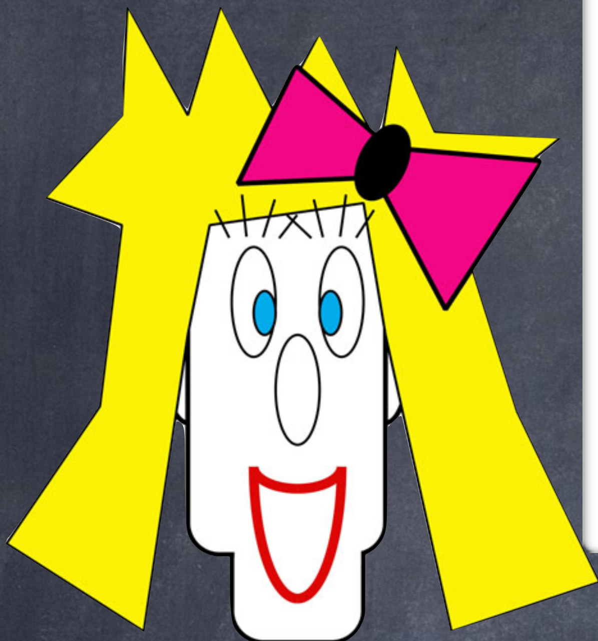
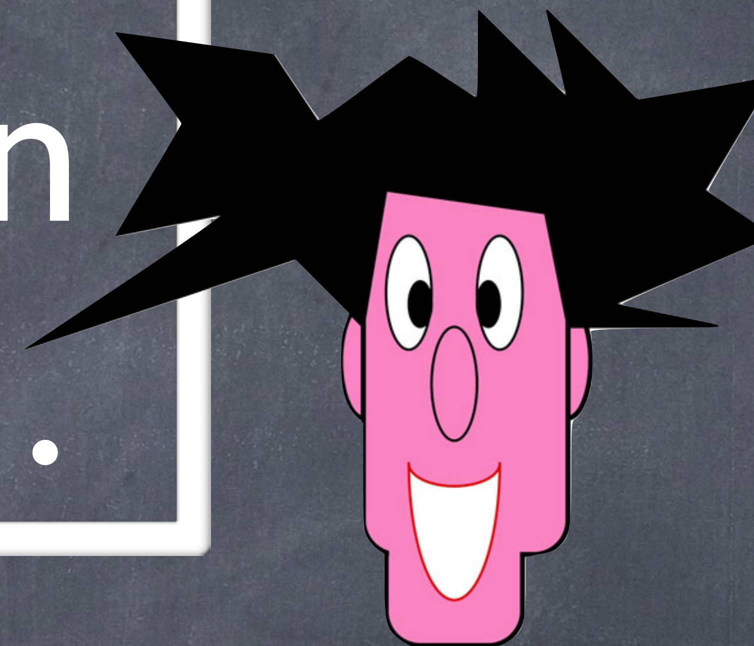
Winter 2016
COMP-250: Introduction to Computer Science

Lecture 14, February 25, 2016

Alice and Bob's Adventures in Cryptoland...



