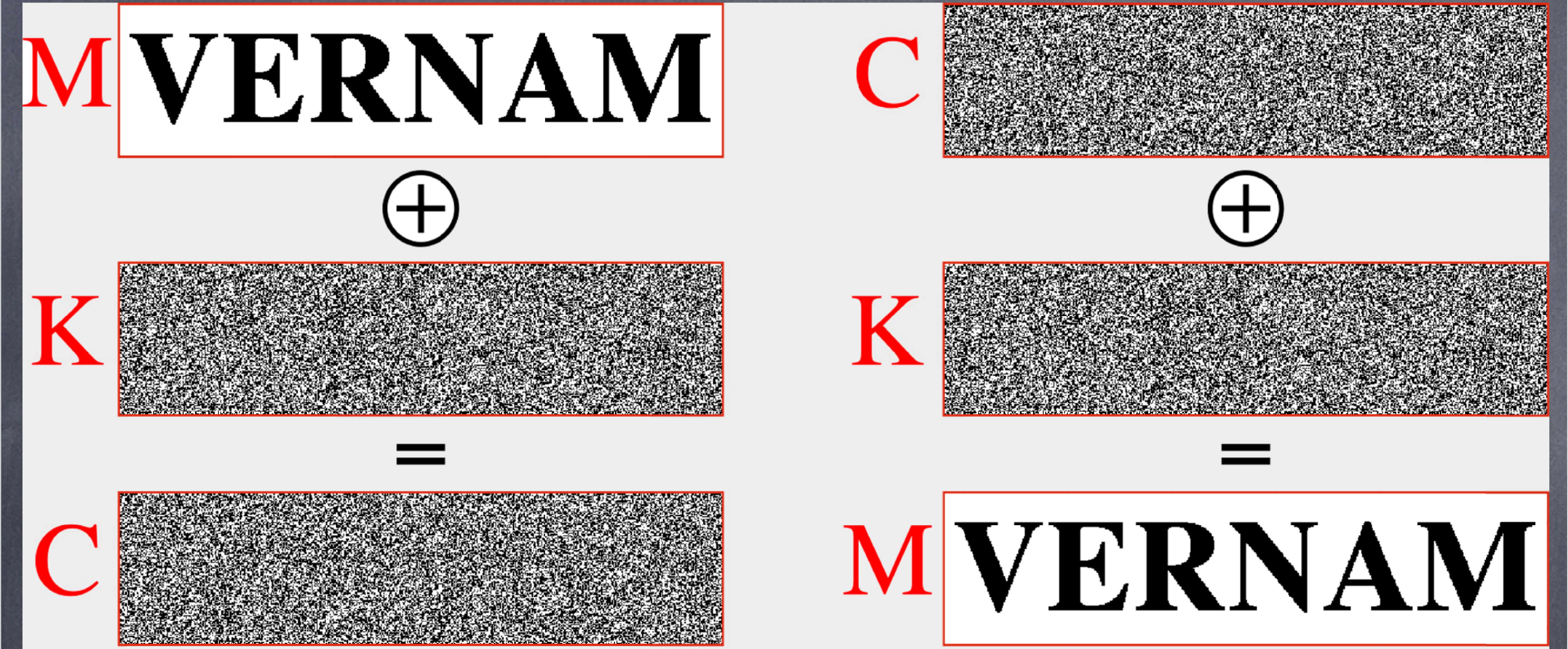
m ⊕	k =	c
1	1	0
0	1	1
1	1	0
0	0	0
0	0	0
1	1	0
0	1	1
0	0	0
1	1	0
1	1	0
1	0	1
1	1	0
1	0	1
0	1	1
0	1	1
1	1	0

⊕ =



c ⊕	k=	m
0	1	1
1	1	0
0	1	1
0	0	0
0	0	0
0	1	1
1	1	0
0	0	0
0	1	1
0	1	1
1	0	1
0	1	1
1	0	1
1	1	0
1	1	0
0	1	1

⊕ =



M VERNAM

⊕

K

=

C

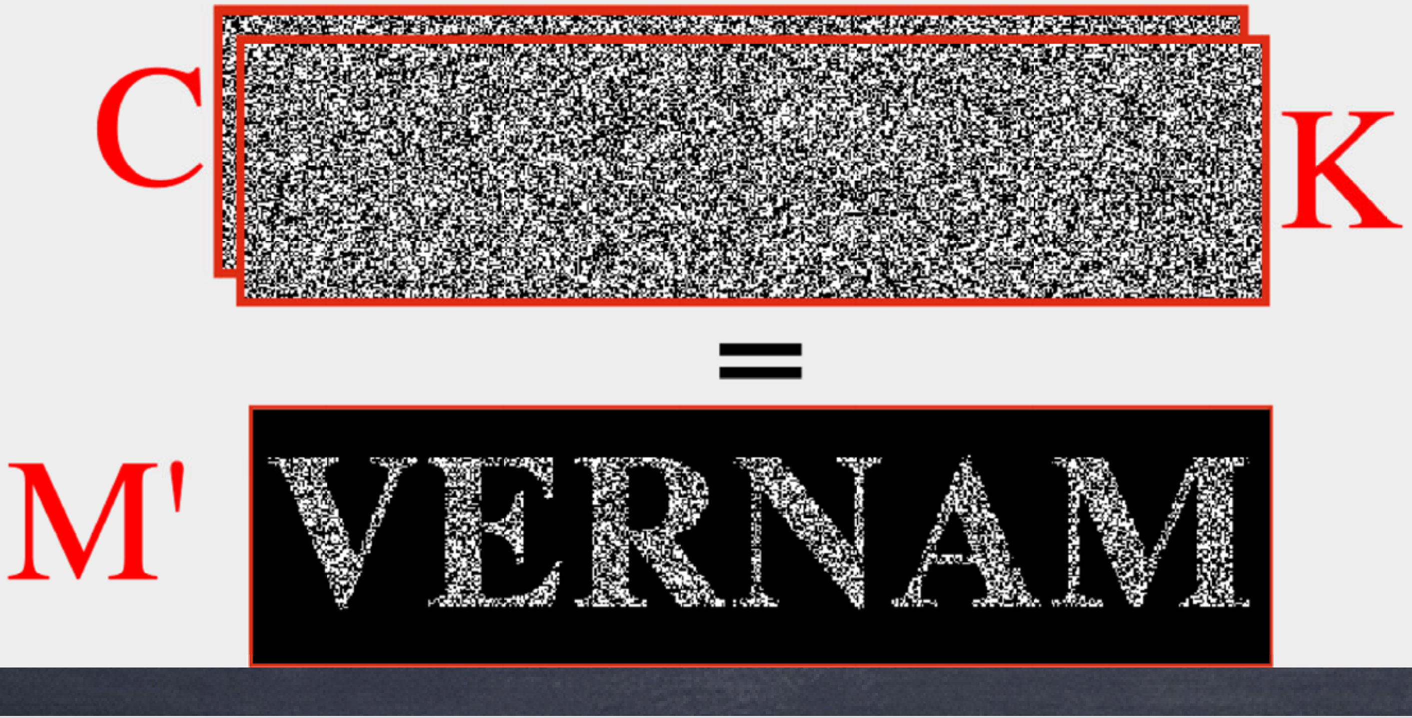
C

⊕

K

=

M VERNAM

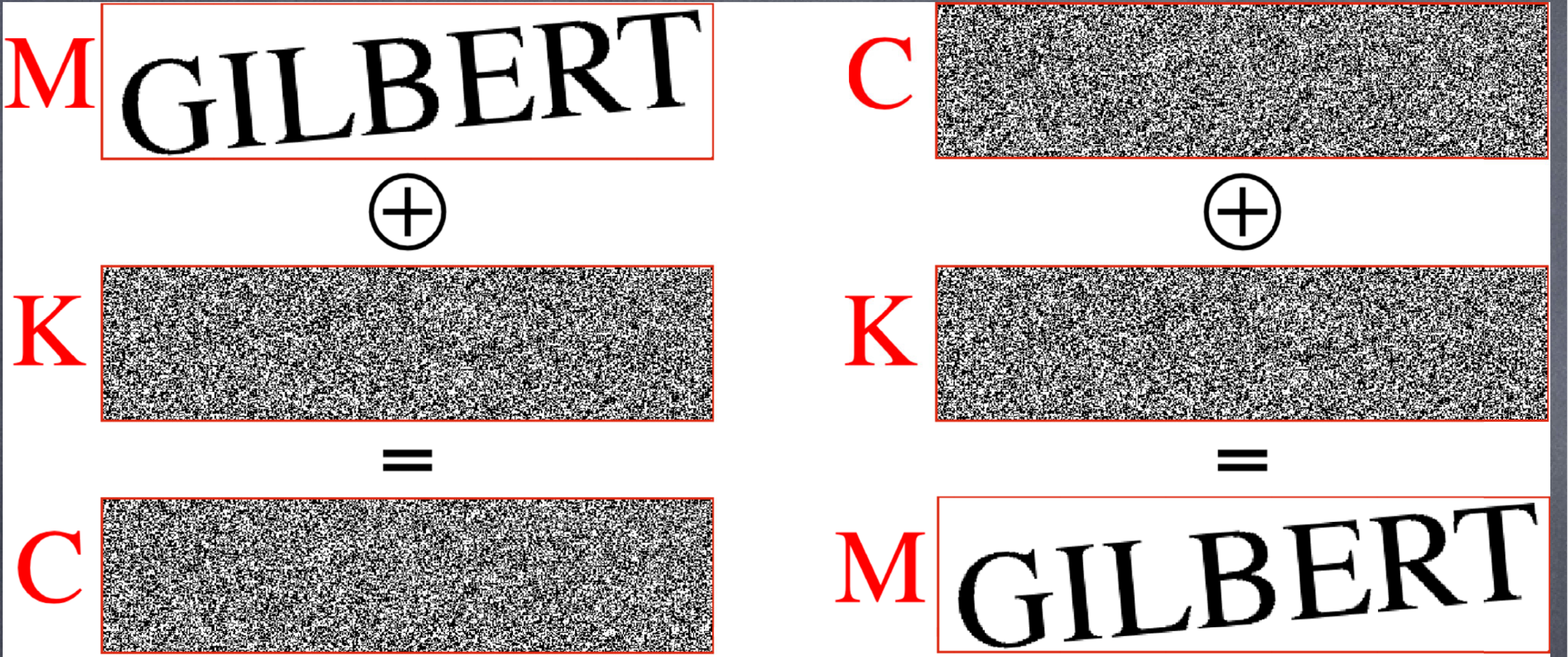


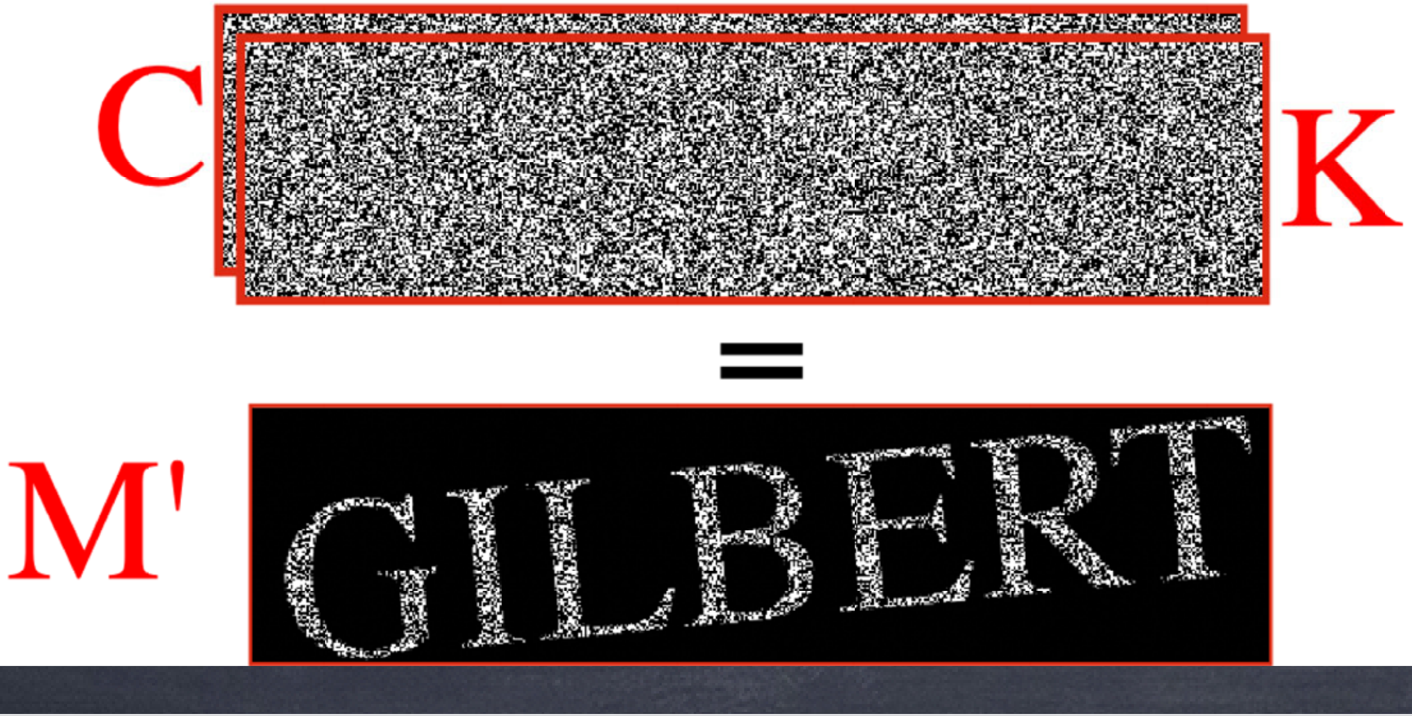
C

K

=

M' VERNAM

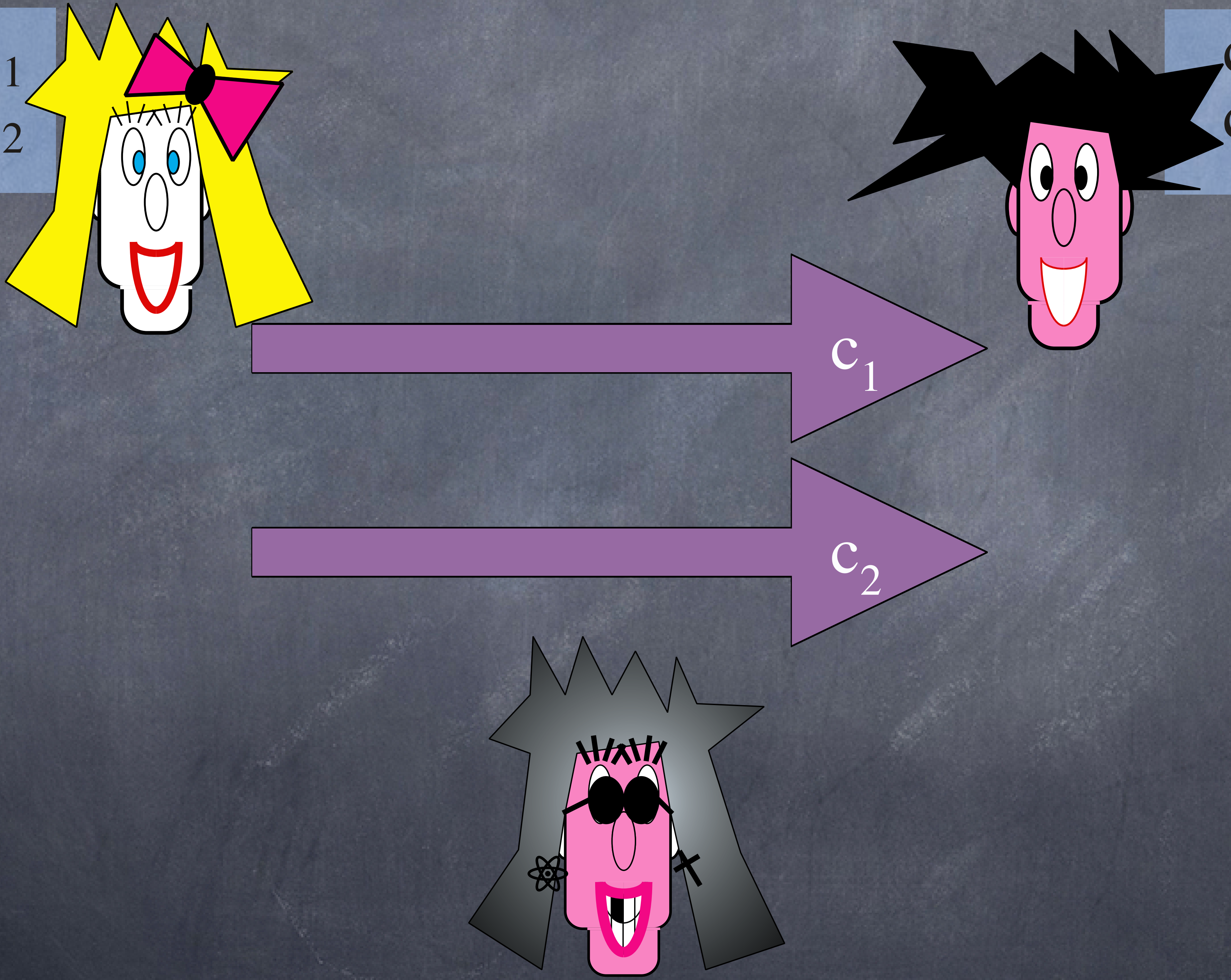

M
GILBERT
⊕
K
=
C
C
⊕
K
=
M
GILBERT


C
K
=
M'
GILBERT

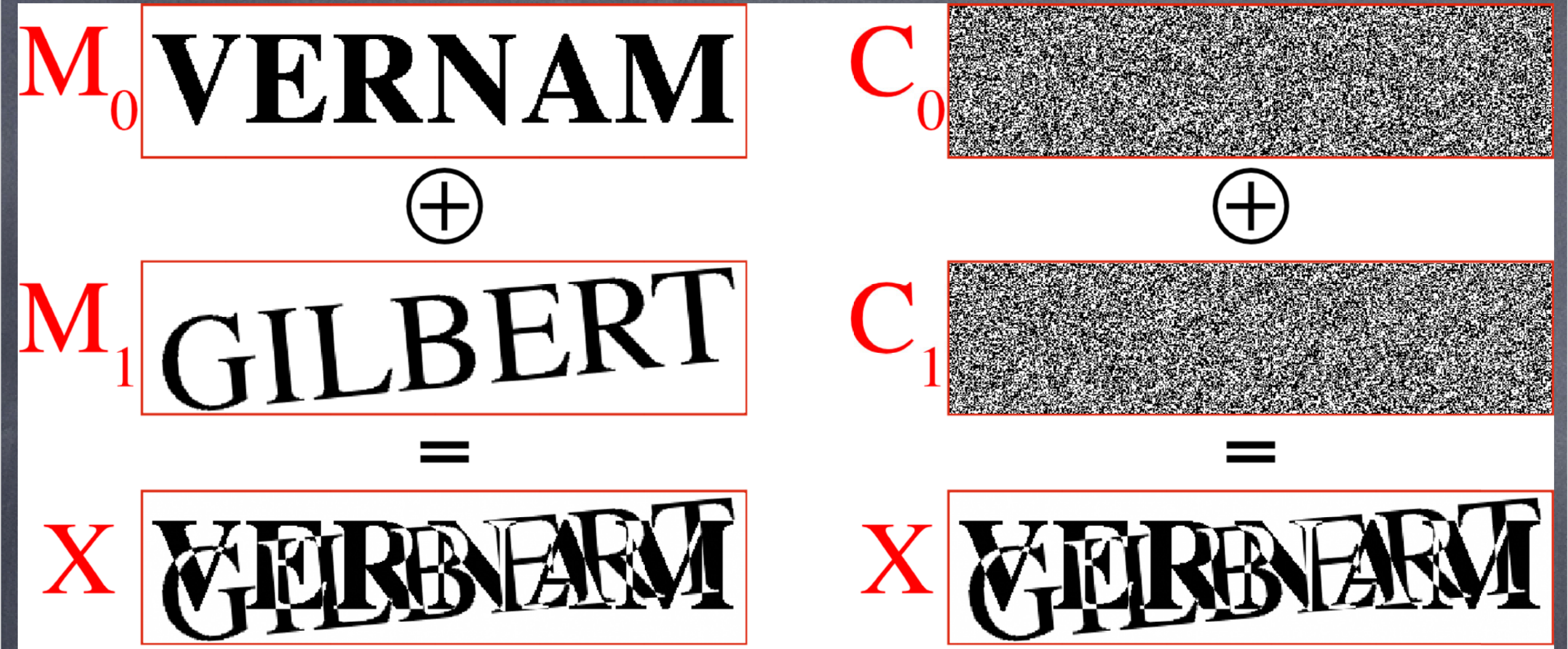
VERNAM's One-Time Pad

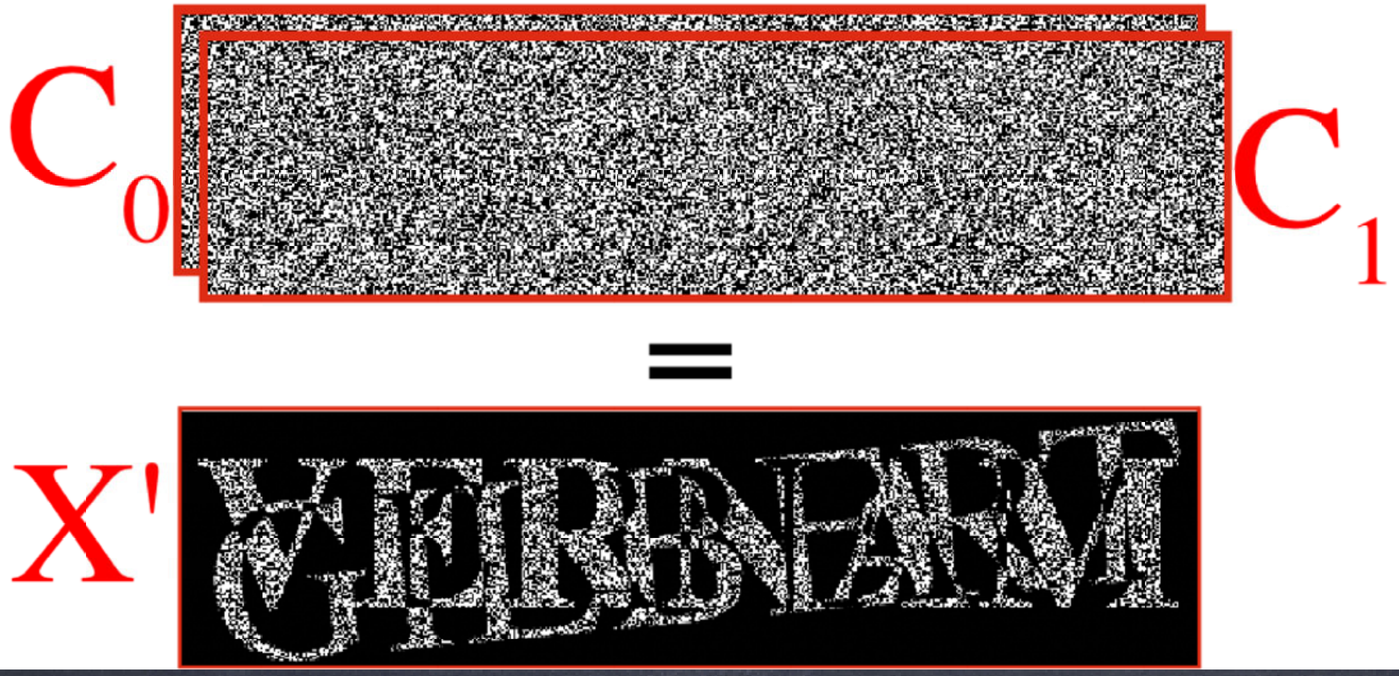
$m_1 \oplus k = c_1$
$m_2 \oplus k = c_2$