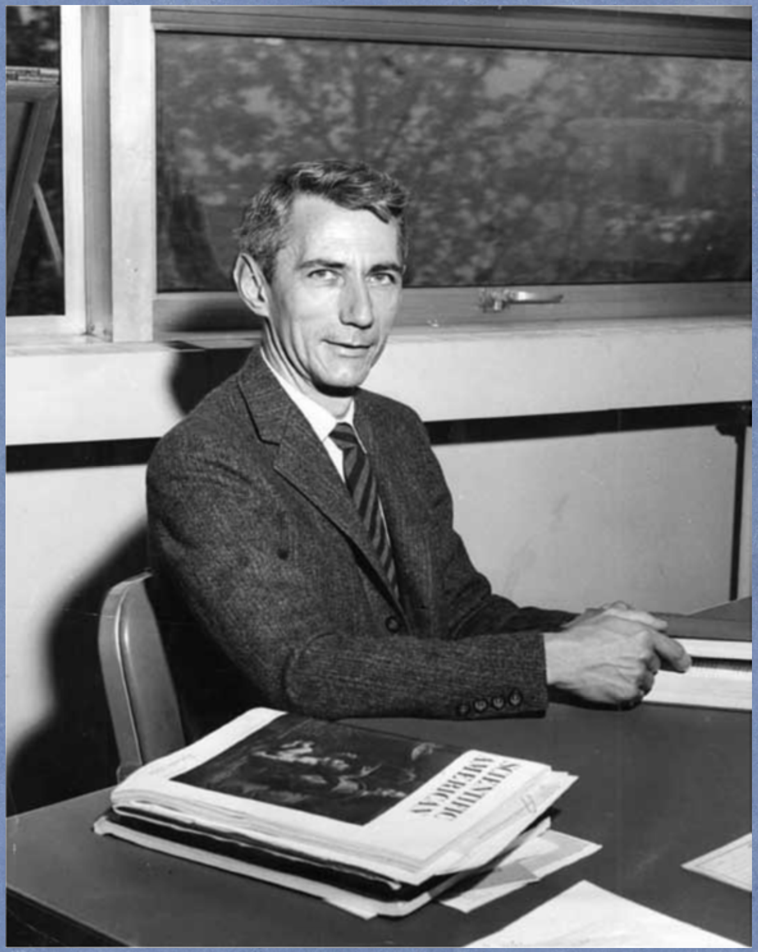
Alice



Bob

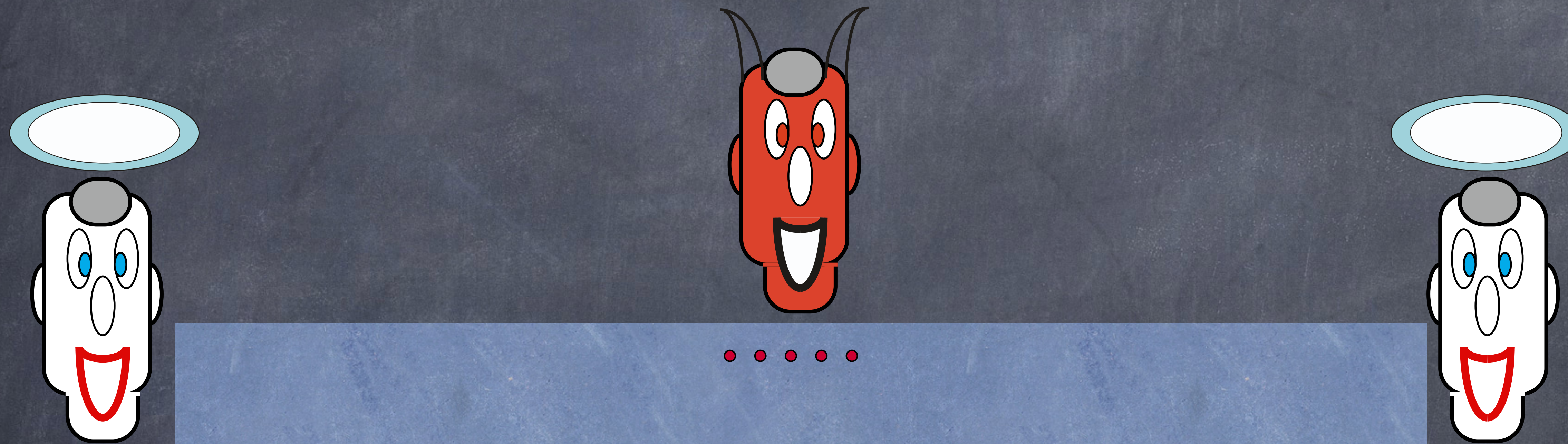
Information

Theoretical



Cryptography

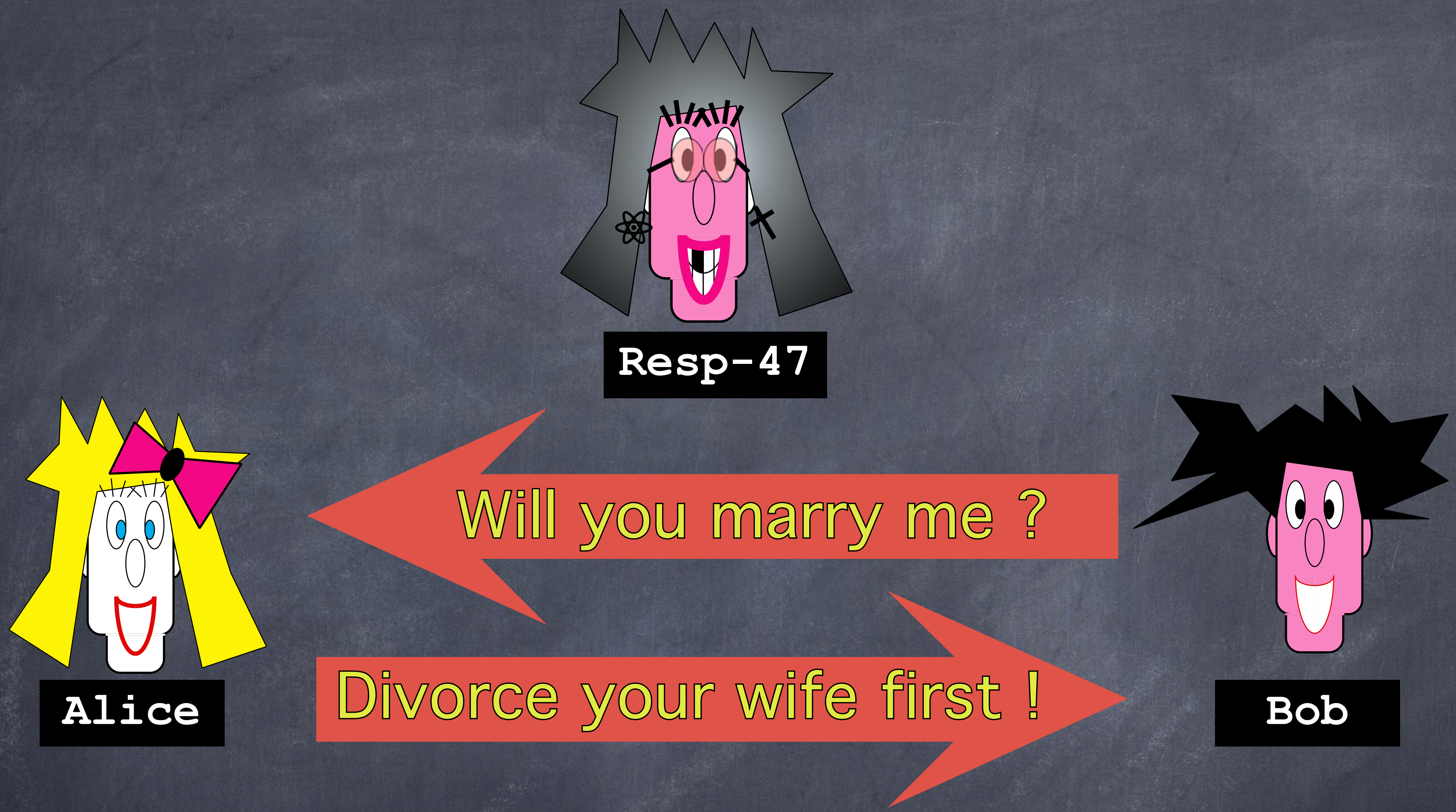
Information Theoretical Cryptography

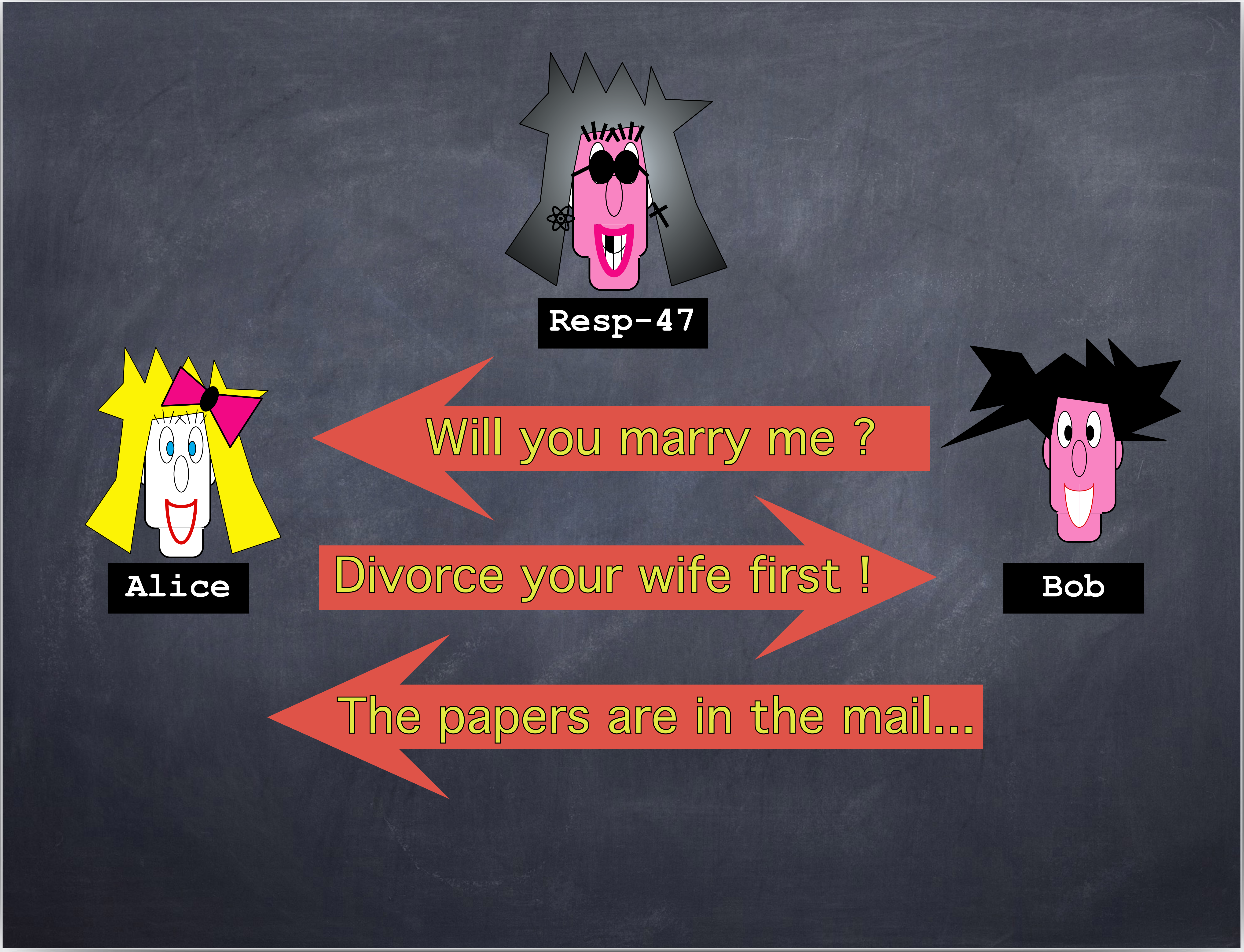


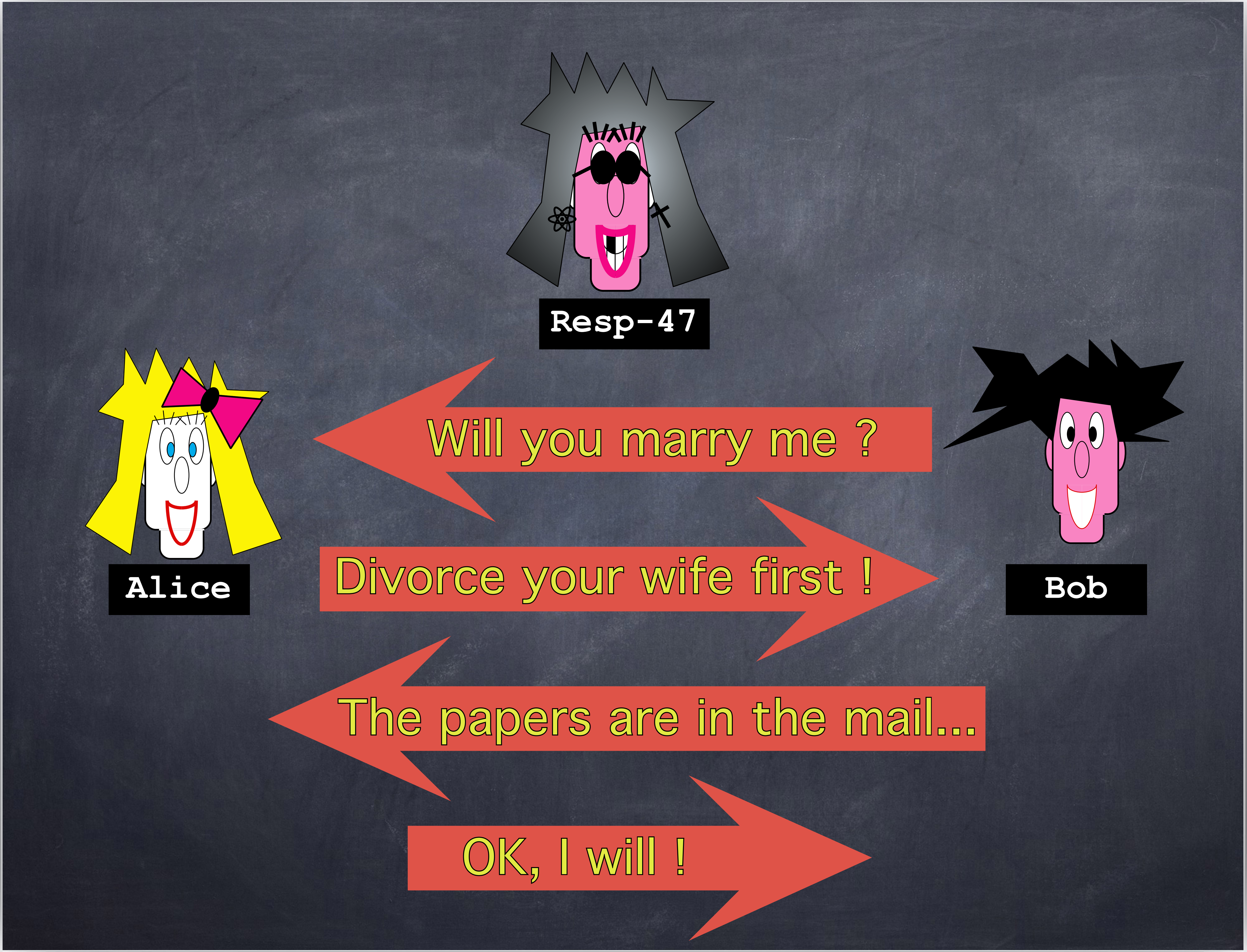
Key Distribution

Encryption

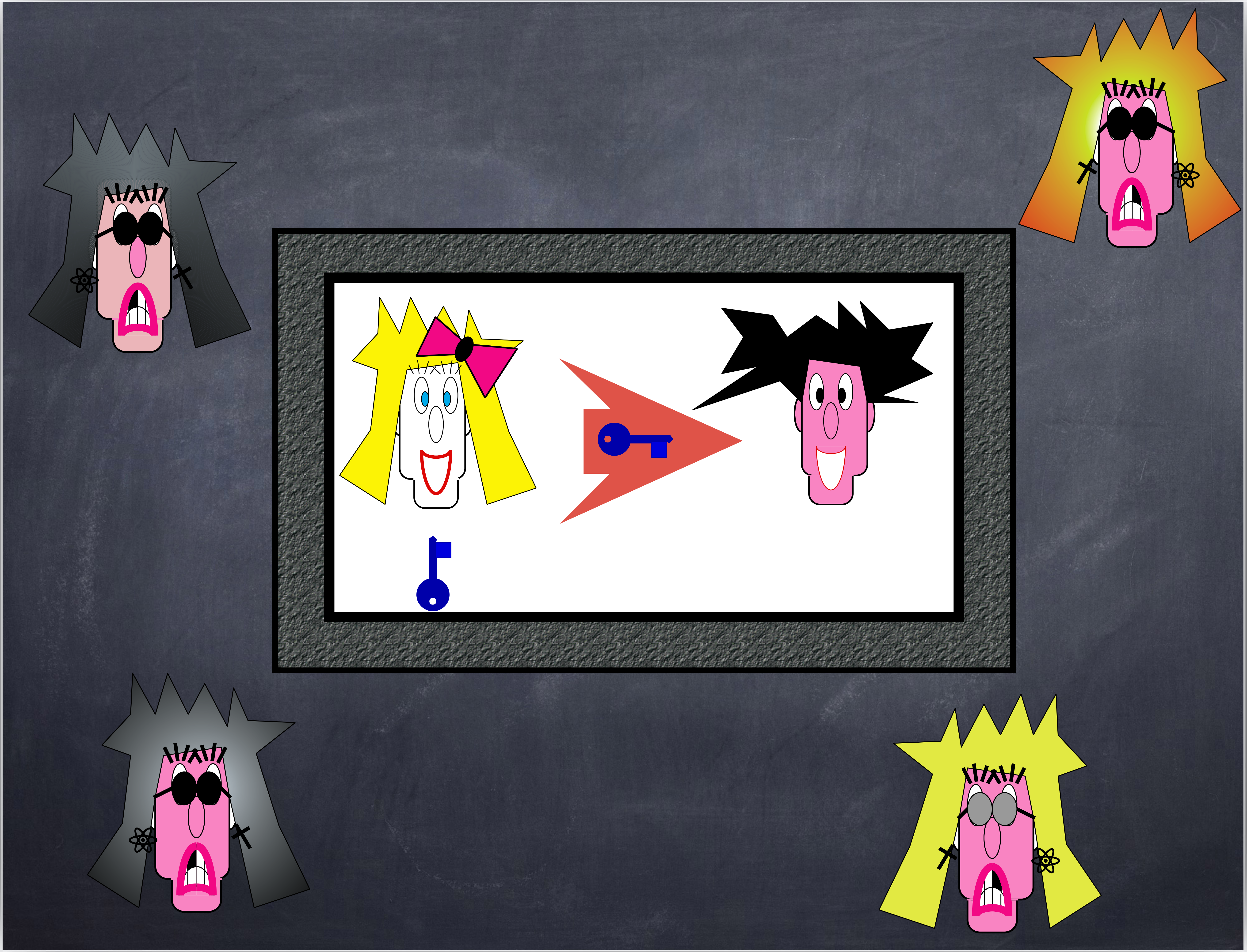

Resp-47
Will you marry me ?
Bob


Resp-47
Alice
Will you marry me ?
Divorce your wife first !
Bob


Resp-47
Alice
Bob
Will you marry me ?
Divorce your wife first !
The papers are in the mail...