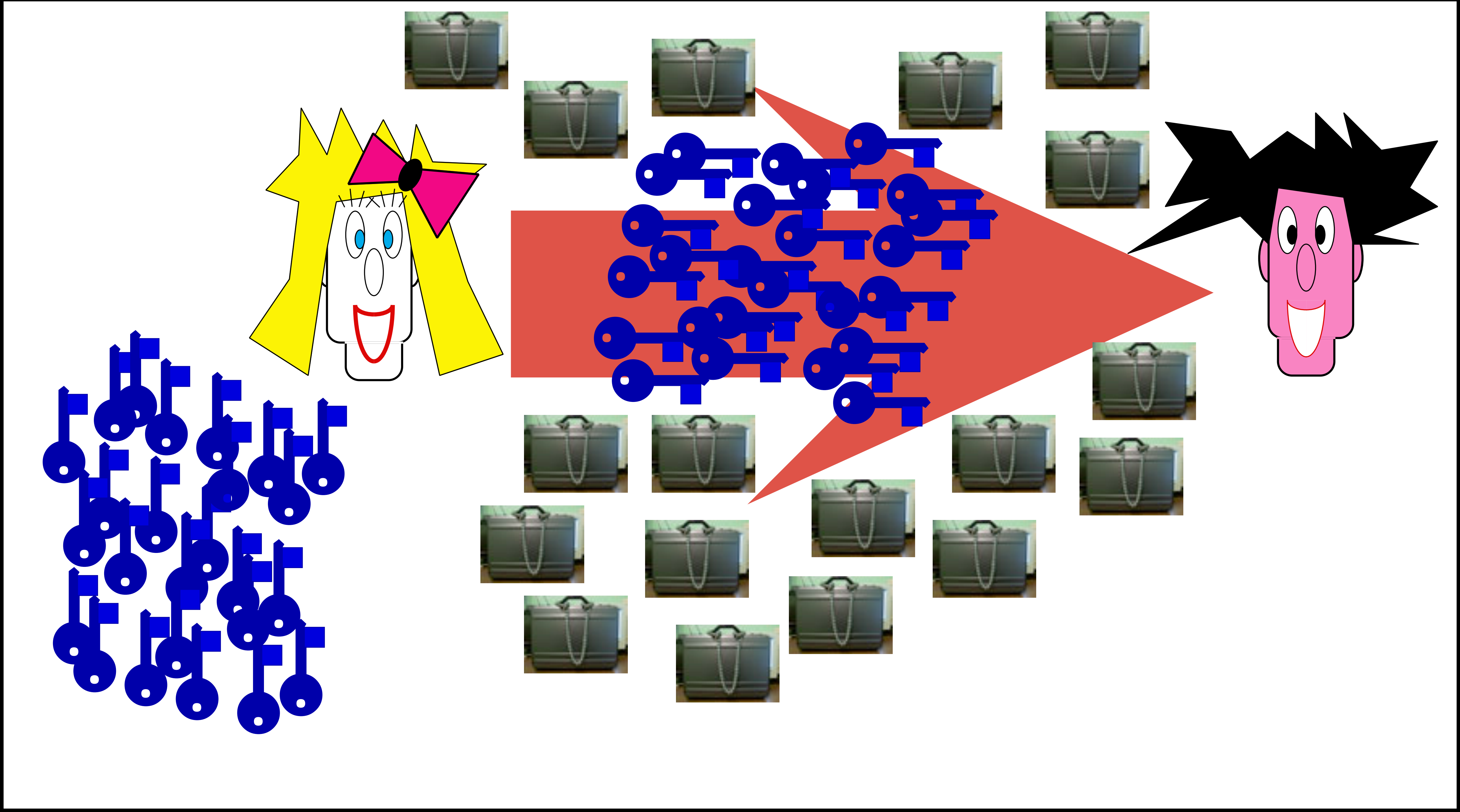
$c_1 \oplus k = m_1$
$c_2 \oplus k = m_2$



$c_1 \oplus c_2 = m_1 \oplus m_2$

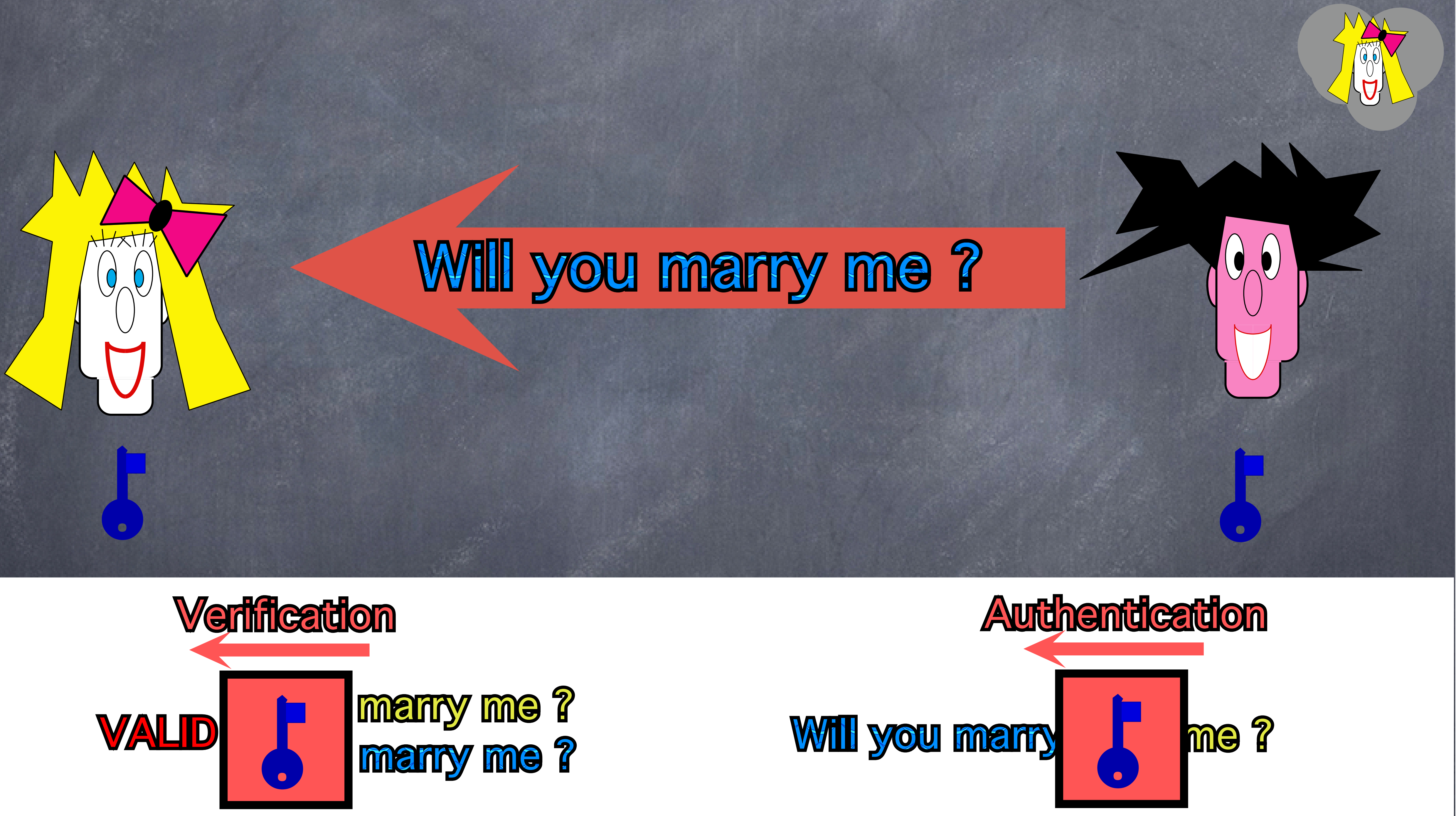

M_0
VERNAM
⊕
M_1
GILBERT
=
X
C_0
⊕
C_1
=
X


C_0
C_1
=
X'



Authentication

Authentication



Will you marry me ?

Verification

VALID

marry me ?

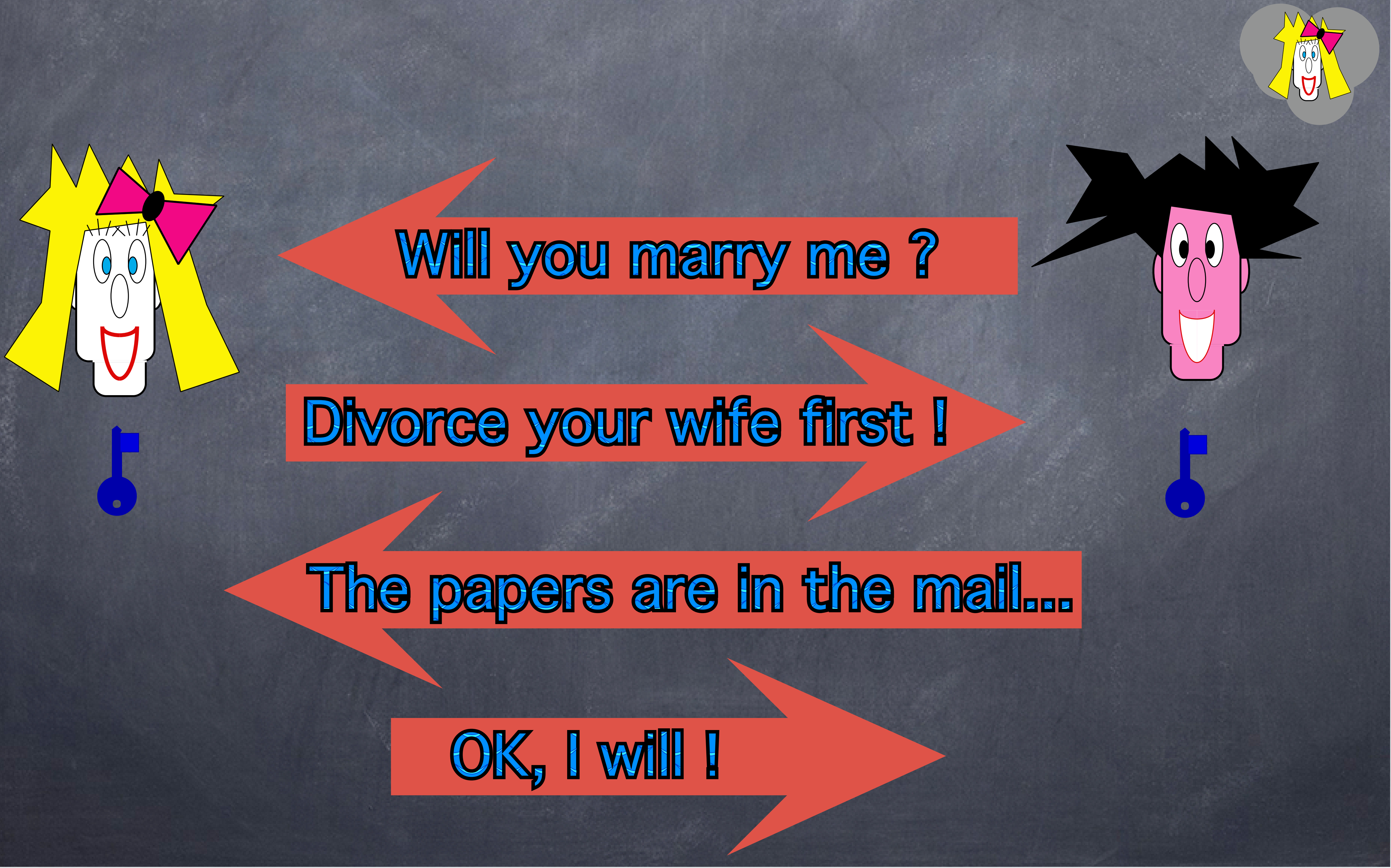
marry me ?

Authentication

Will you marry

me ?

Authentication



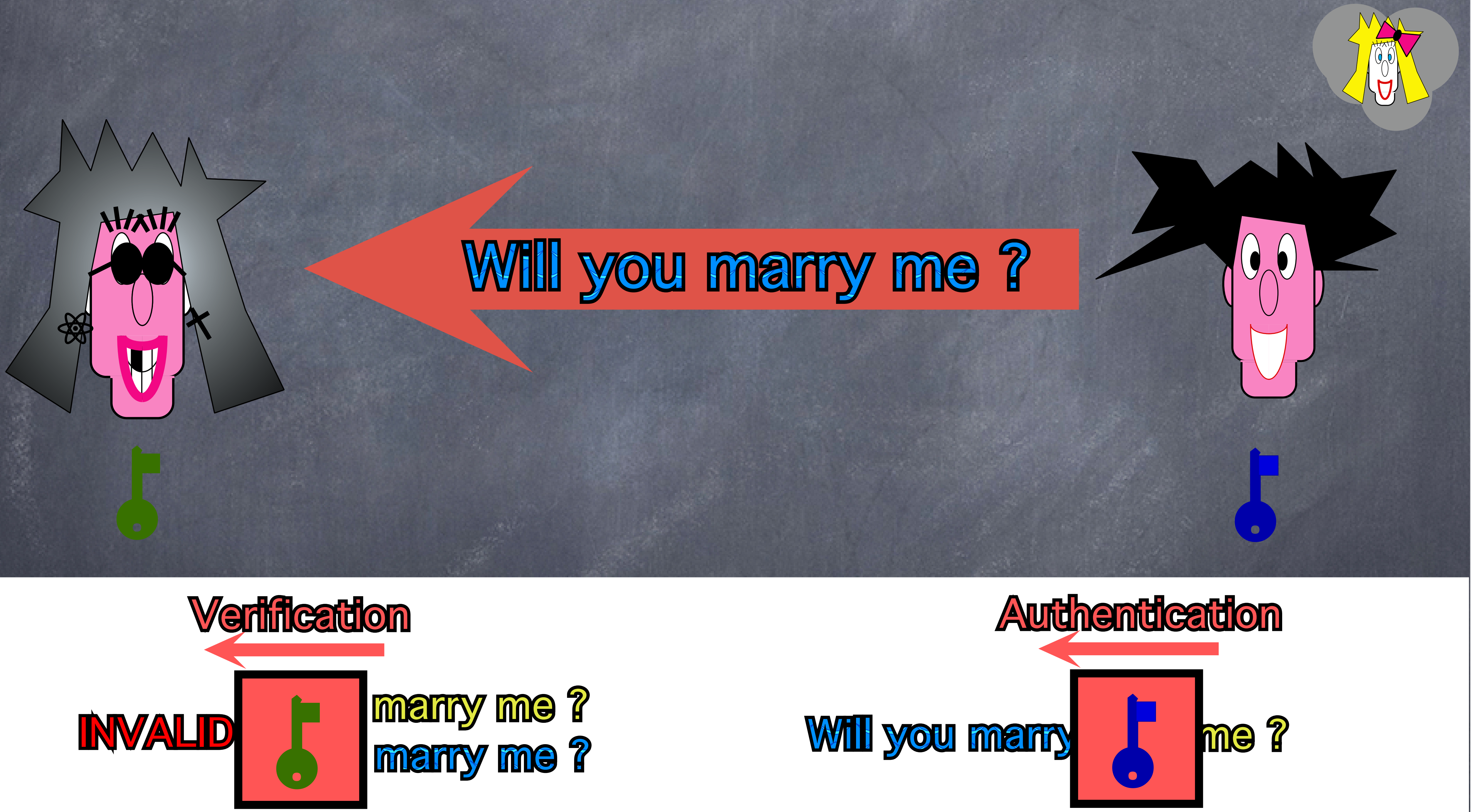
Will you marry me ?

Divorce your wife first !

The papers are in the mail...

OK, I will !

Authentication



Will you marry me ?

Verification

INVALID

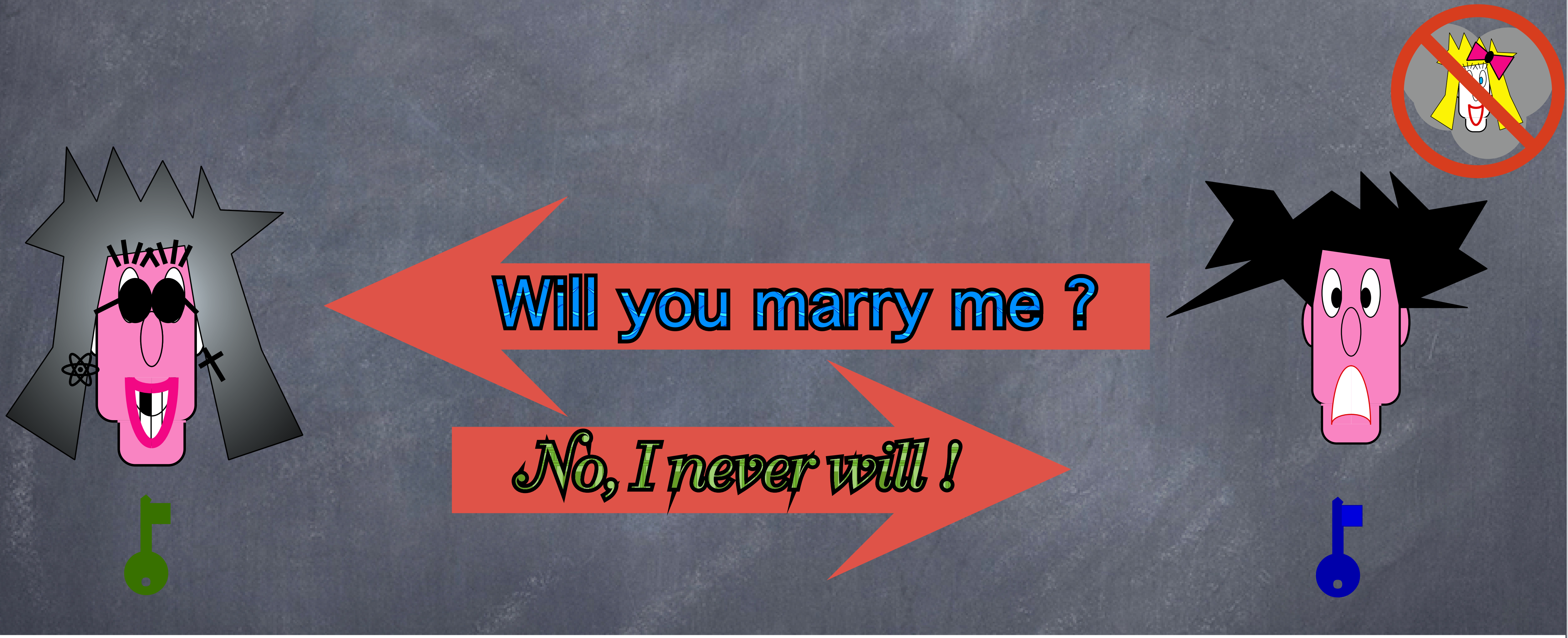
marry me ?

marry me ?

Authentication

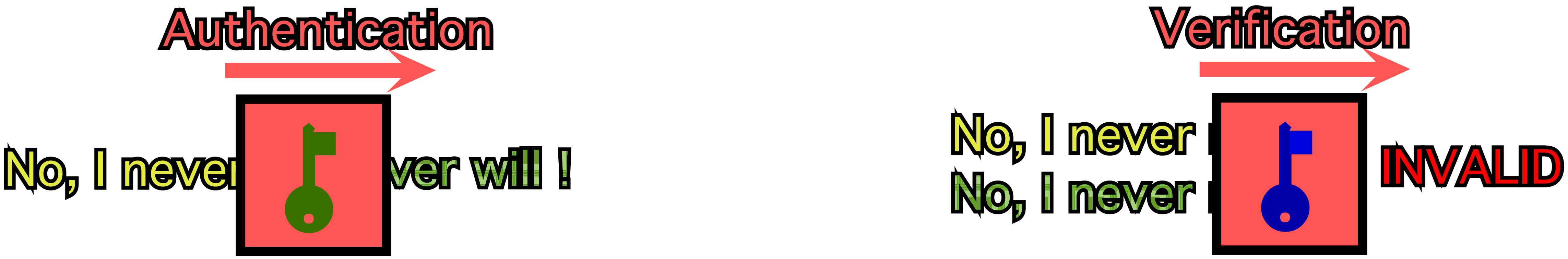
Will you marry me ?

Authentication



Will you marry me ?

No, I never will !



Authentication

No, I never ver will !

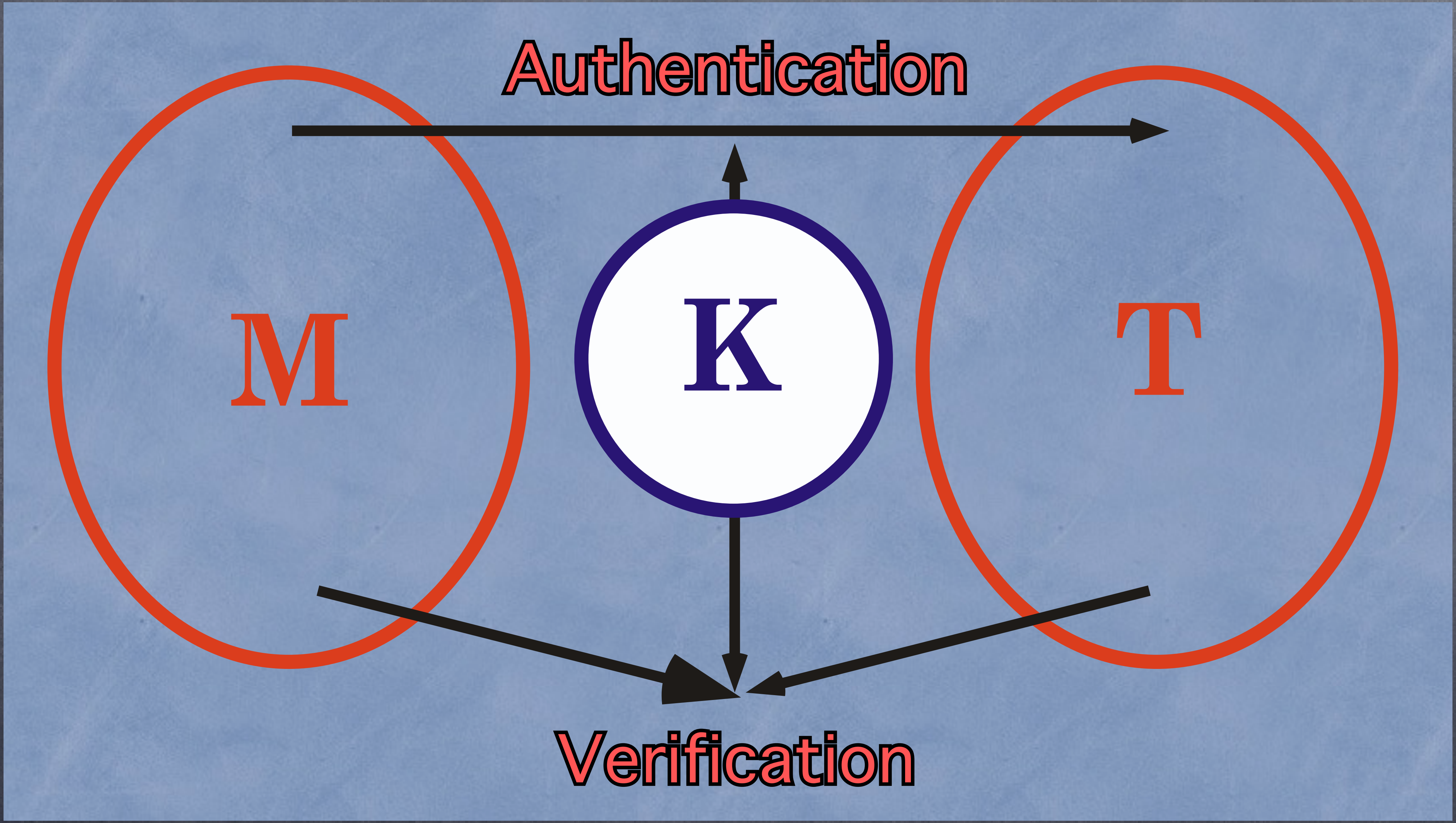
Verification

No, I never

No, I never

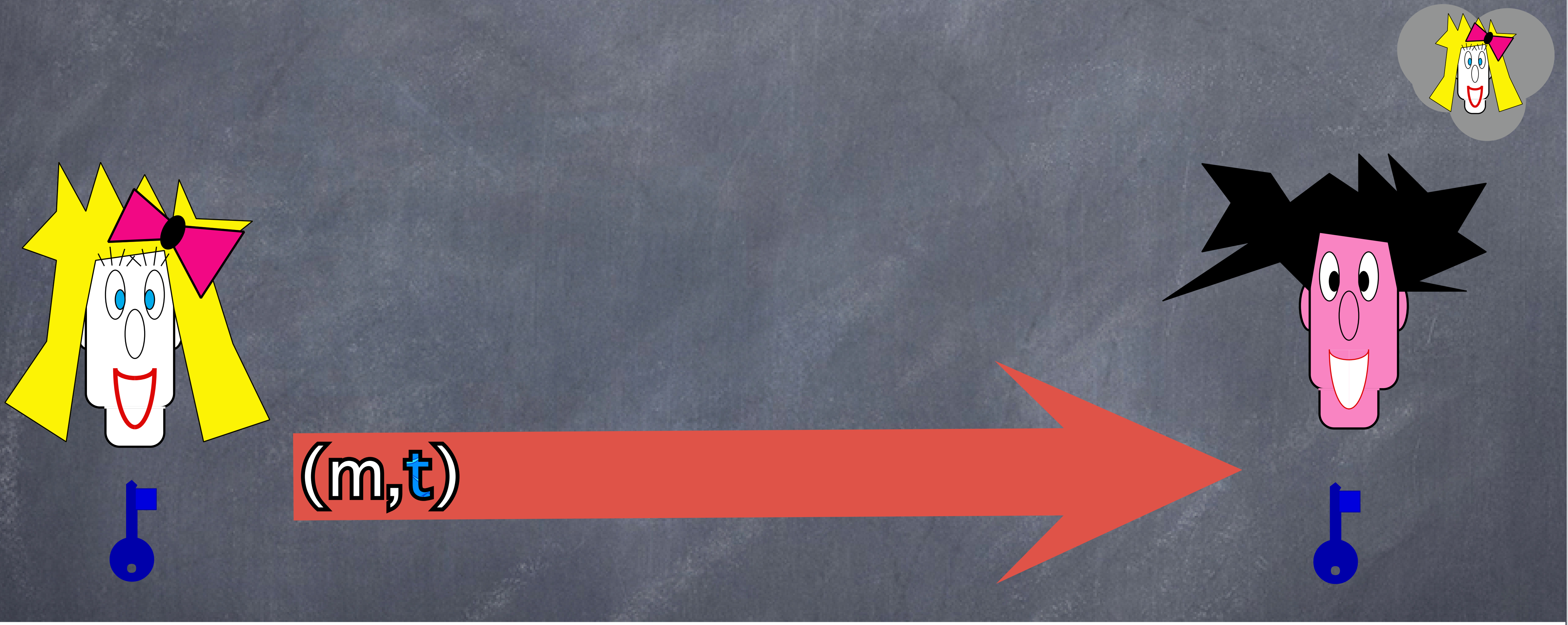
INVALID

Symmetric Authentication



Information Theoretical Security

Symmetric Authentication



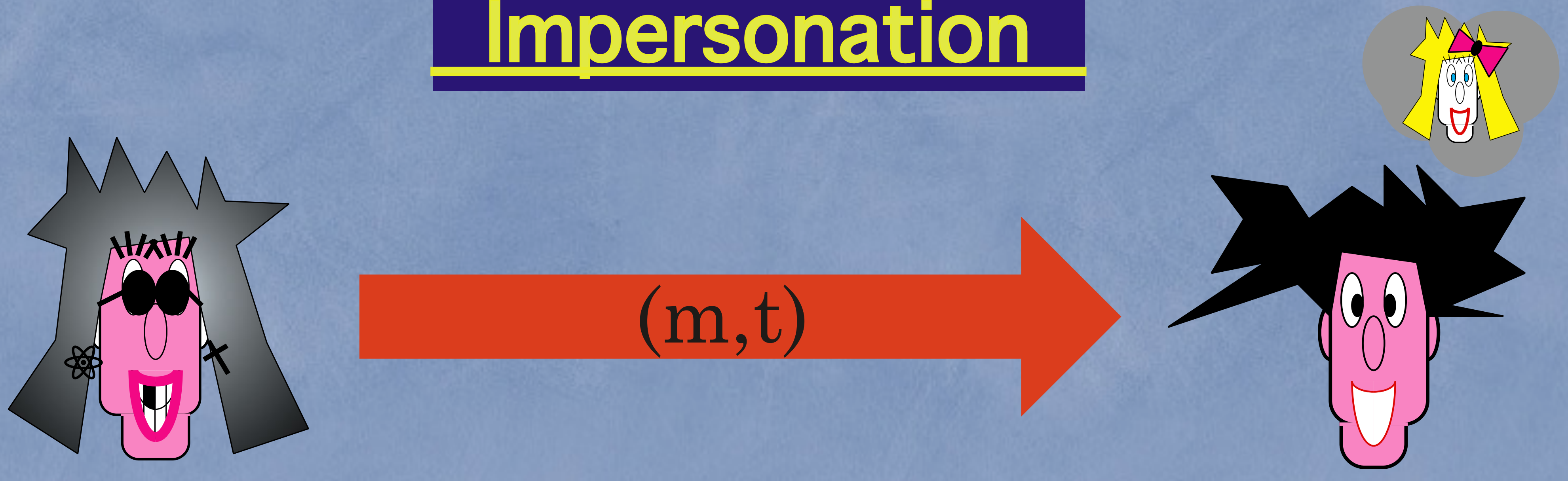
Authentication

t := A🔑(m)

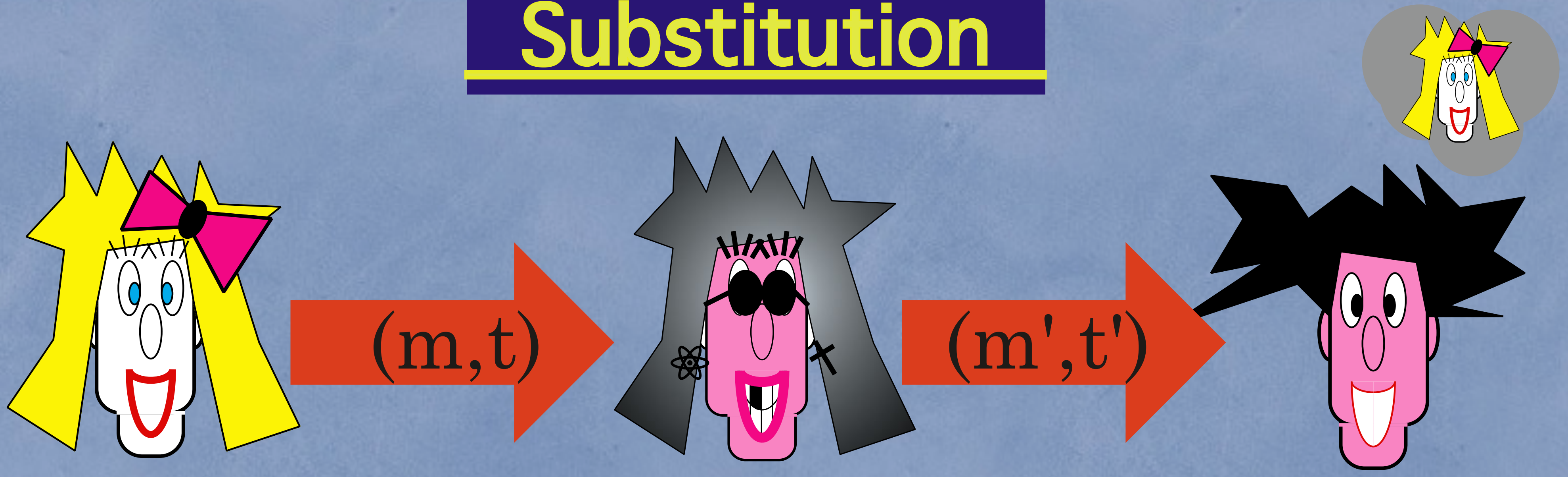
Verification

t = A🔑(m) ?