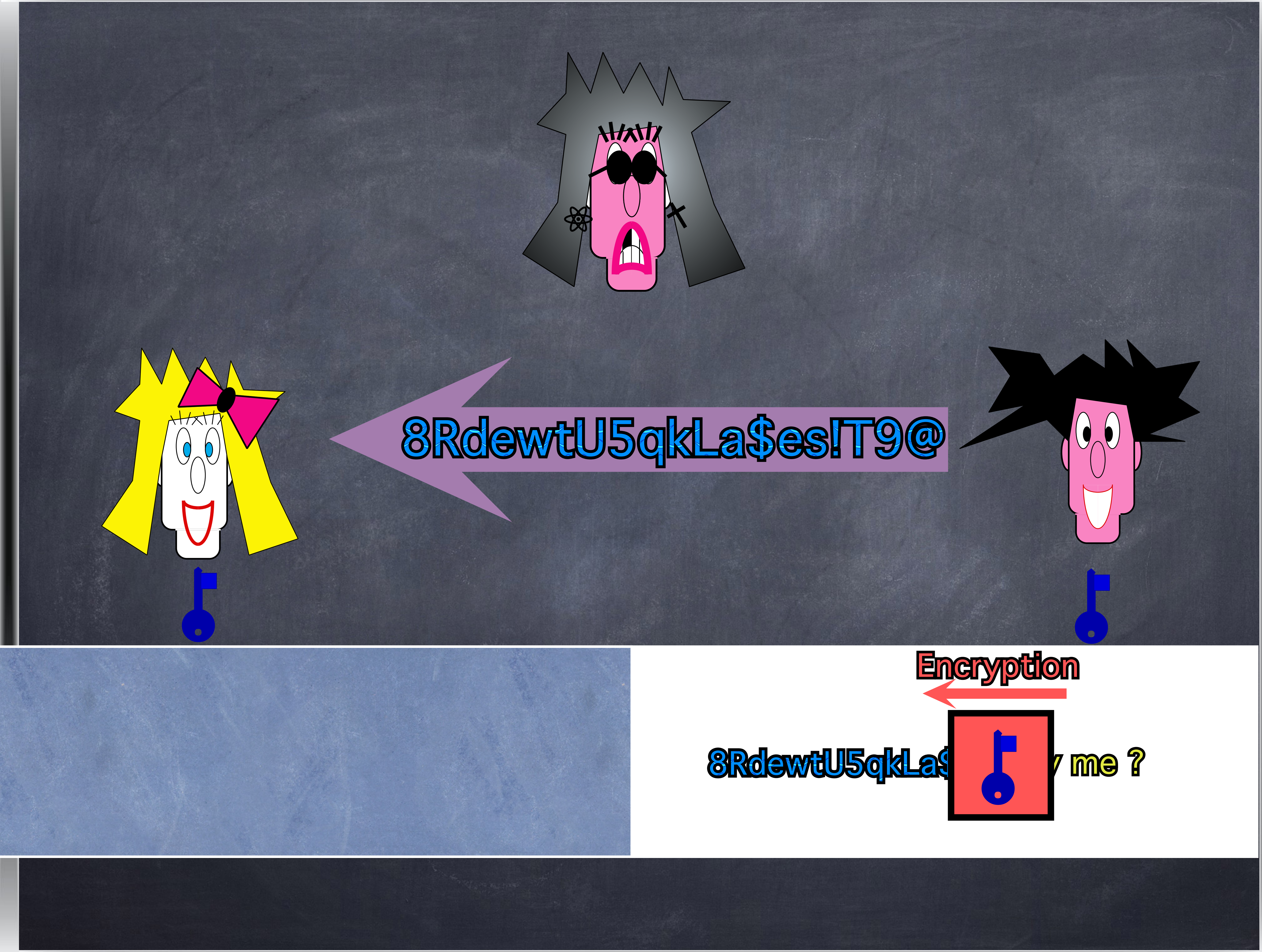

Resp-47
Alice
Bob
Will you marry me ?
Divorce your wife first !
The papers are in the mail...
OK, I will !

Key Distribution



Encryption


8RdewtU5qkLa$es!T9@

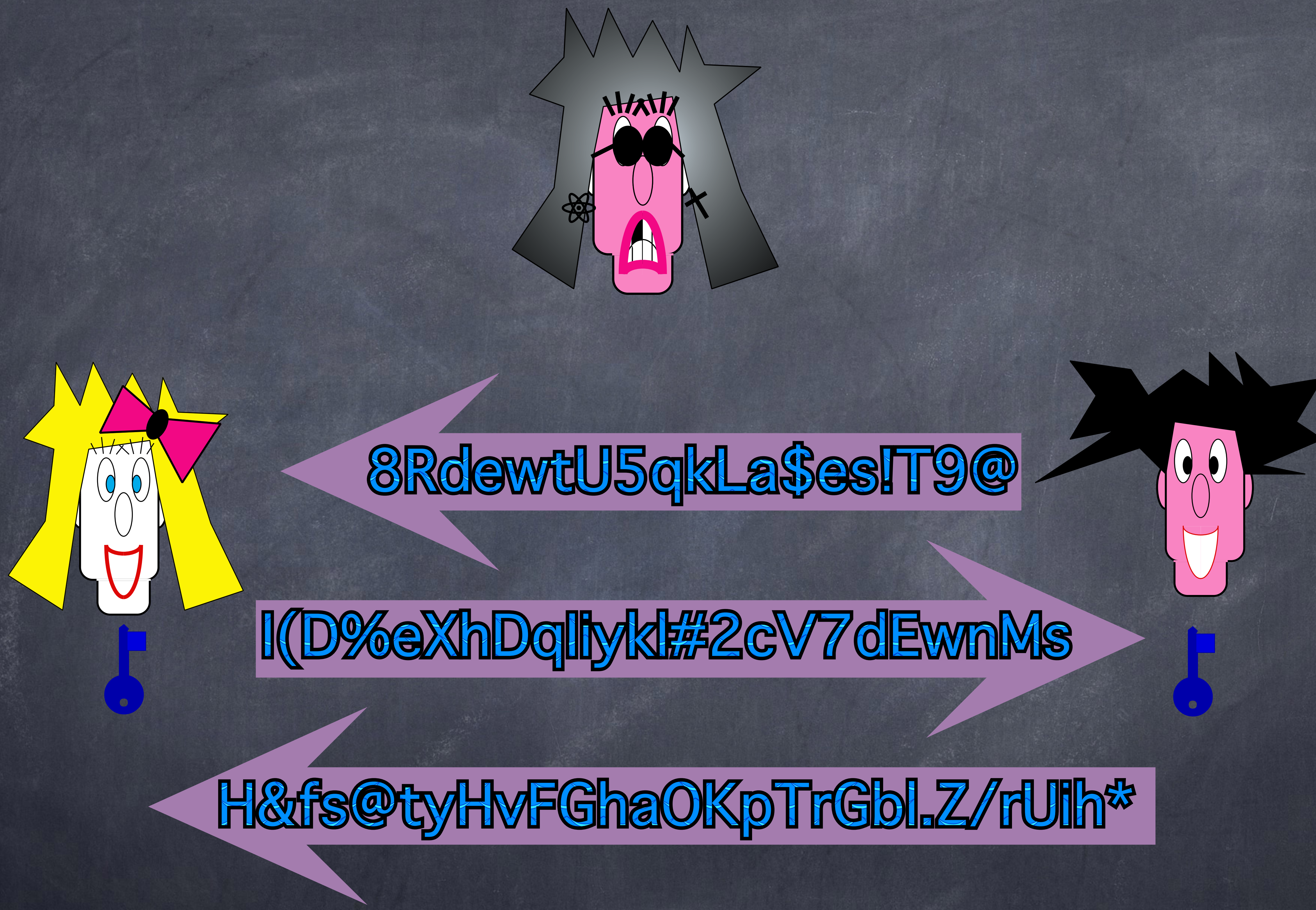

8RdewtU5qkLa$es!T9@
Encryption
8RdewtU5qkLa$
me ?


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

8RdewtU5qkLa$es!T9@
I(D%eXhDqliykl#2cV7dEwnMs

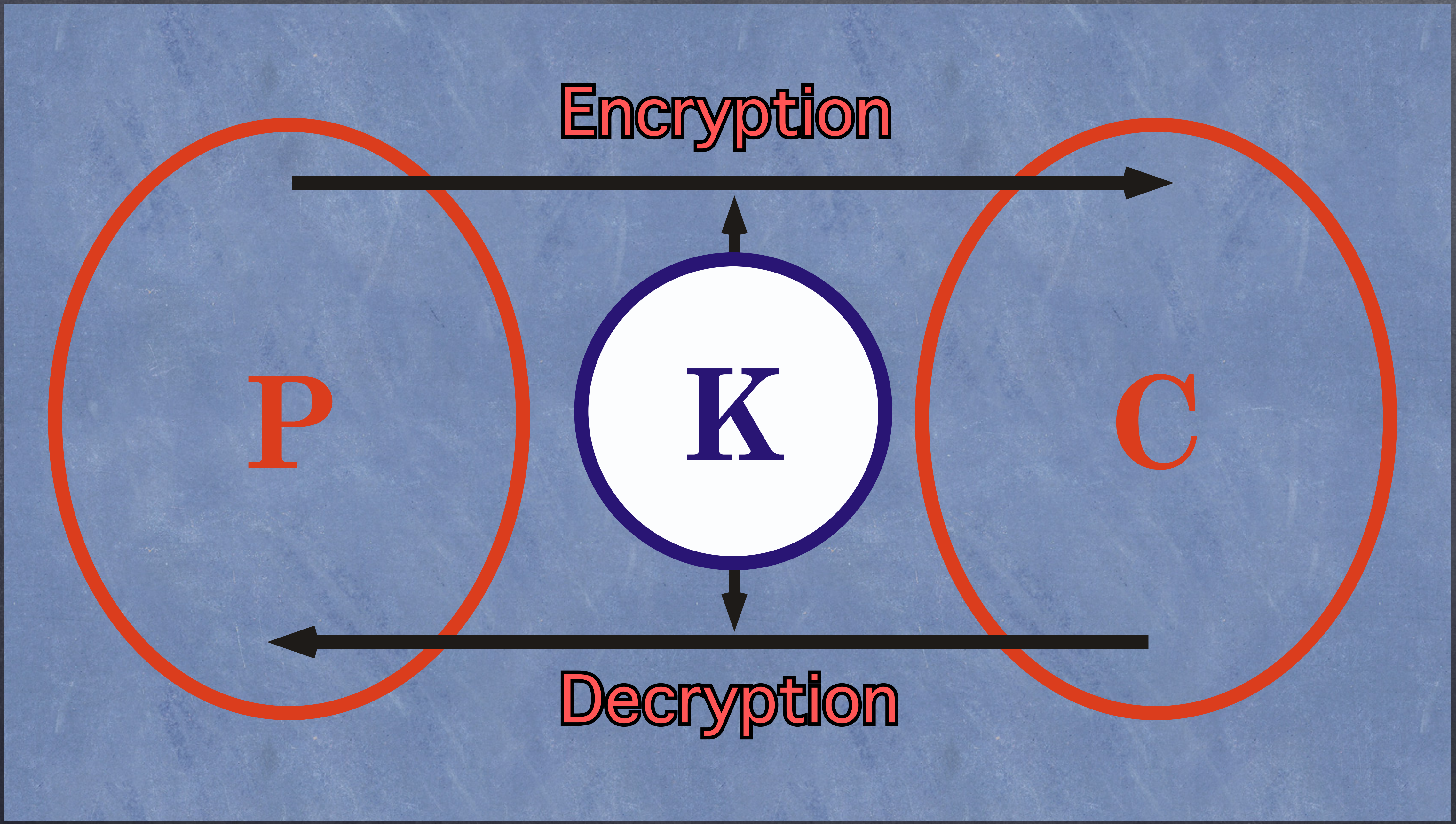

8RdewtU5qkLa$es!T9@
I(D%eXhDqliykl#2cV7dEwnMs
Encryption
Divorce your wif
2cV7dEwnMs

8RdewtU5qkLa$es!T9@
I(D%eXhDqIiykl#2cV7dEwnMs
Encryption
Decryption
Divorce your wif
2cV7dEwnMs
I(D%eXhDqIiykl#2cV7
ur wife first !


8RdewtU5qkLa$es!T9@
I(D%eXhDqliykl#2cV7dEwnMs
H&fs@tyHvFGhaOKpTrGbl.Z/rUih*

8RdewtU5qkLa$es!T9@
I(D%eXhDqliykl#2cV7dEwnMs
H&fs@tyHvFGhaOKpTrGbl.Z/rUih*
B7B3tdsjUila