Impersonation

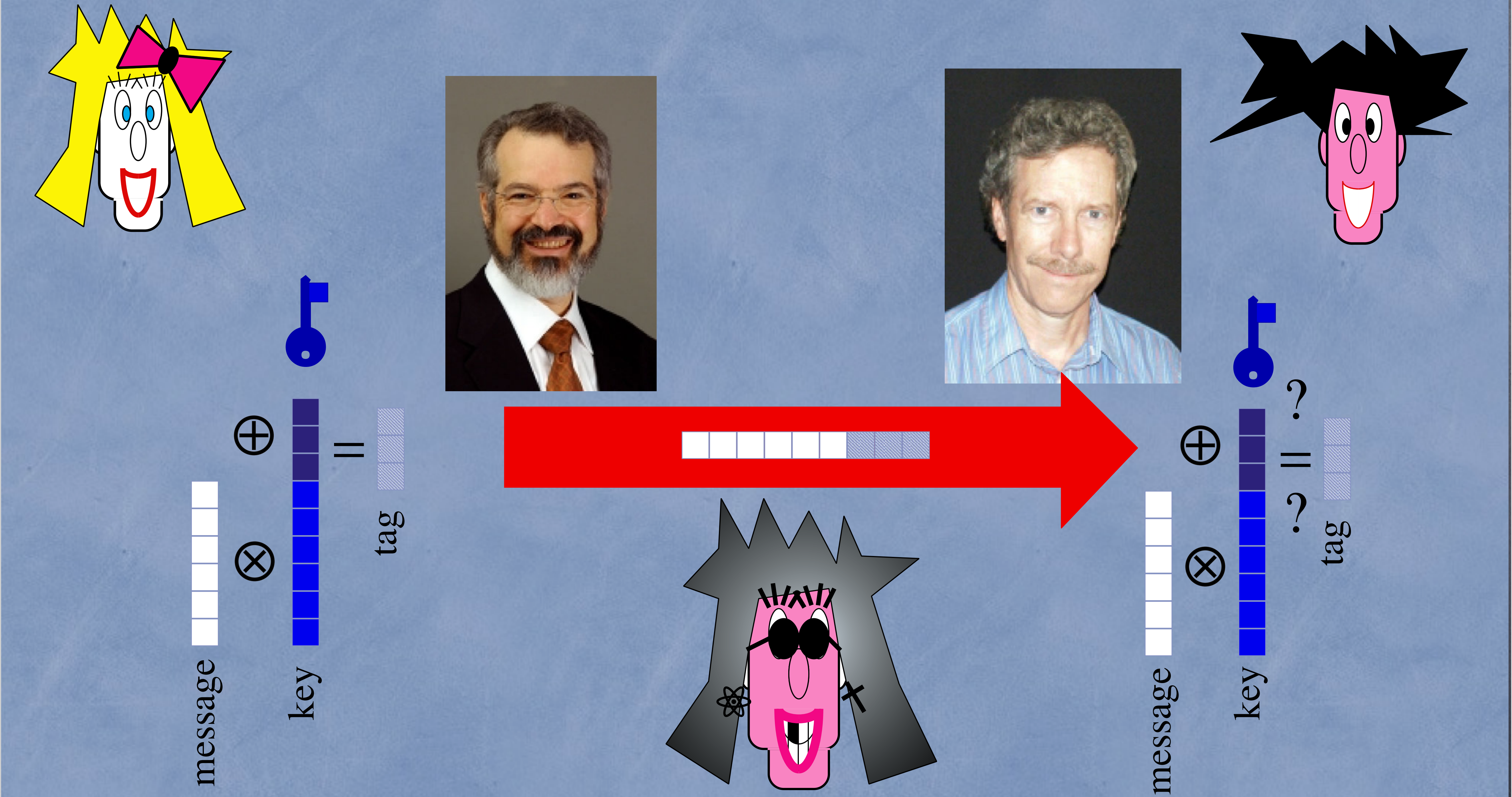


Substitution

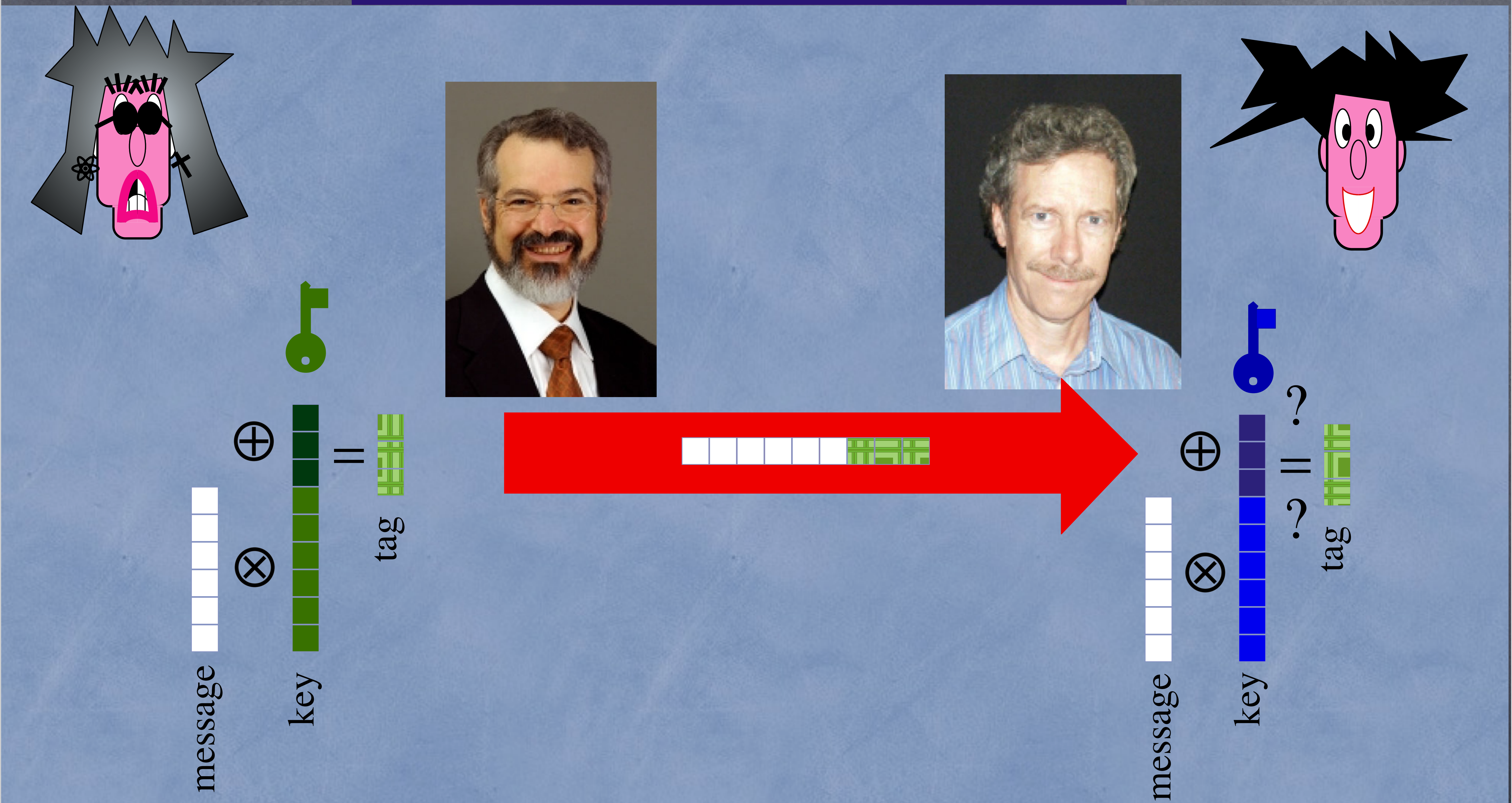


Information Theoretical Security

Wegman-Carter
One-Time Authentication



Wegman-Carter One-Time Authentication



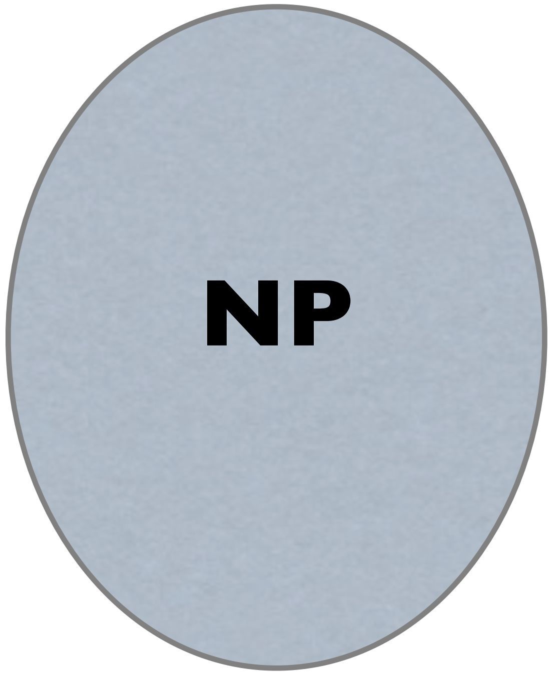
Complexity

Theoretical

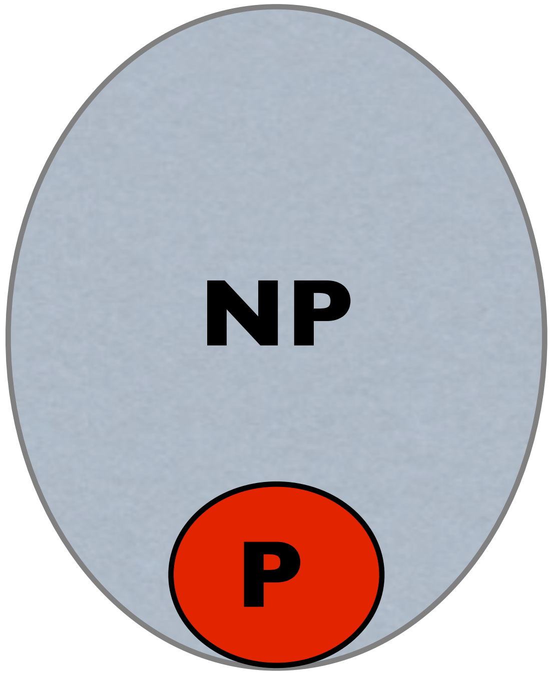
Cryptography

Complexity Theory

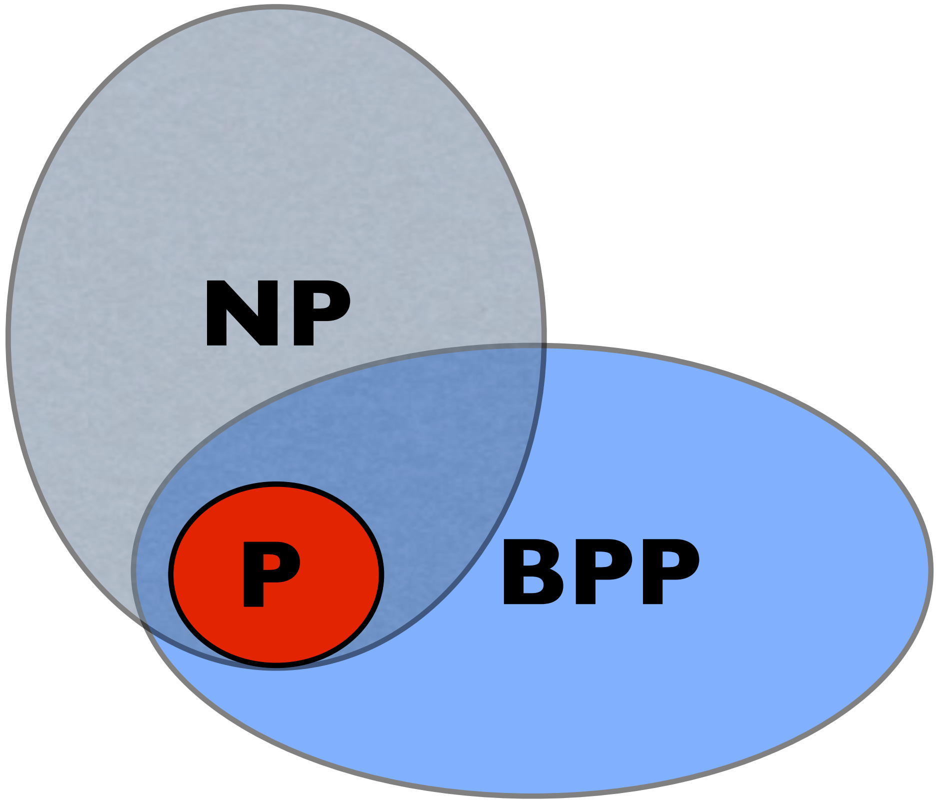
Complexity Theory



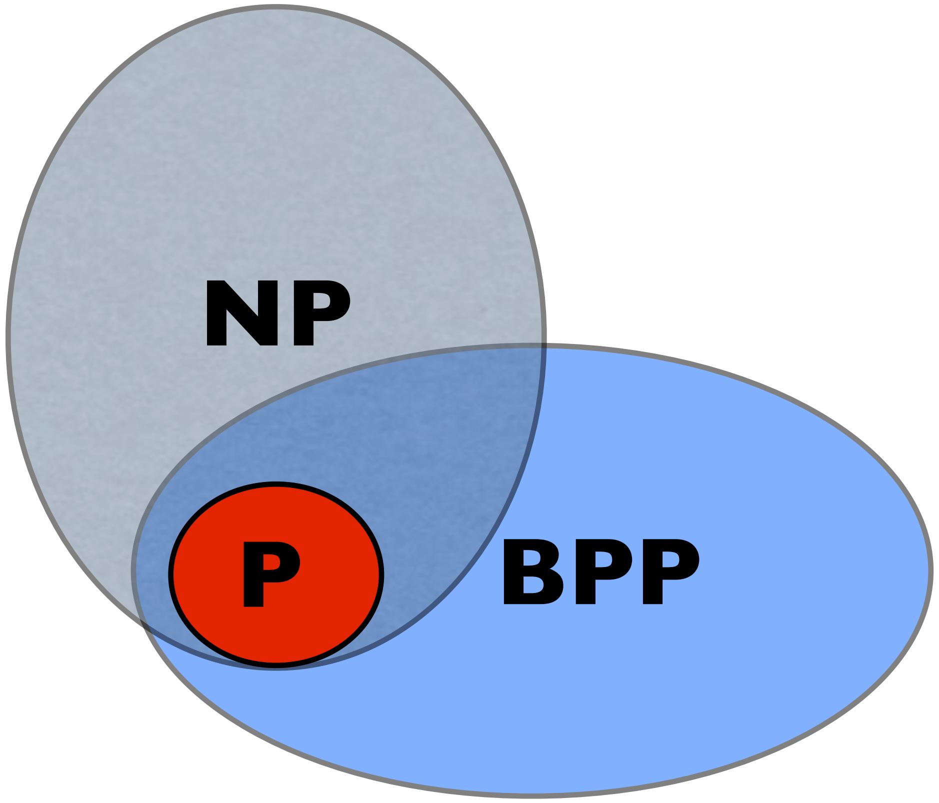
Complexity Theory



Complexity Theory



Complexity Theory

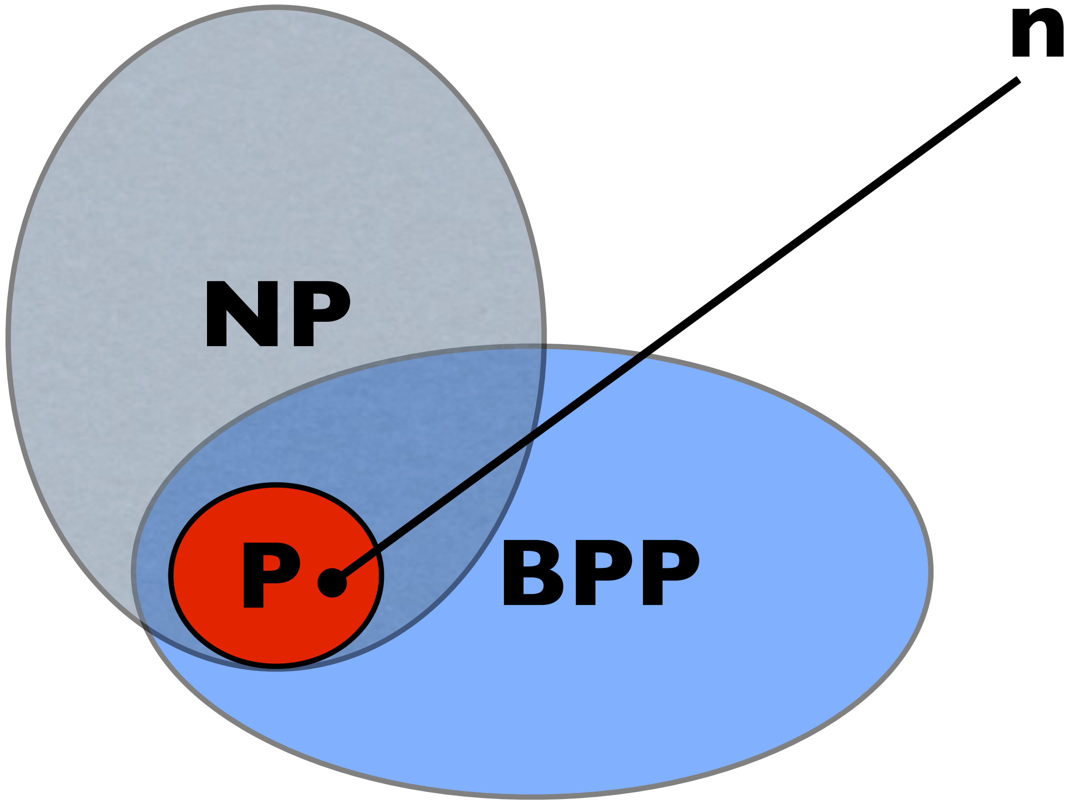


Bounded-Probability Polynomial-time

$\forall x \in L$ Prob[M(x)=accept]$\approx$1

$\forall x \notin L$ Prob[M(x)=accept]$\approx$0

Complexity Theory



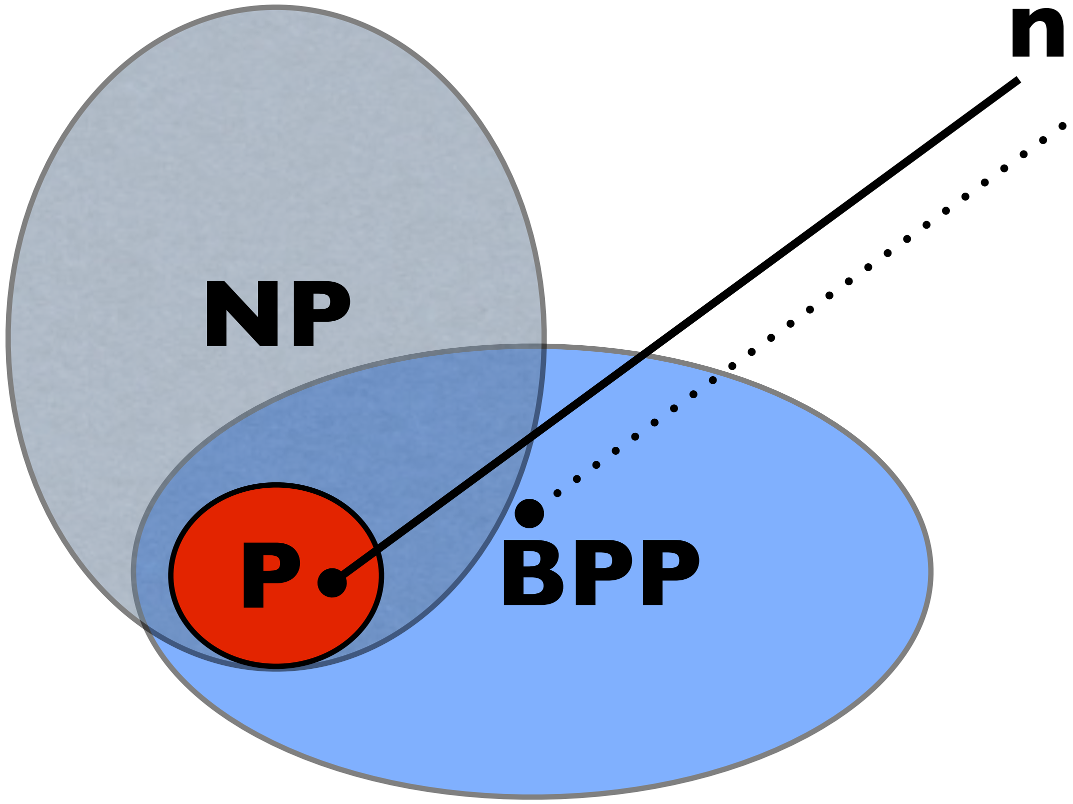
Deciding if a number is prime

Bounded-Probability Polynomial-time

$\forall x \in L$ **Prob[M(x)=accept]**≈ 1

$\forall x \notin L$ **Prob[M(x)=accept]**≈ 0

Complexity Theory



Deciding if a number is prime

Bounded-Probability Polynomial-time

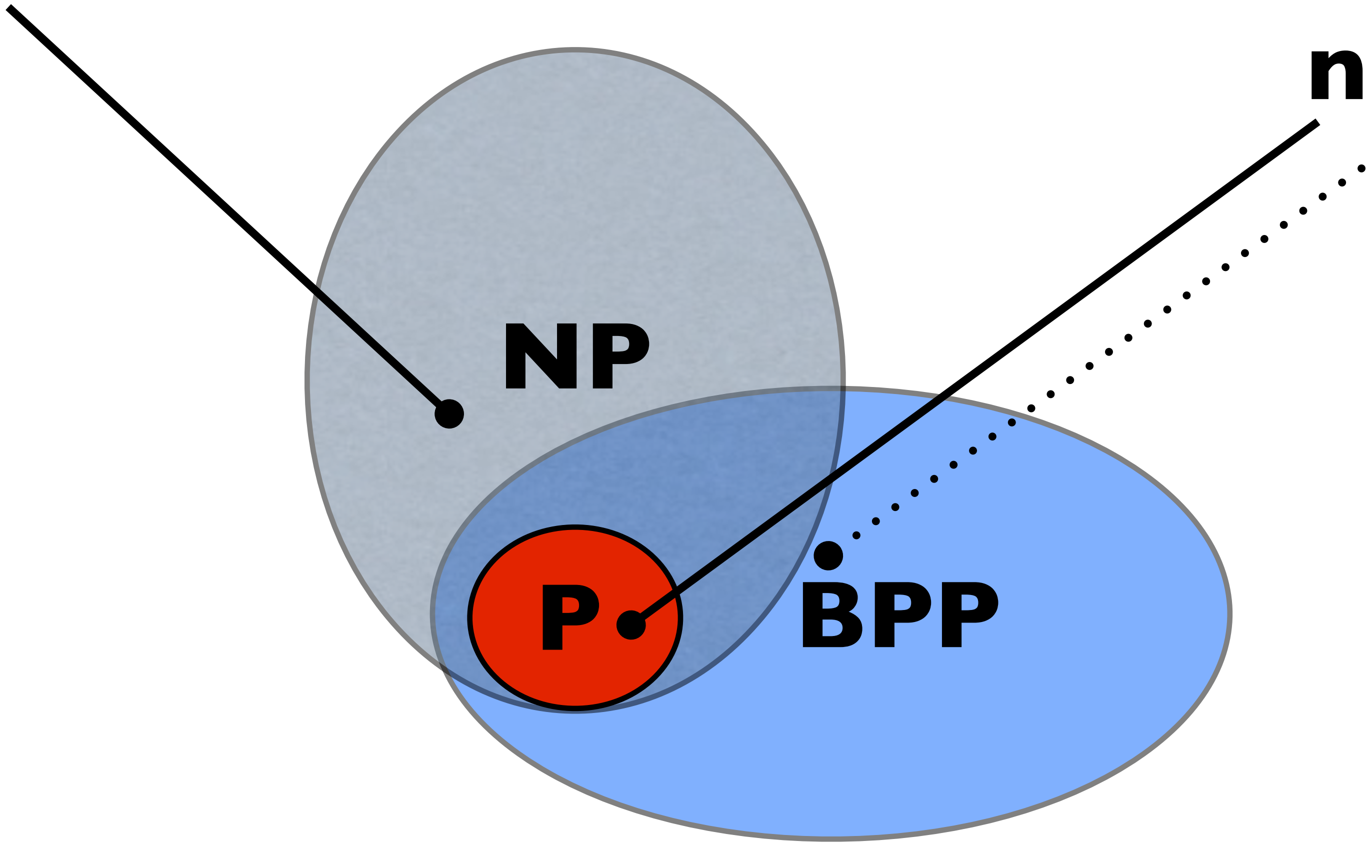
$\forall x \in L \ \mathrm{Prob}[M(x)=\mathrm{accept}] \approx 1$

$\forall x \notin L \ \mathrm{Prob}[M(x)=\mathrm{accept}] \approx 0$

Complexity Theory

Decomposing a number into primes

Deciding if a number is prime

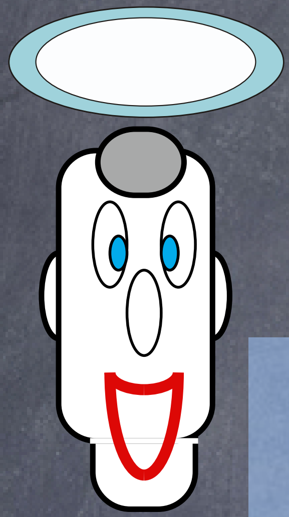


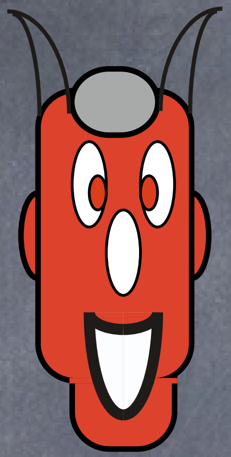
Bounded-Probability Polynomial-time

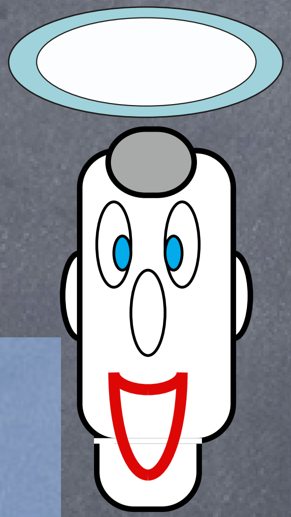
$\forall x \in L$ Prob[M(x)=accept]$\approx$1