Symmetric Encryption

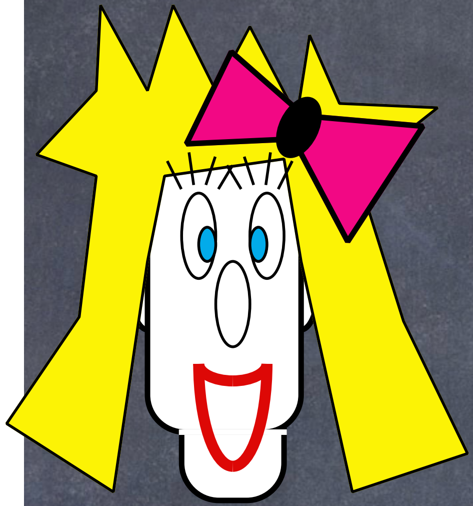


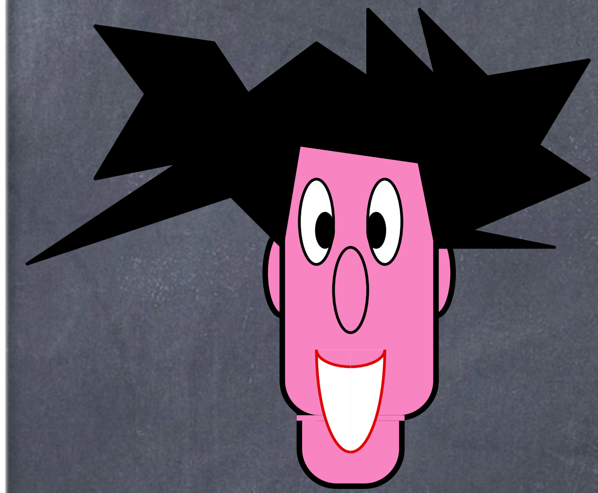
Information Theoretical Security

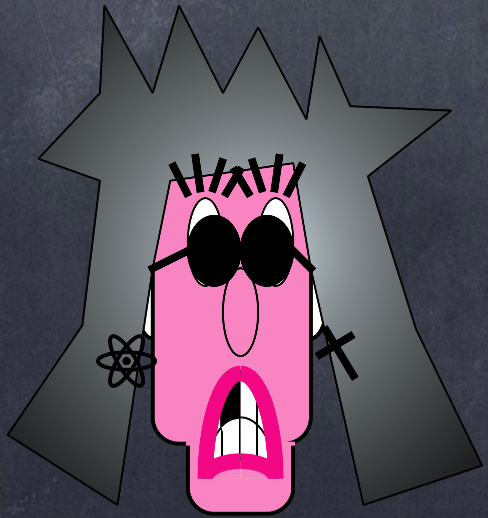
Symmetric Encryption

Ceasar's Cipher

VERNAM's Cipher





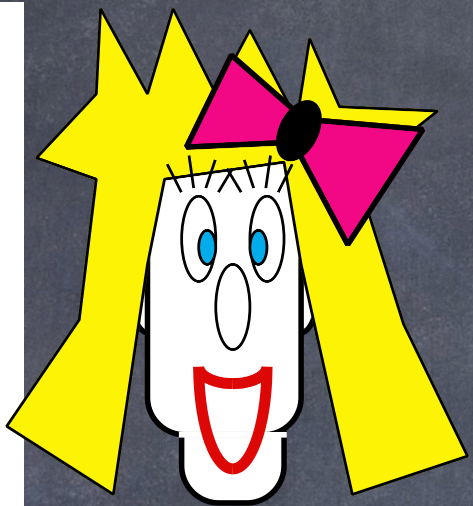


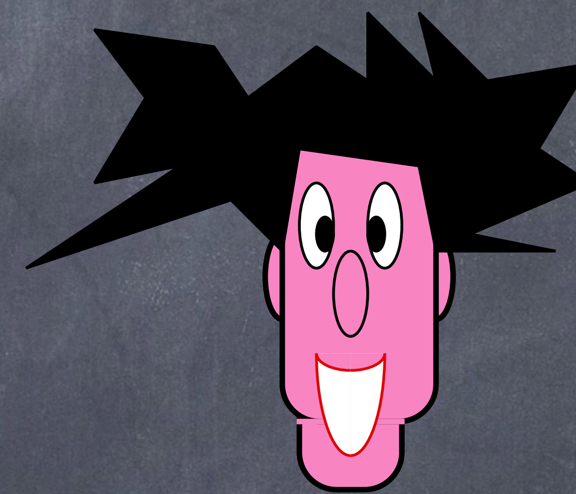
m
1
0
1
0
0
1
0
0
1
1
1
1
1
0
0
1

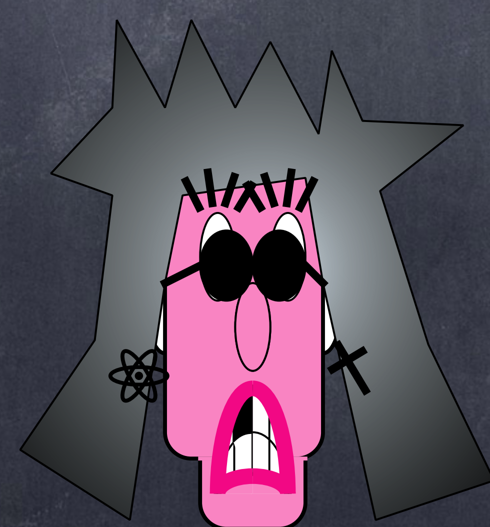
1910's

VERNAM's Cipher

$m \oplus k$	
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



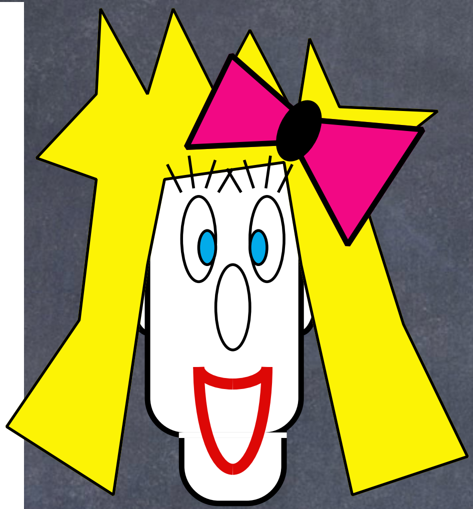


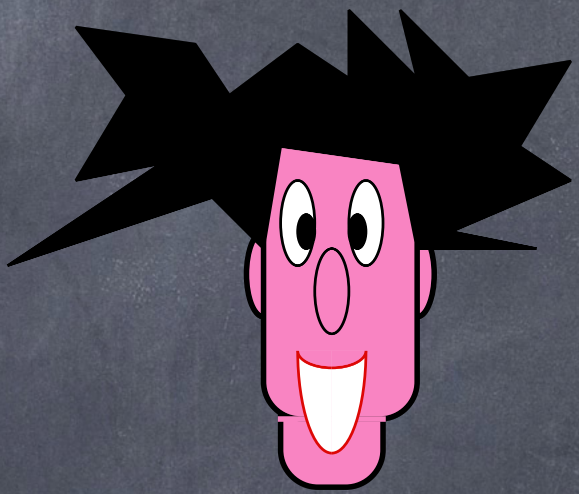


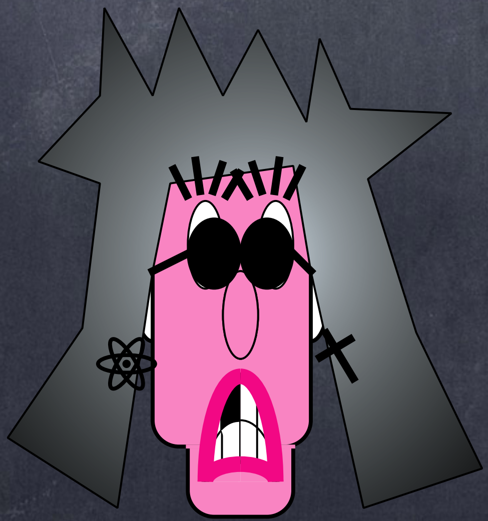
VERNAM's Cipher

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

$\oplus \; =$

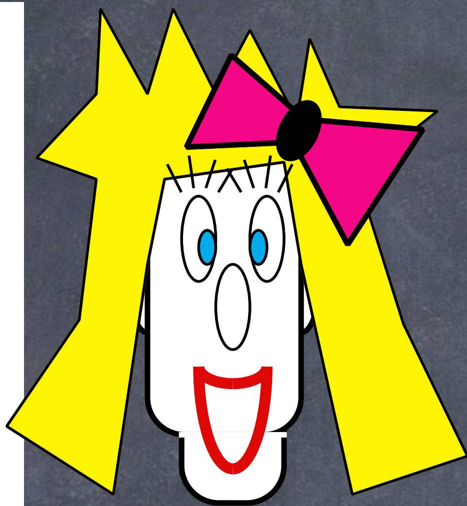


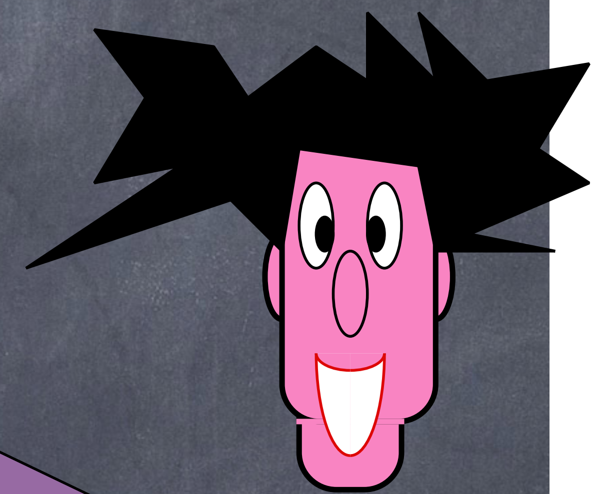




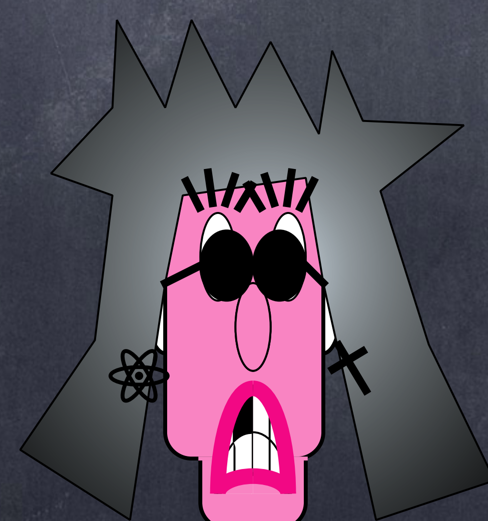