$\forall x \notin L$ Prob[M(x)=accept]$\approx$0

Complexity Theoretical Symmetric Cryptography







• • • • •

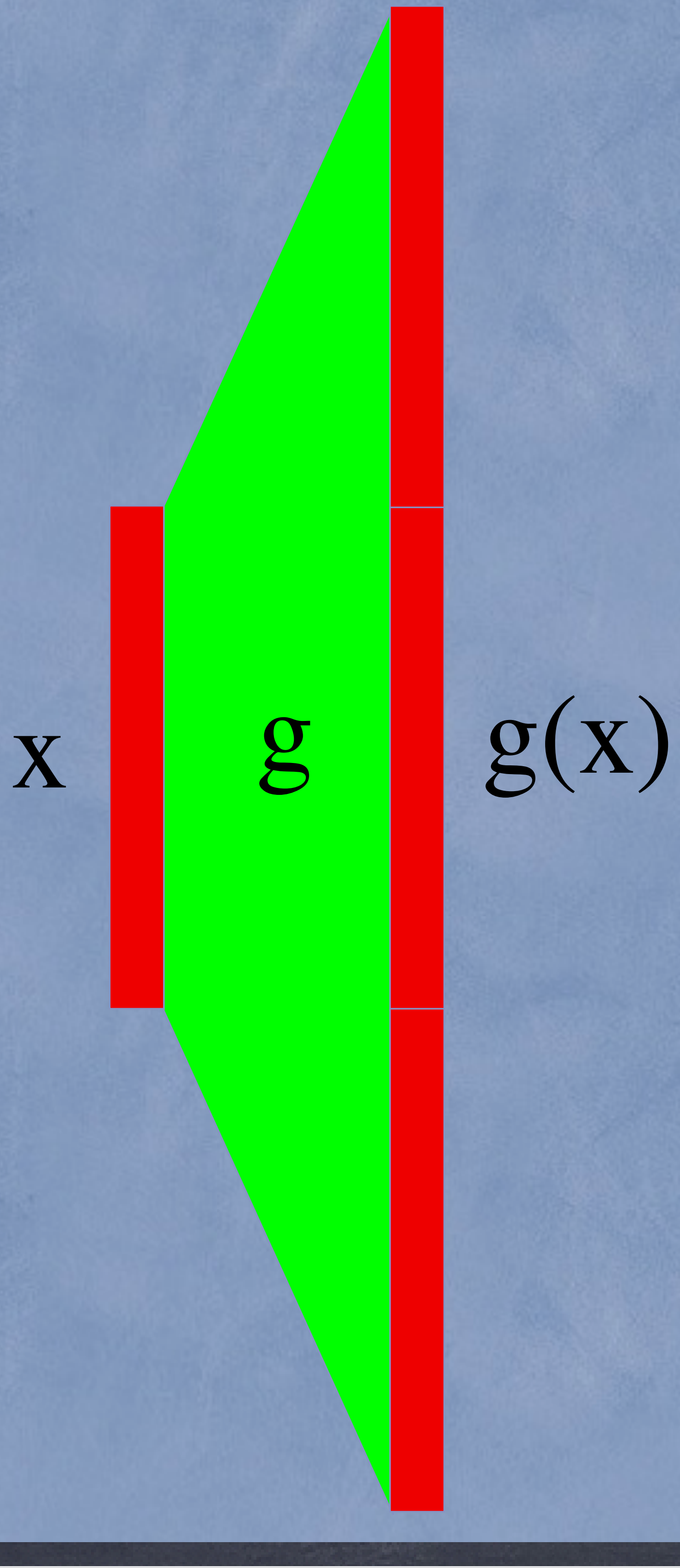
Encryption

Authentication

• • • • •

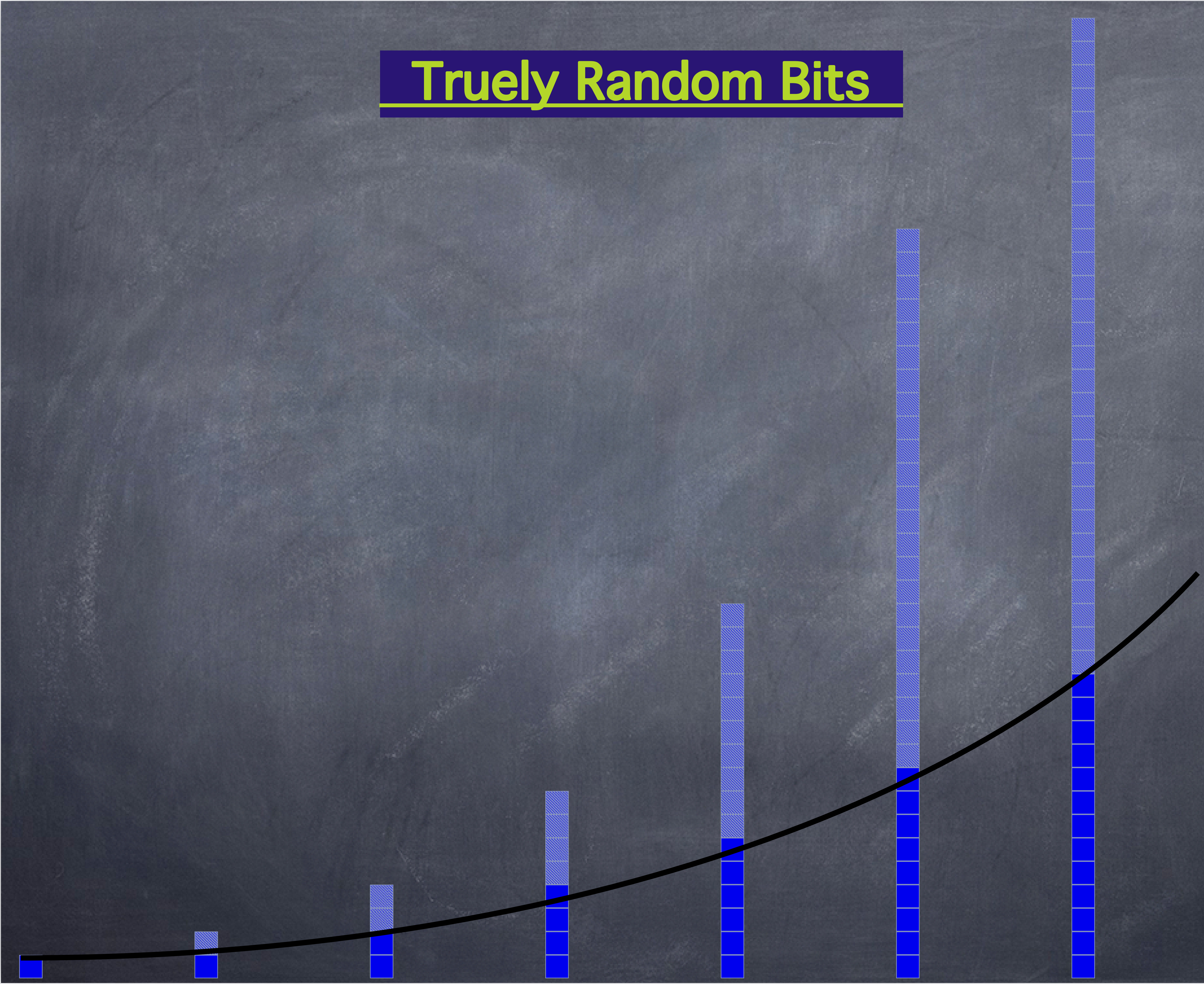
Pseudo-random Bit Generator

RANDOM

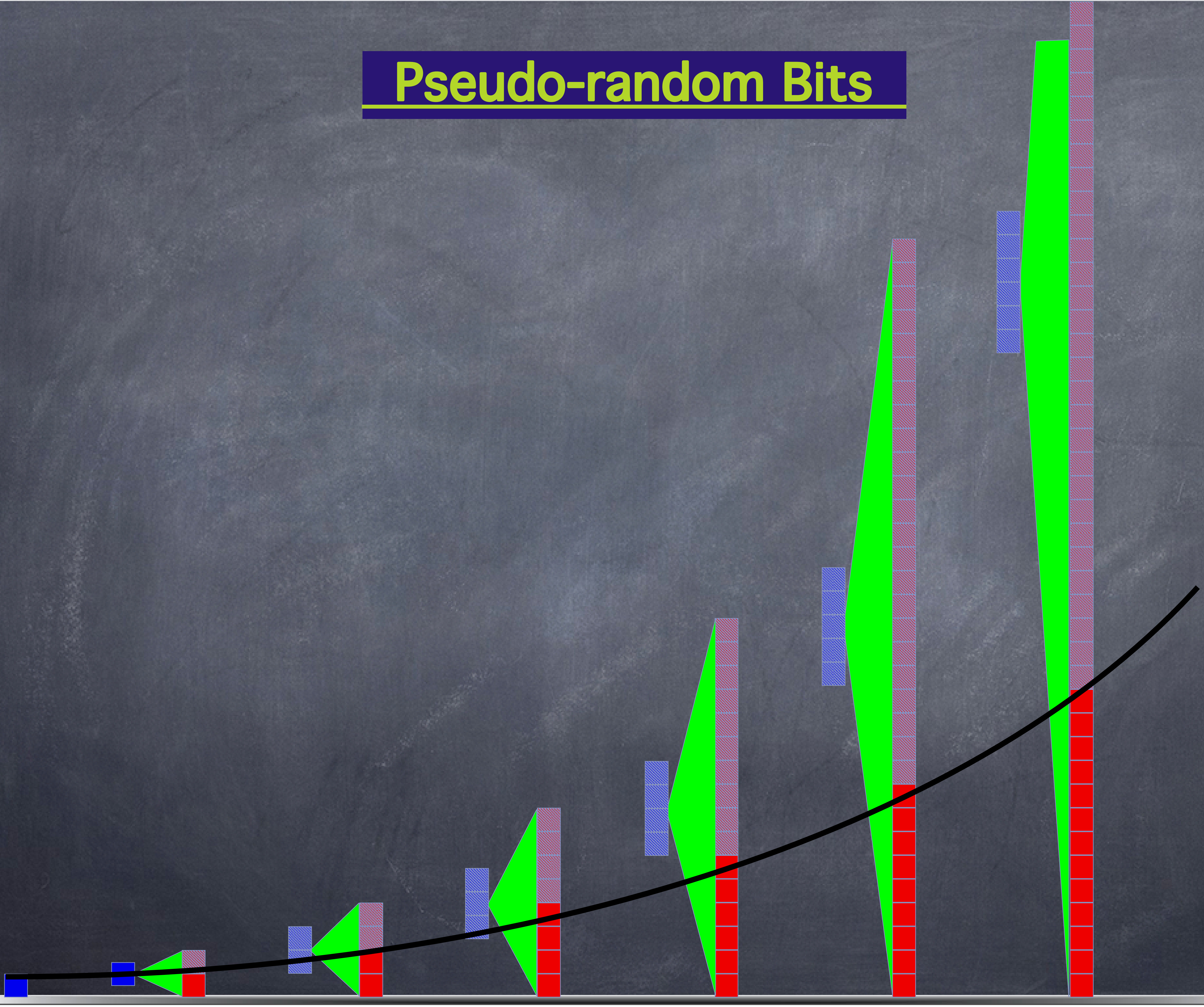


SEEMS
RANDOM

Truely Random Bits

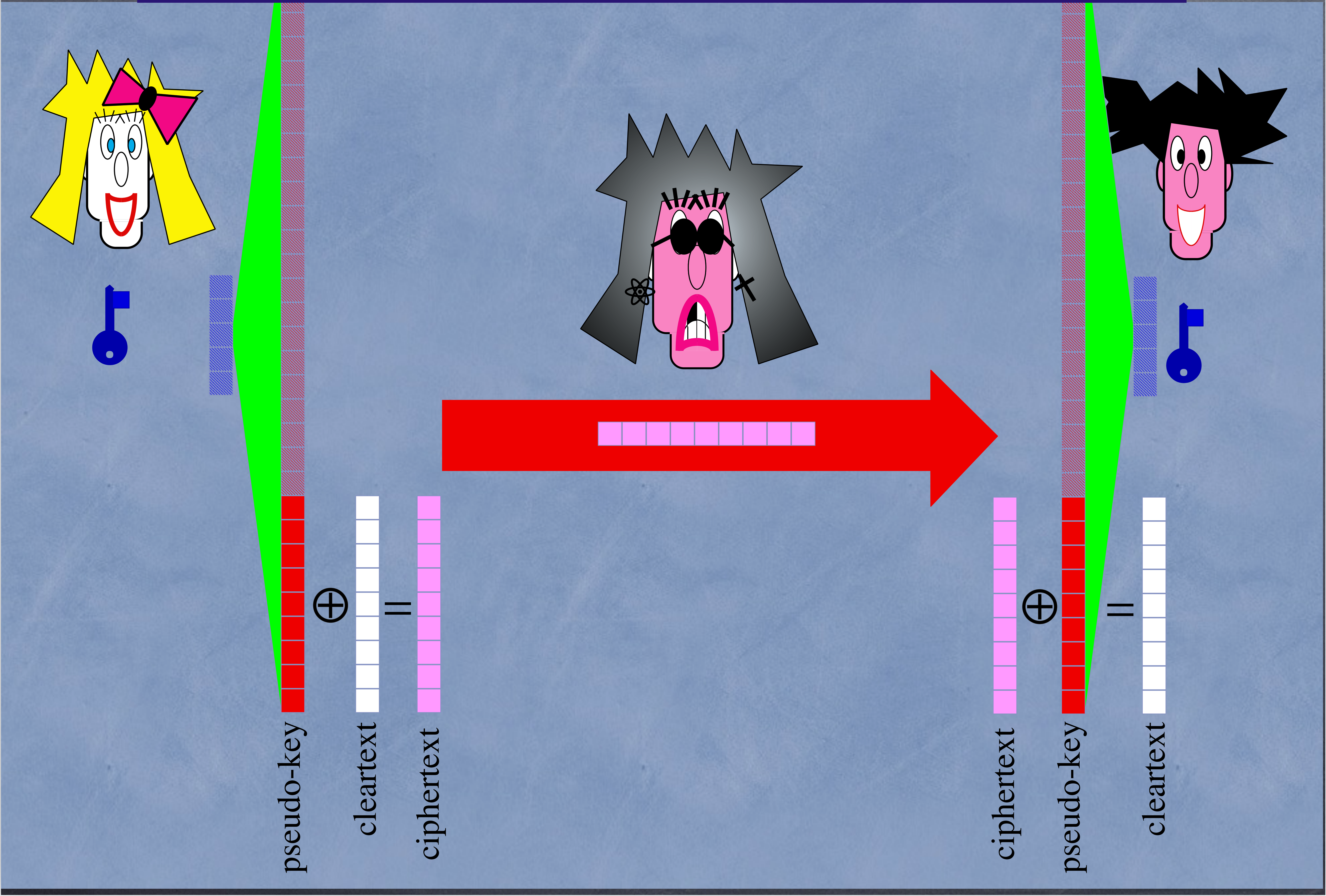


Pseudo-random Bits

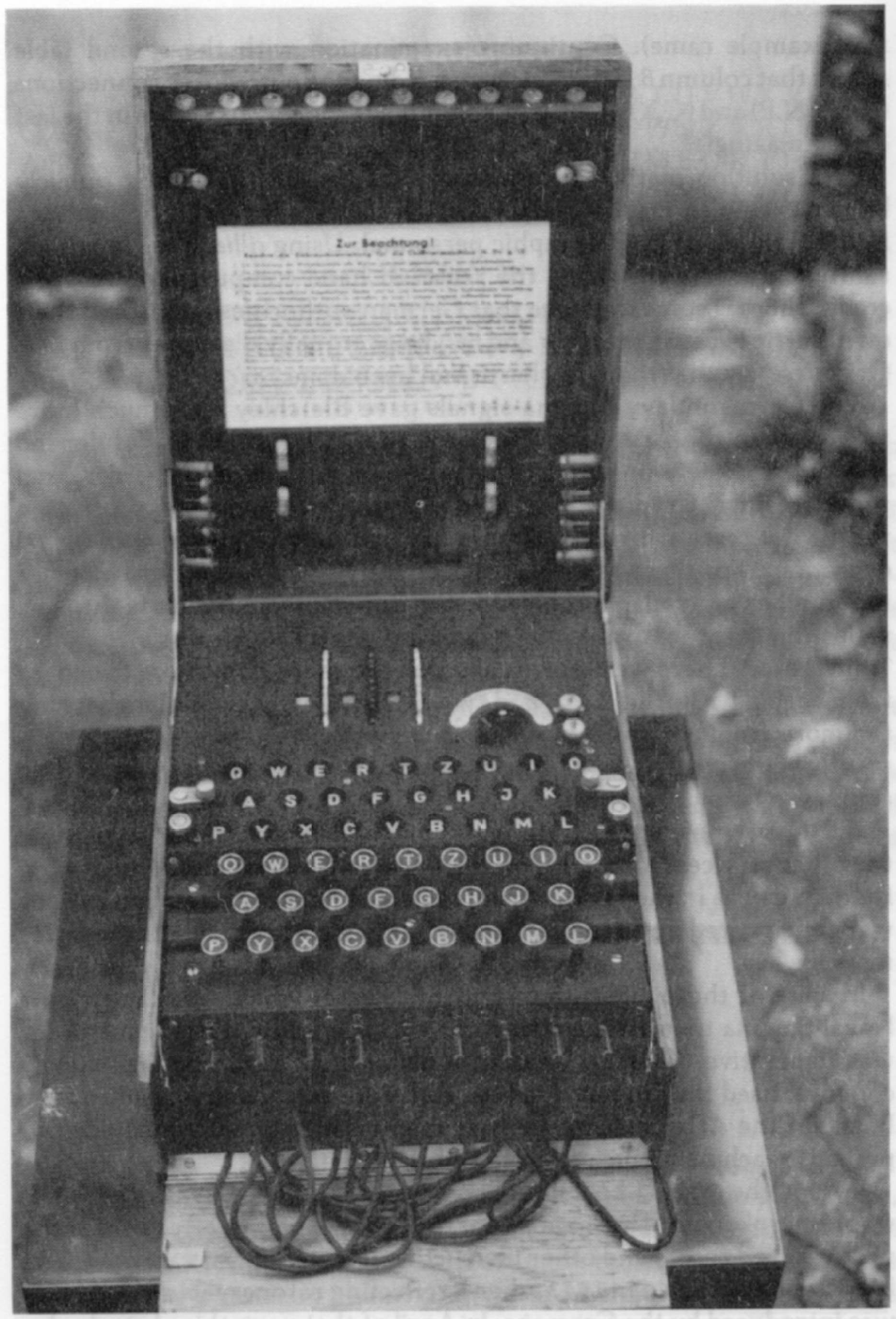


Encryption

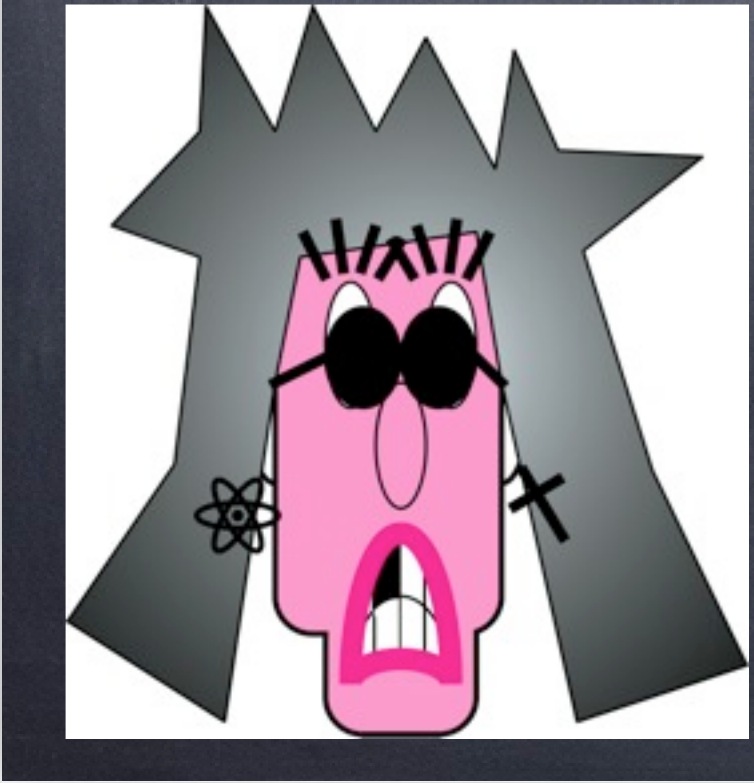
Stream Cipher from Pseudo-random Bits

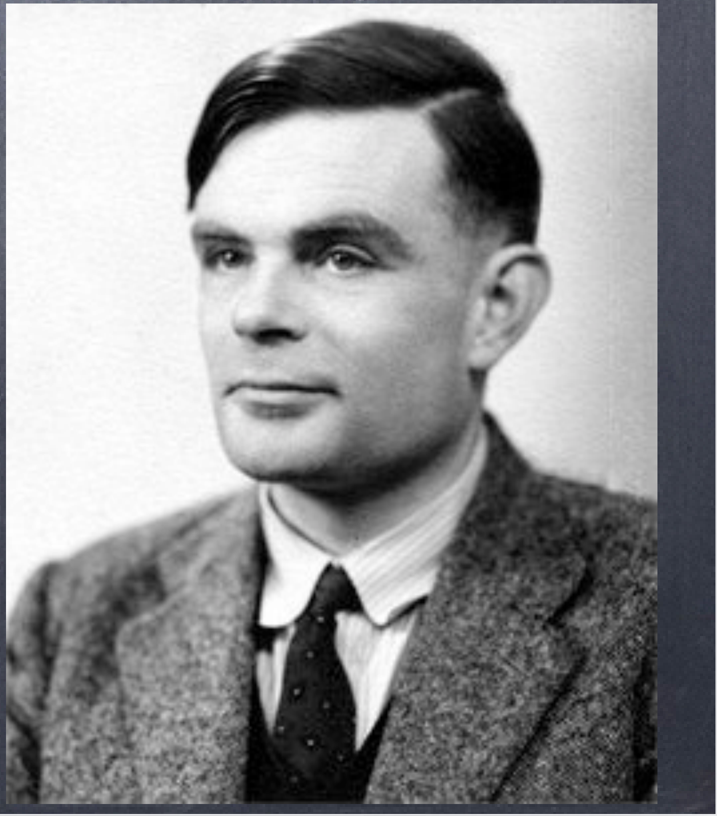


The Enigma Machine

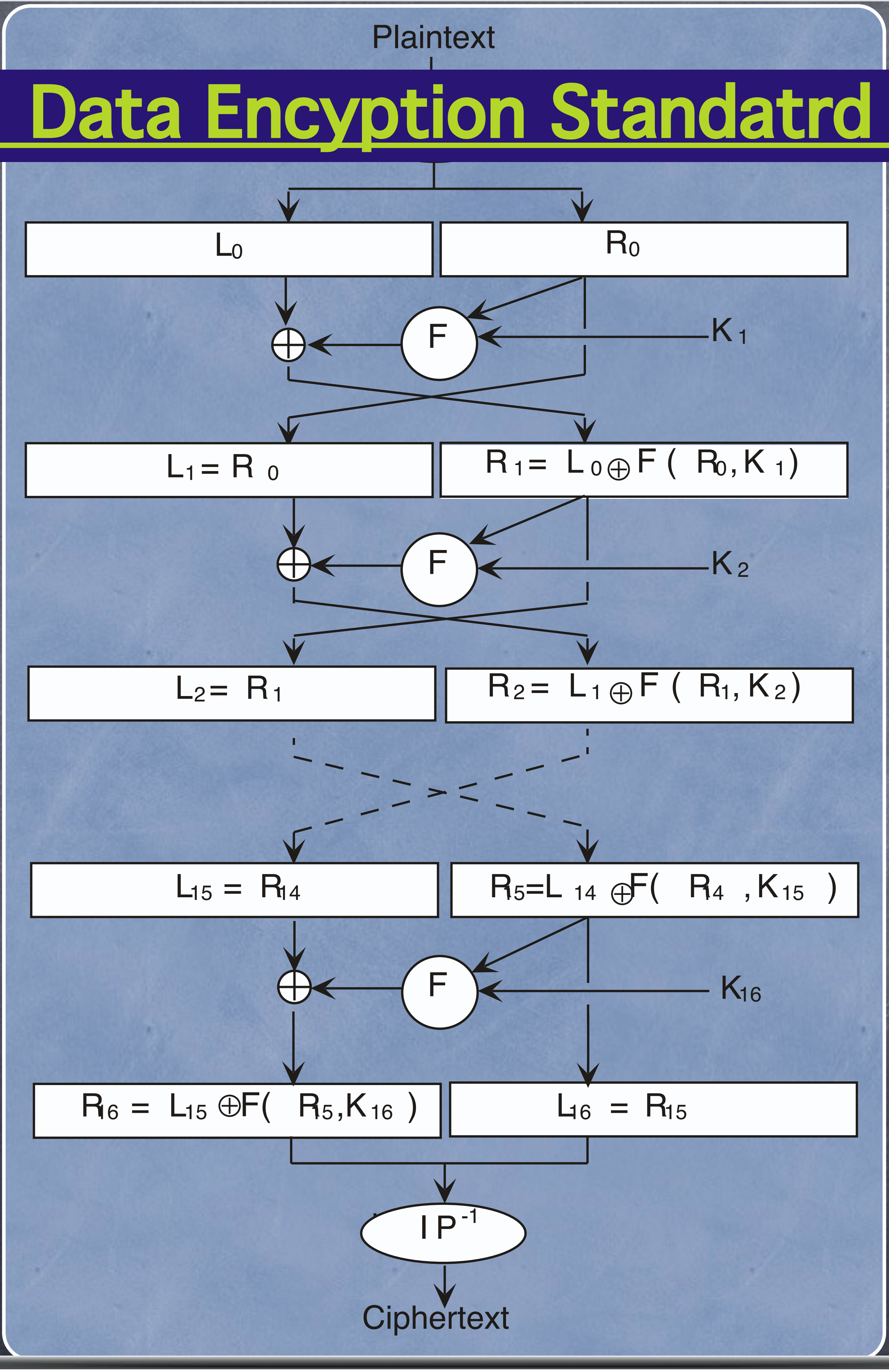


GERMAN ARMY MILITARY ENIGMA. THIS MODEL WAS THE MOST WIDELY USED VERSION OF THE GERMAN WAR-TIME ENIGMAS.





Data Encryption Standatrd


Plaintext
L_0
R_0
F
K_1
$L_1 = R_0$
$R_1 = L_0 \oplus F(R_0, K_1)$
F
K_2
$L_2 = R_1$
$R_2 = L_1 \oplus F(R_1, K_2)$
$L_{15} = R_{14}$
$R_{15} = L_{14} \oplus F(R_{14}, K_{15})$
F
K_{16}
$R_{16} = L_{15} \oplus F(R_{15}, K_{16})$
$L_{16} = R_{15}$
IP^{-1}
Ciphertext

Advanced Encyption Standatrd

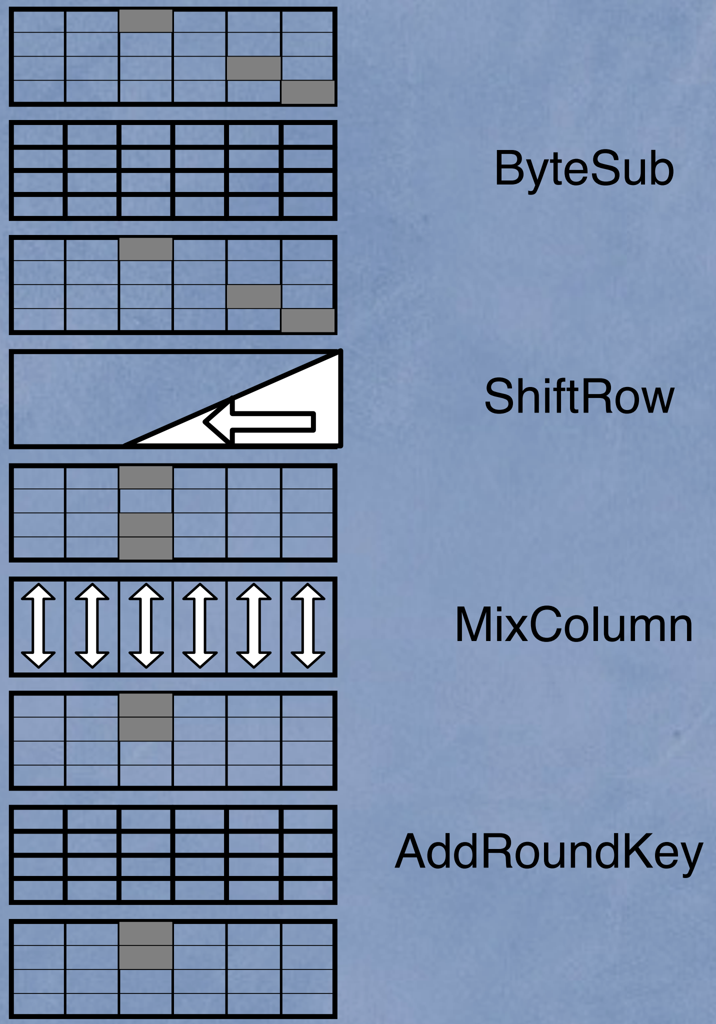
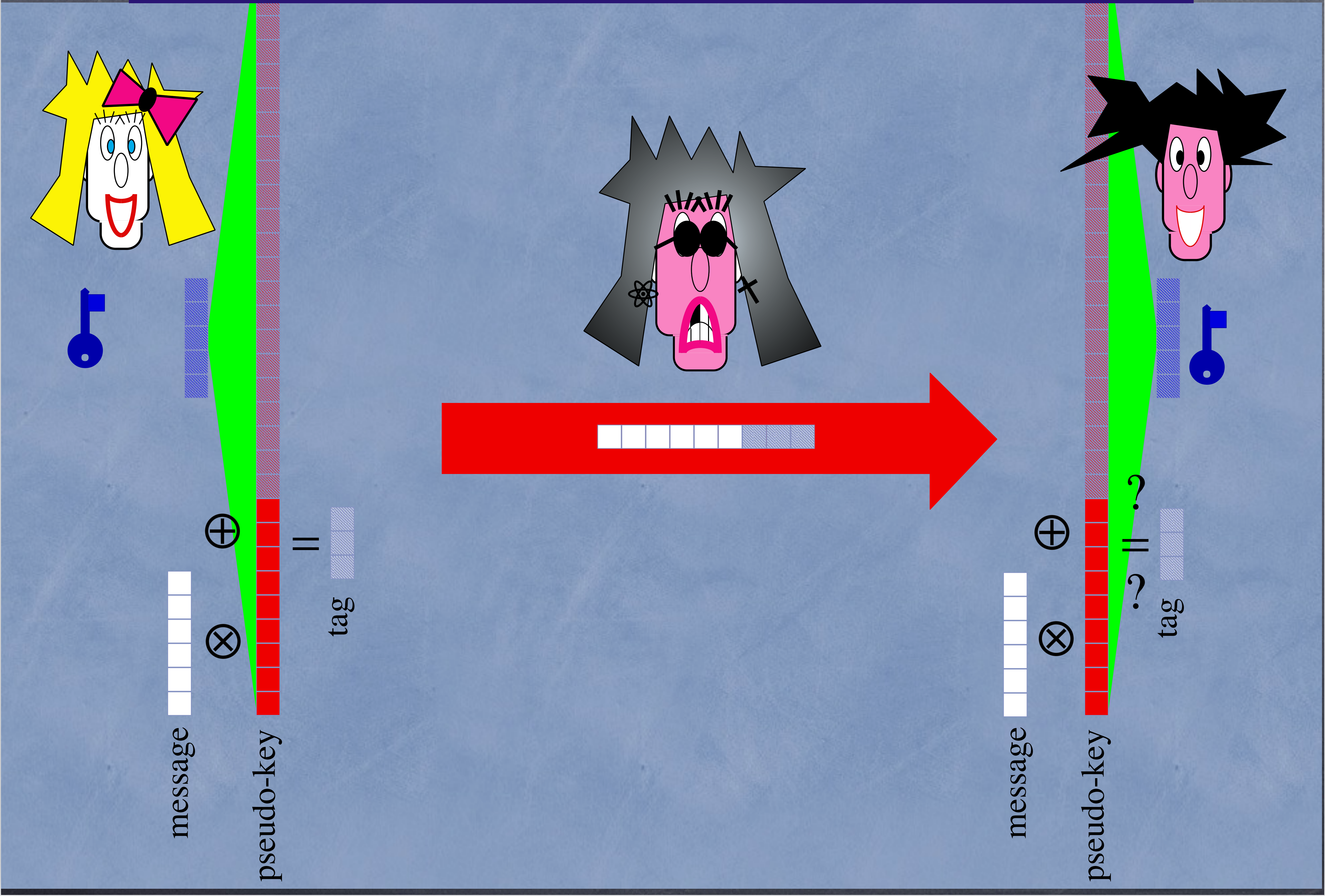