VERNAM's Cipher

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



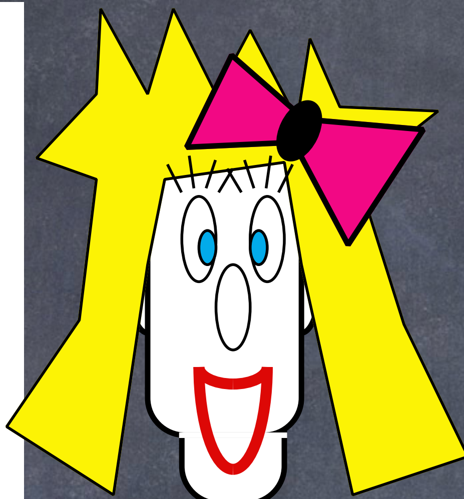


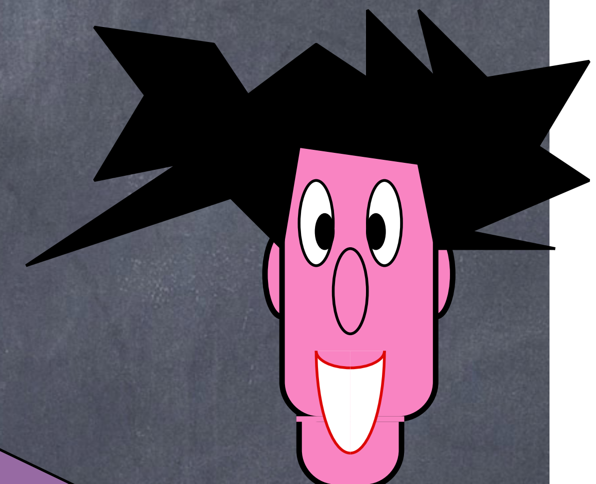
c
0
1
0
0
0
0
1
0
0
0
1
0
1
1
1
0



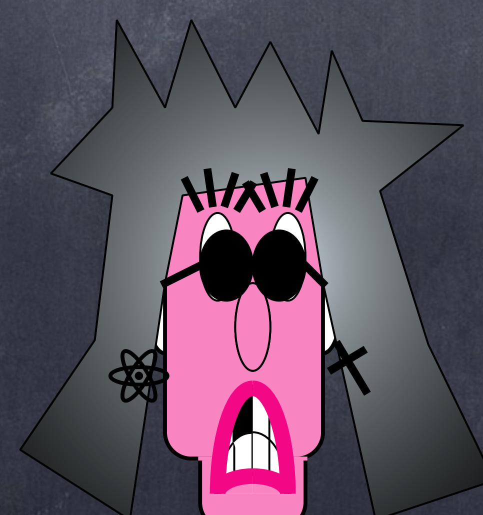
VERNAM's Cipher

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	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



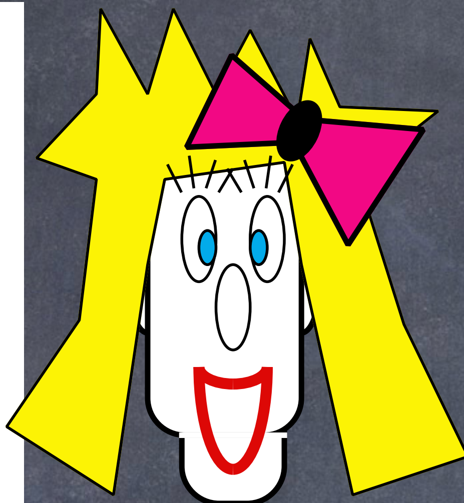


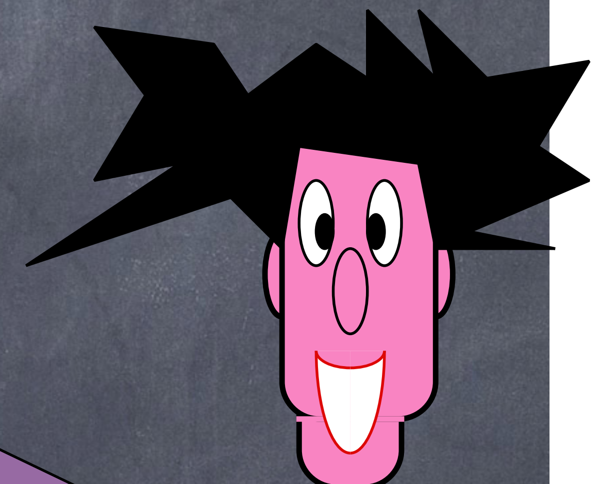
c	$\oplus$ k
0	1
1	1
0	1
0	0
0	0
0	1
1	1
0	0
0	1
0	1
1	0
0	1
1	0
1	1
1	1
0	1



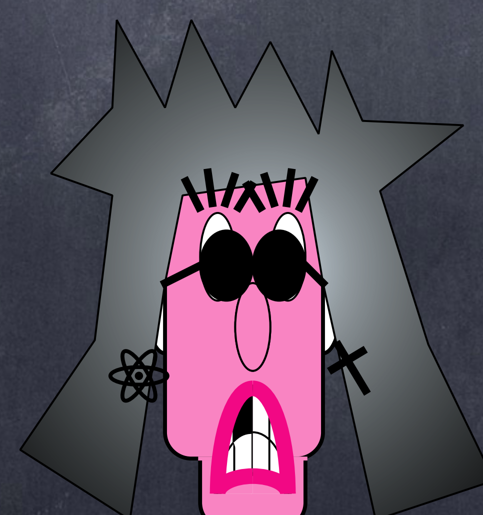
VERNAM's Cipher

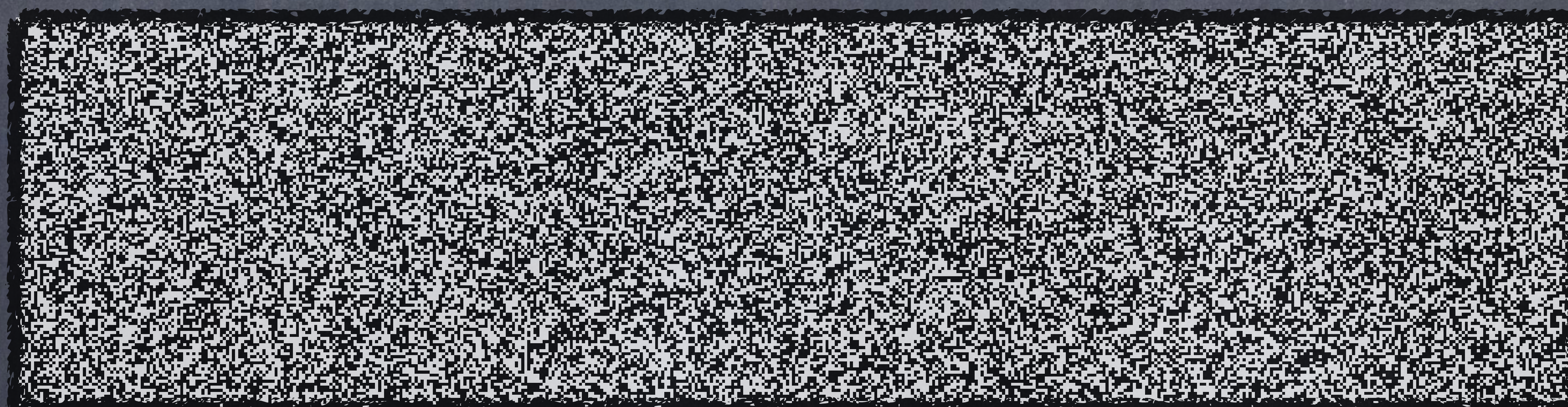
m	$\oplus k$	$= c$
1	1	0
0	1	1
1	1	0
0	0	0
0	0	0
1	1	0
0	1	1
0	0	0
1	1	0
1	1	0
1	0	1
1	1	0
1	0	1
0	1	1
0	1	1
1	1	0





c	$\oplus k$	$= m$
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





[redacted]