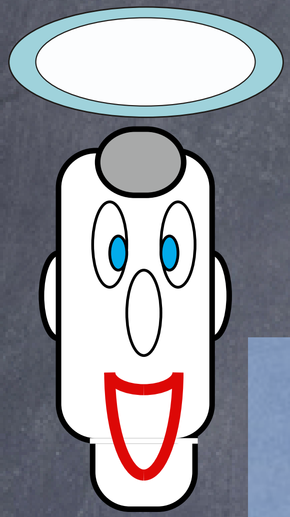
Figure 7: Propagation of activity pattern (in grey) through a single round

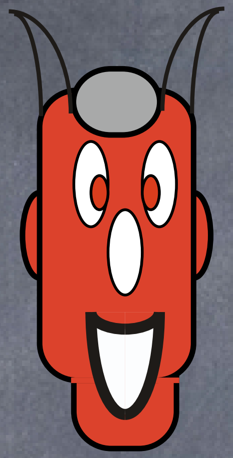
authentication

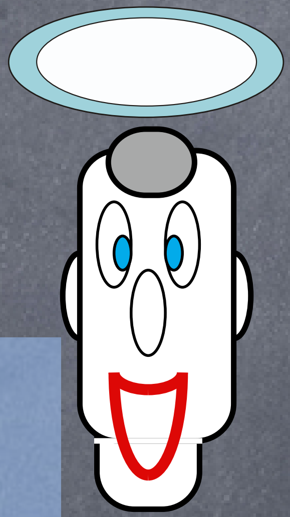
Authentication from Pseudo-random Bits



Complexity Theoretical Asymmetric Cryptography







● ● ● ● ●

public key distribution

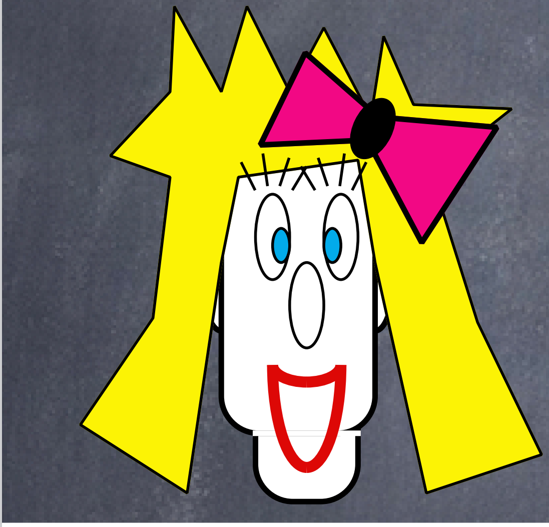
asymmetric encryption

asymmetric authentication

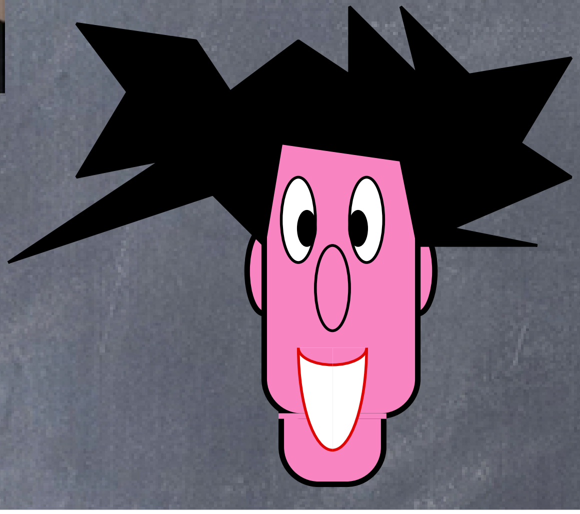
● ● ● ● ●

Public Key Distribution

Public-Key Distribution

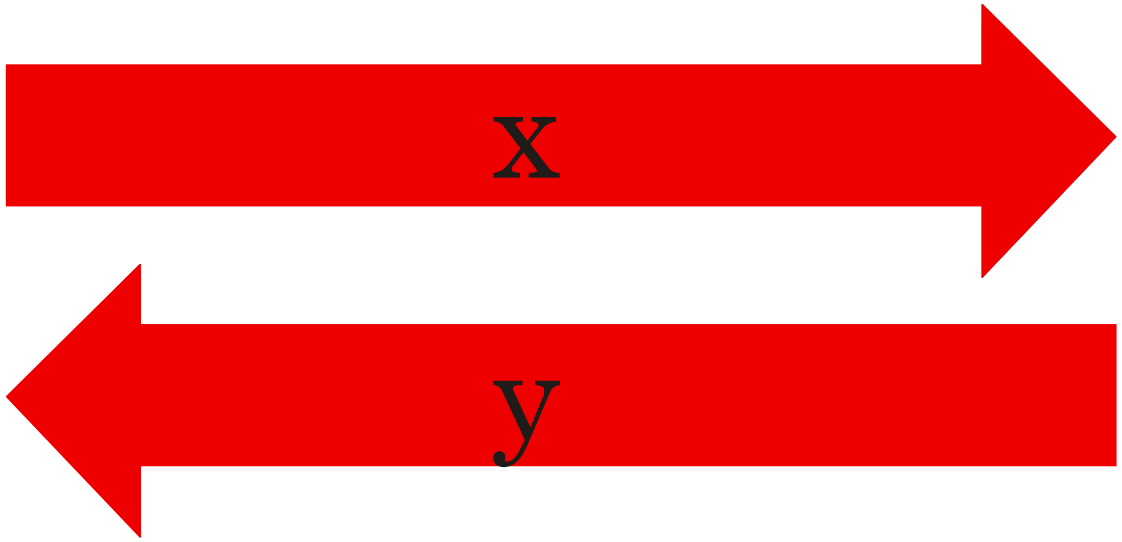


p



$x := f(p,a)$

$y := f(p,b)$



x

y

$k := f(y,a)$

$k := f(x,b)$

$f(f(p,a),b) = k = f(f(p,b),a)$

Diffie-Hellman Key Exchange

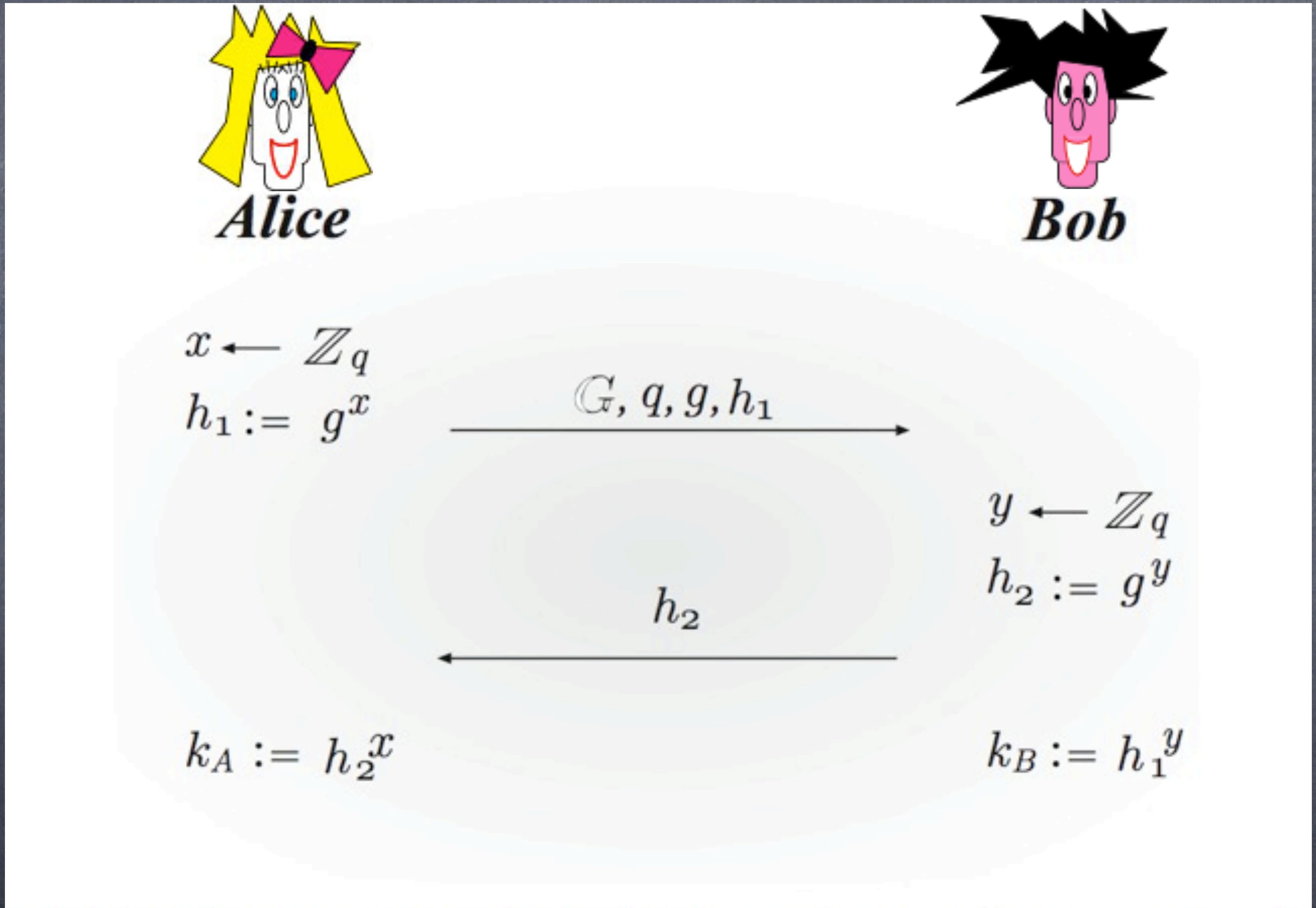


FIGURE 9.2: The Diffie-Hellman key-exchange protocol.

The Discrete Logarithm and Diffie-Hellman Assumptions

Fix a cyclic group $\mathbb{G}$ and a generator $\mathbf{g} \in \mathbb{G}$.
Given two group elements $\boldsymbol{h_1}, \boldsymbol{h_2}$, define

$$\mathbf{DH_g}(\boldsymbol{h_1},\boldsymbol{h_2}) \overset{\mathbf{def}}{=} \mathbf{g}^{\log_g h_1 \cdot \log_g h_2}.$$

That is, if $\boldsymbol{h_1} = \mathbf{g}^x$ and $\boldsymbol{h_2} = \mathbf{g}^y$ then

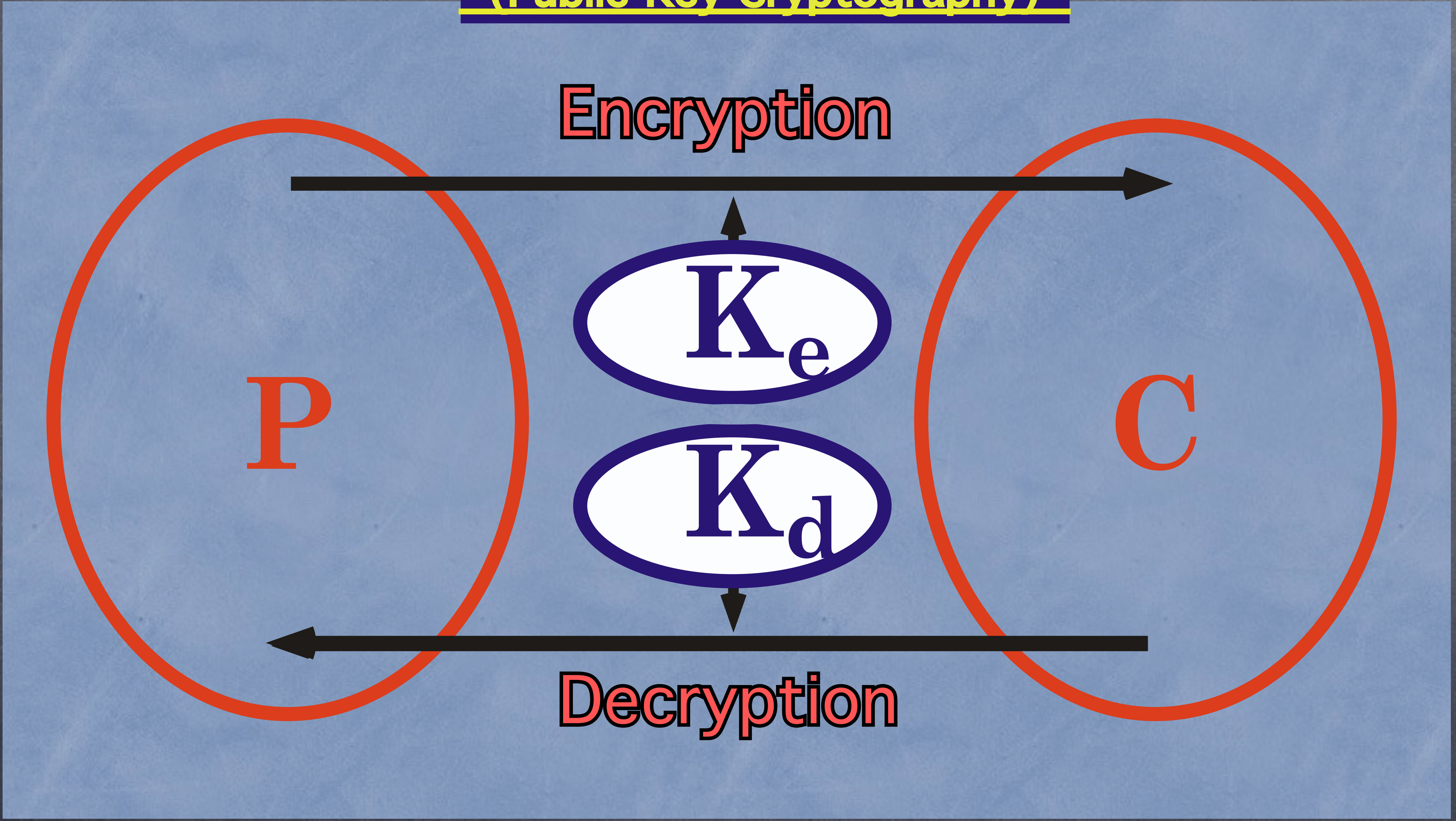
$$\mathbf{DH_g}(\boldsymbol{h_1},\boldsymbol{h_2}) = \mathbf{g}^{x \cdot y} = \boldsymbol{h_1}^y = \boldsymbol{h_2}^x \;.$$

- The ***CDH*** *problem* is to compute $\mathbf{DH_g}(\boldsymbol{h_1},\boldsymbol{h_2})$ given randomly-chosen $\boldsymbol{h_1}$ and $\boldsymbol{h_2}$.

Public Key Encryption

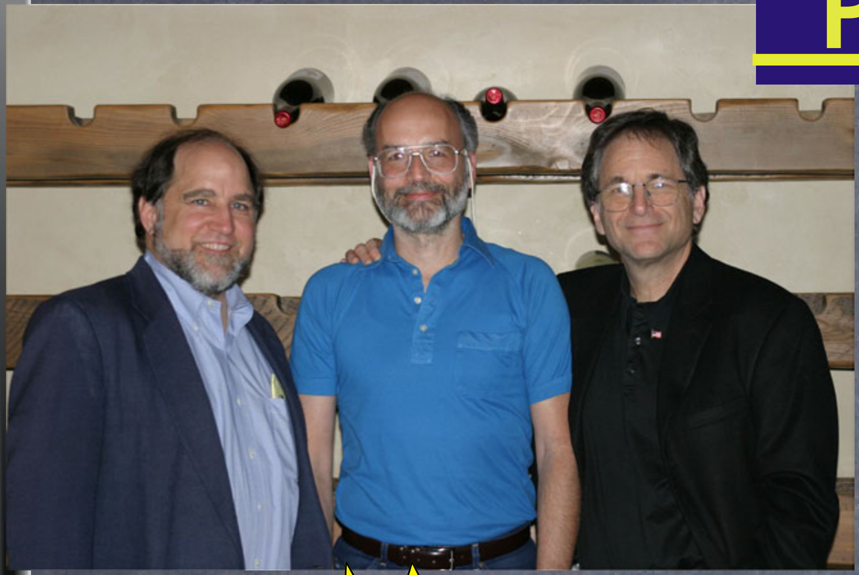
Asymmetric Encryption

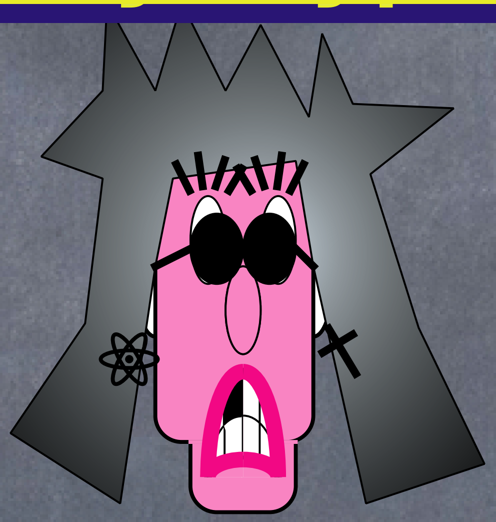
(Public-Key Cryptography)

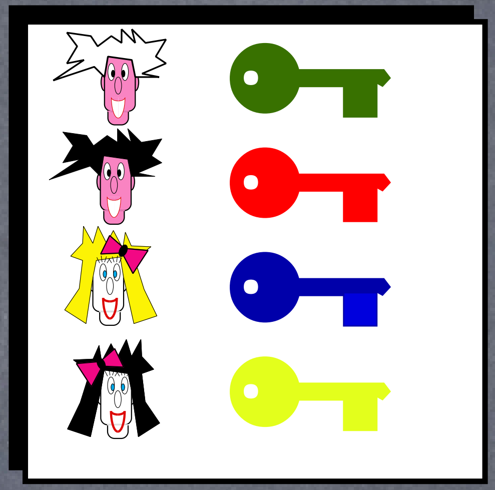


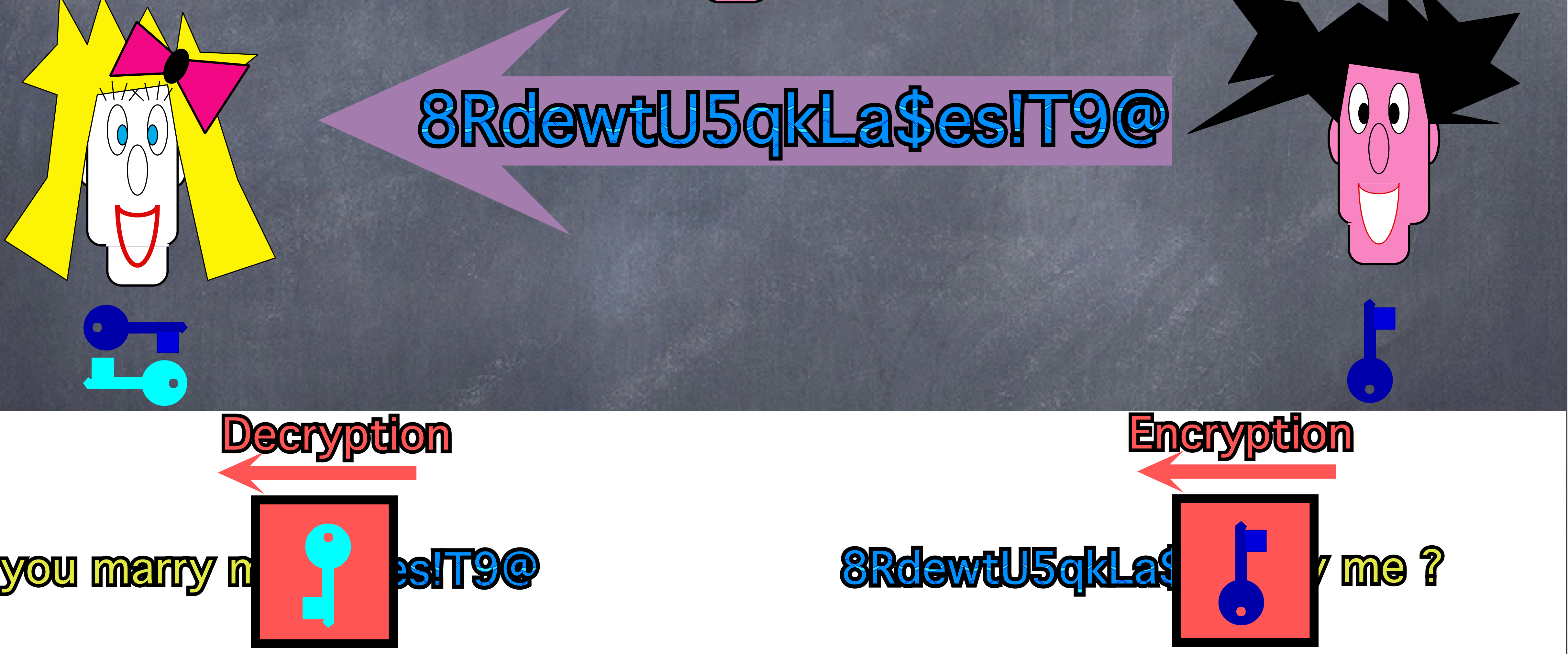
Complexity Theoretical Security

Public-Key Cryptography









8RdewtU5qkLa$es!T9@

Decryption

Will you marry m es!T9@

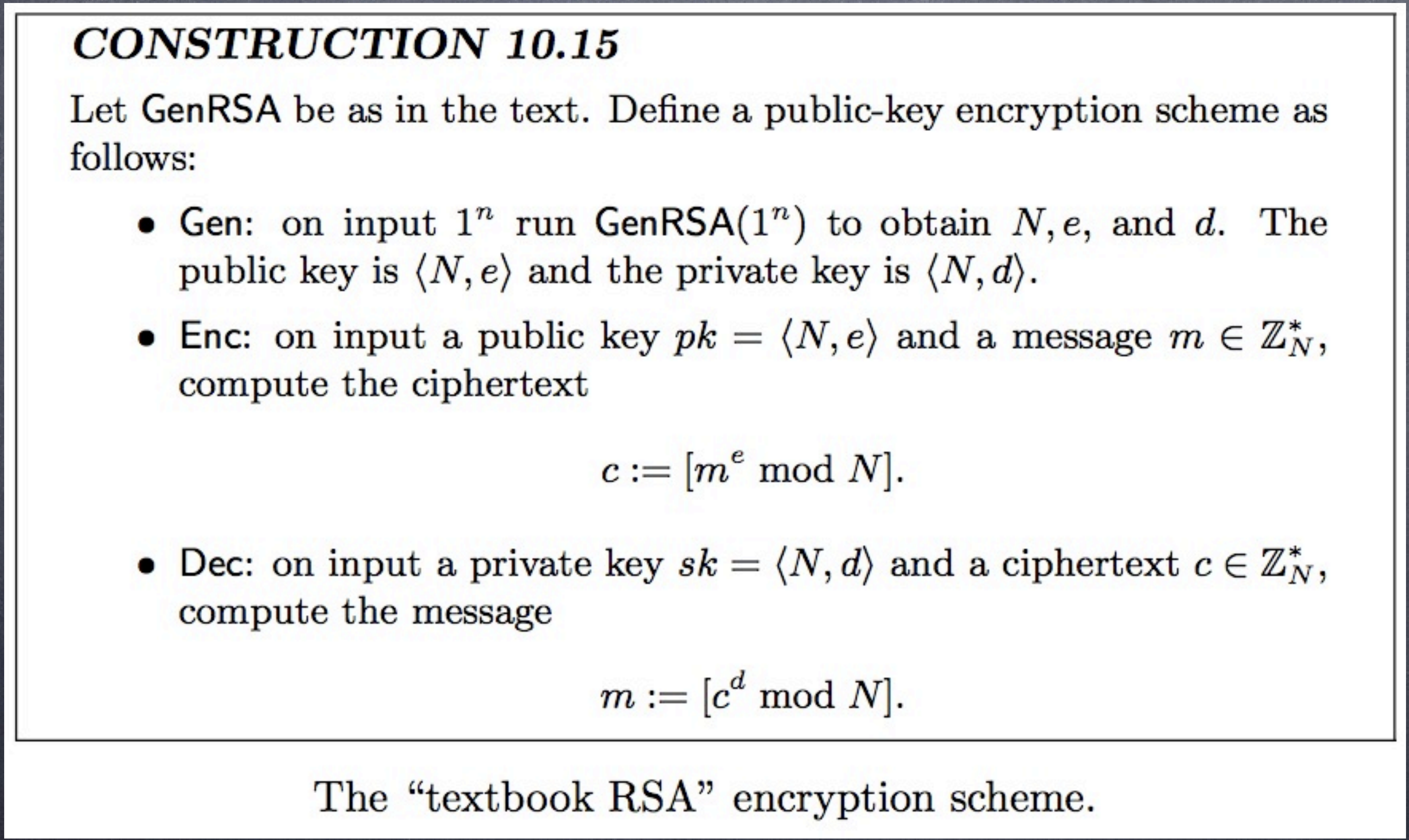
Encryption

8RdewtU5qkLa$ y me ?

RSA Encryption

Ron Rivest, Adi Shamir and Len Adleman

RSA Encryption



CONSTRUCTION 10.15

Let GenRSA be as in the text. Define a public-key encryption scheme as follows:

- Gen: on input 1^n run GenRSA(1^n) to obtain N, e, and d. The public key is $\langle N, e\rangle$ and the private key is $\langle N, d\rangle$.
- Enc: on input a public key $pk = \langle N, e\rangle$ and a message $m \in \mathbb{Z}_N^*$, compute the ciphertext
$$c := [m^e \bmod N].$$
- Dec: on input a private key $sk = \langle N, d\rangle$ and a ciphertext $c \in \mathbb{Z}_N^*$, compute the message
$$m := [c^d \bmod N].$$

The "textbook RSA" encryption scheme.

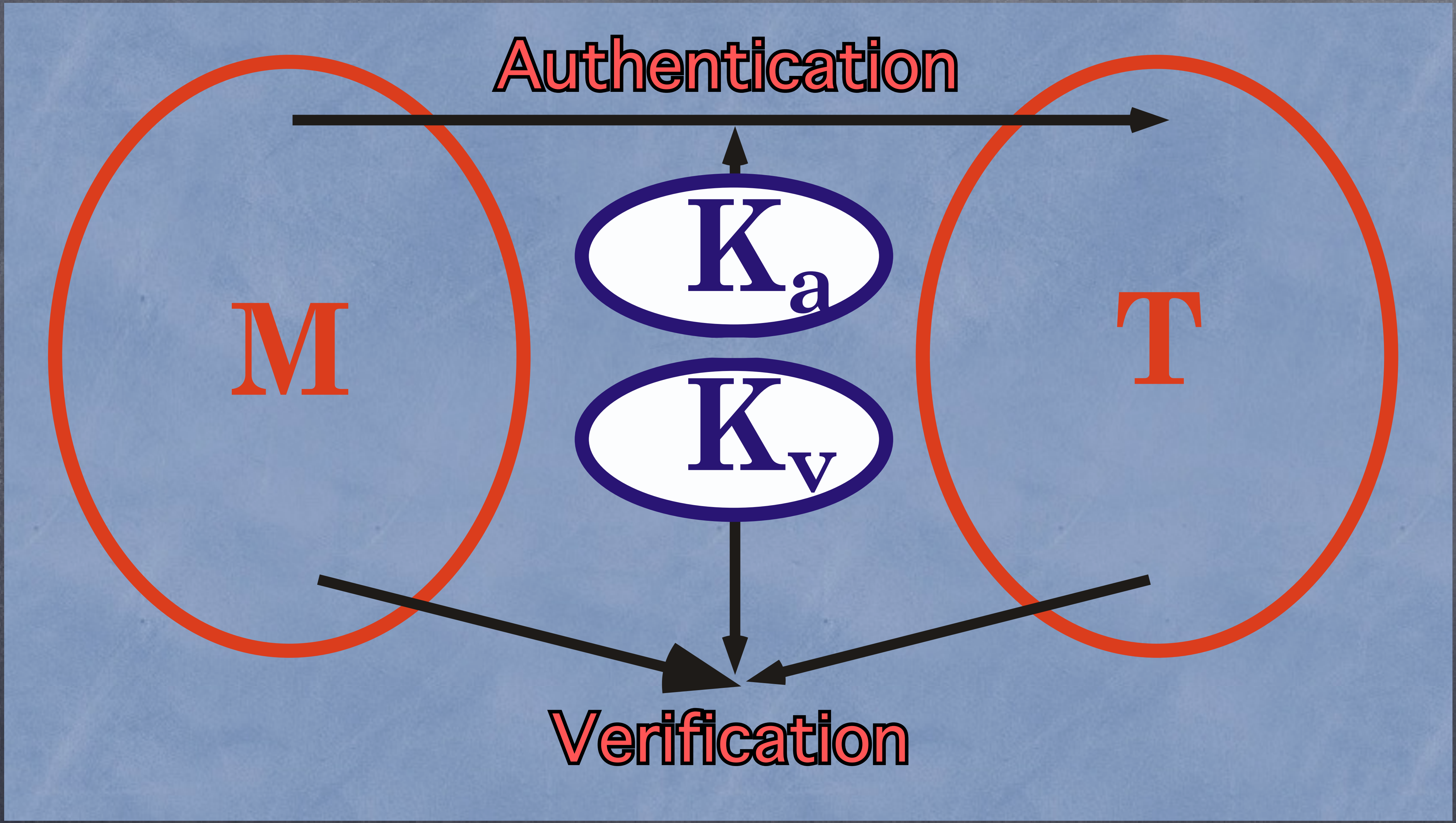
The RSA Assumption

- (Informal) Given a modulus $\boldsymbol{N}$, an exponent $\mathbf{e > 0}$ that is relatively prime to $\boldsymbol{\varphi(N)}$, and an element $\mathbf{y} \in \mathbb{Z}_N^*$, compute $\sqrt[e]{\mathbf{y}} \ \mathbf{mod}\ \boldsymbol{N}$;
- Given $\boldsymbol{N}$,$\mathbf{e}$,$\boldsymbol{y}$, finding $\boldsymbol{x}$ such that $\boldsymbol{x}^{\mathbf{e}} = \mathbf{y \ mod}\ \boldsymbol{N}$ is hard; the success probability of any polynomial-time algorithm is negligeable.
- However, finding such an $\boldsymbol{x}$ is easy given $\boldsymbol{p}$ and $\boldsymbol{q}$ such that $\boldsymbol{N=pq}$: an exponent $\boldsymbol{d}$ can be easily computed so that $\boldsymbol{x} = \sqrt[e]{\mathbf{y}}\ \mathbf{mod}\ \boldsymbol{N} = \mathbf{y}^{\boldsymbol{d}}\ \mathbf{mod}\ \boldsymbol{N}$.

Digital Signatures

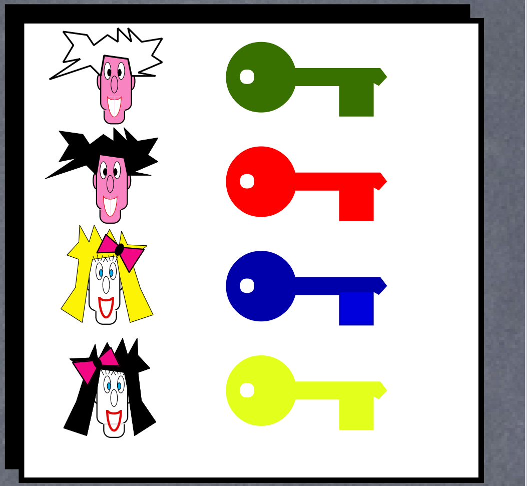
Asymmetric Authentication

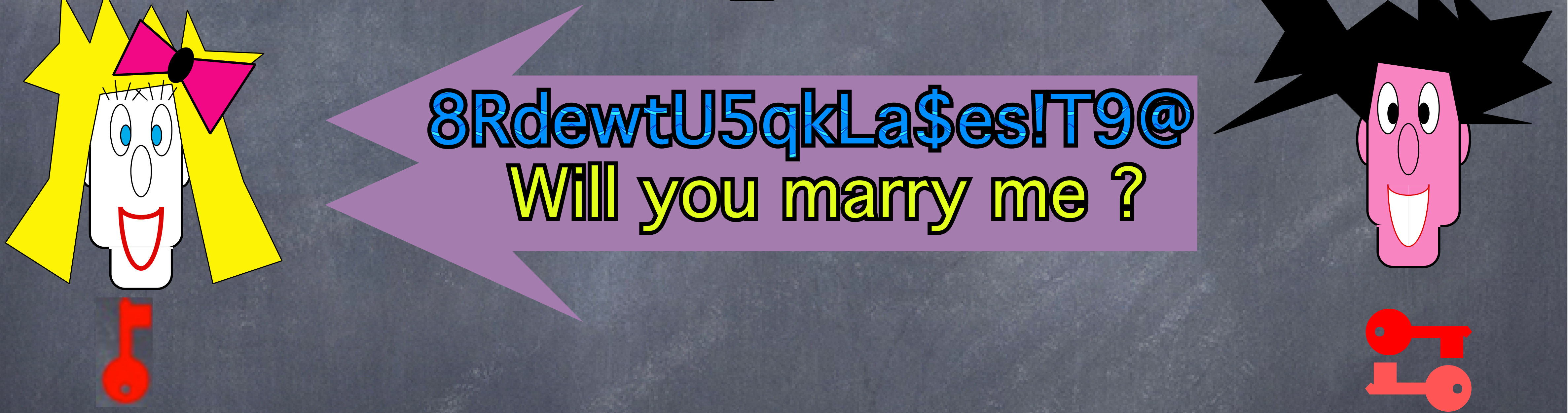
(Digital Signature Scheme)



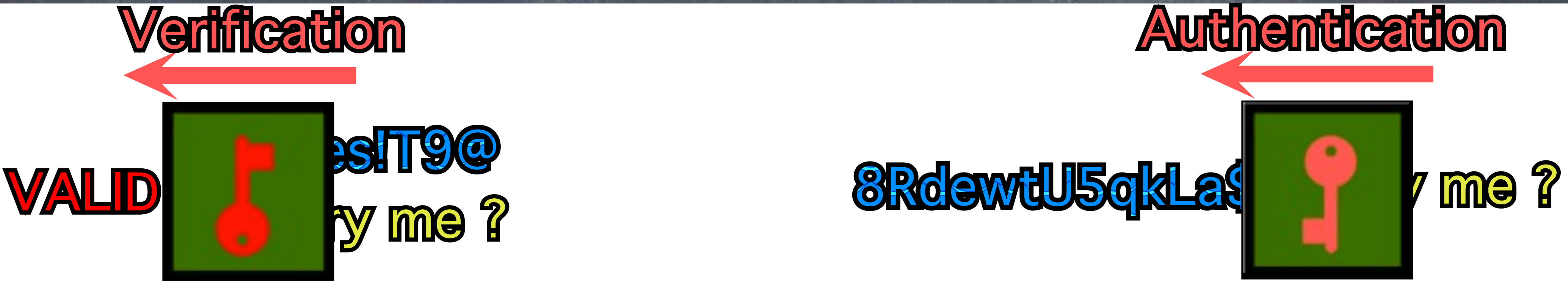
Complexity Theoretical Security

Digital Signature





8RdewtU5qkLa$es!T9@
Will you marry me ?



Verification

VALID
es!T9@
ry me ?

Authentication

8RdewtU5qkLa$
y me ?

COMP-330A
Probabilistic Computations and Cryptography

Prof. Claude Crépeau

School of Computer Science
McGill University