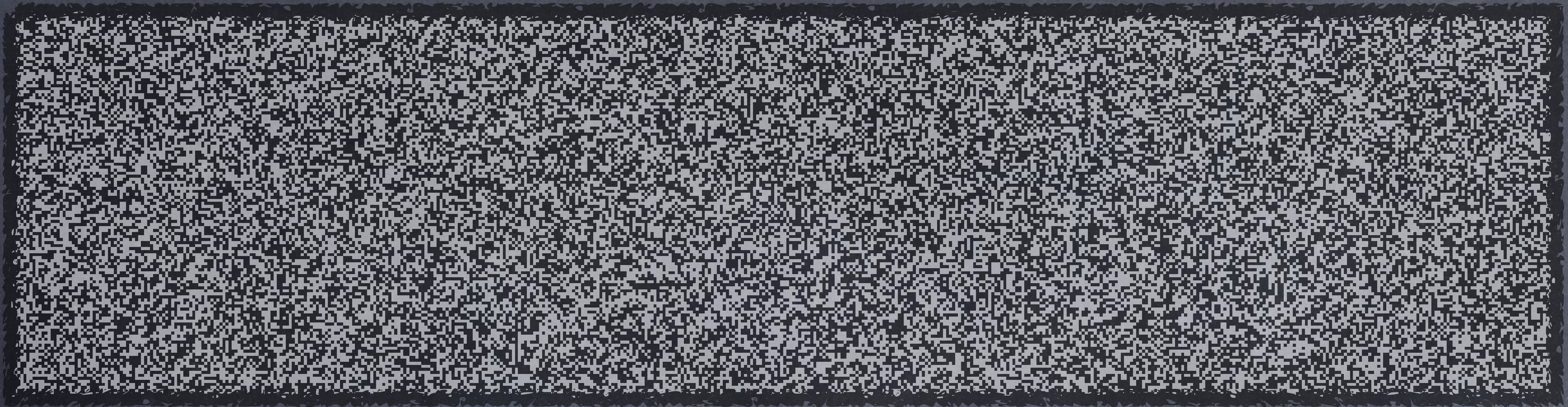
[redacted]

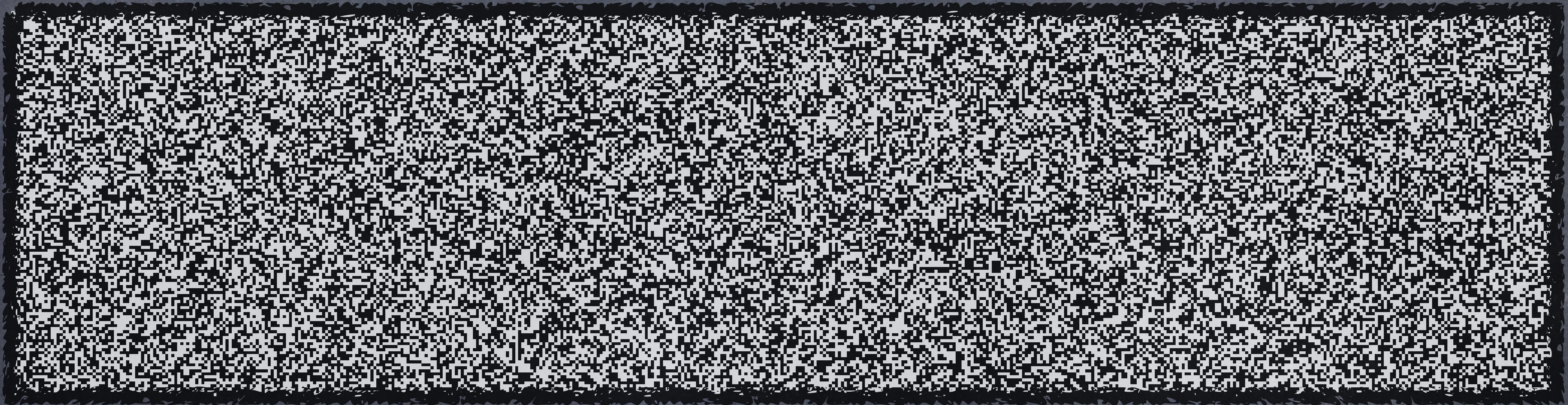
VERNAM

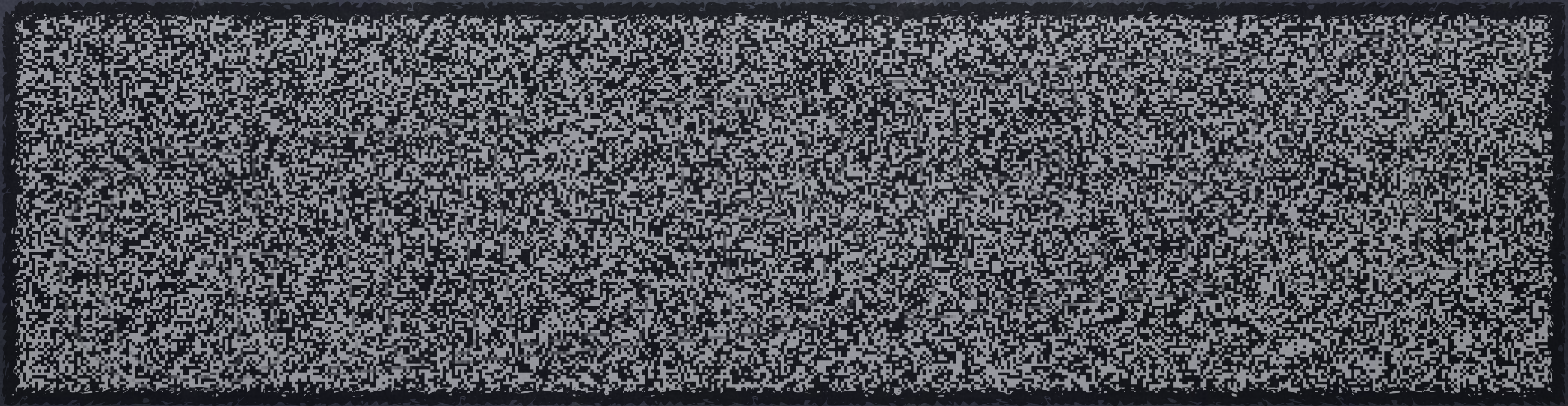
[redacted]

[redacted]

GILBERT







[redacted]

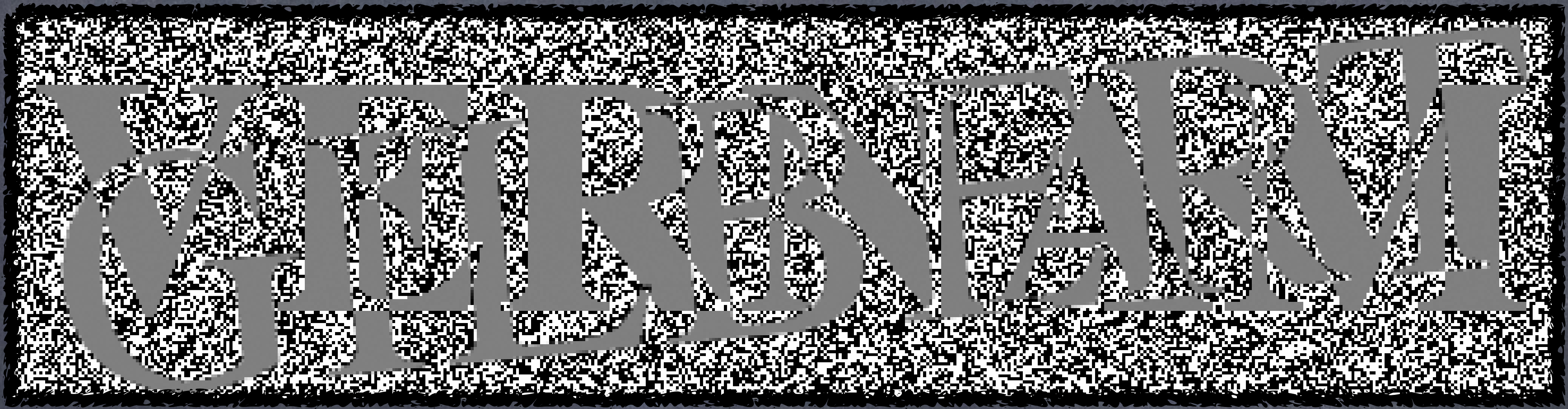
[redacted]

[redacted]

[redacted]

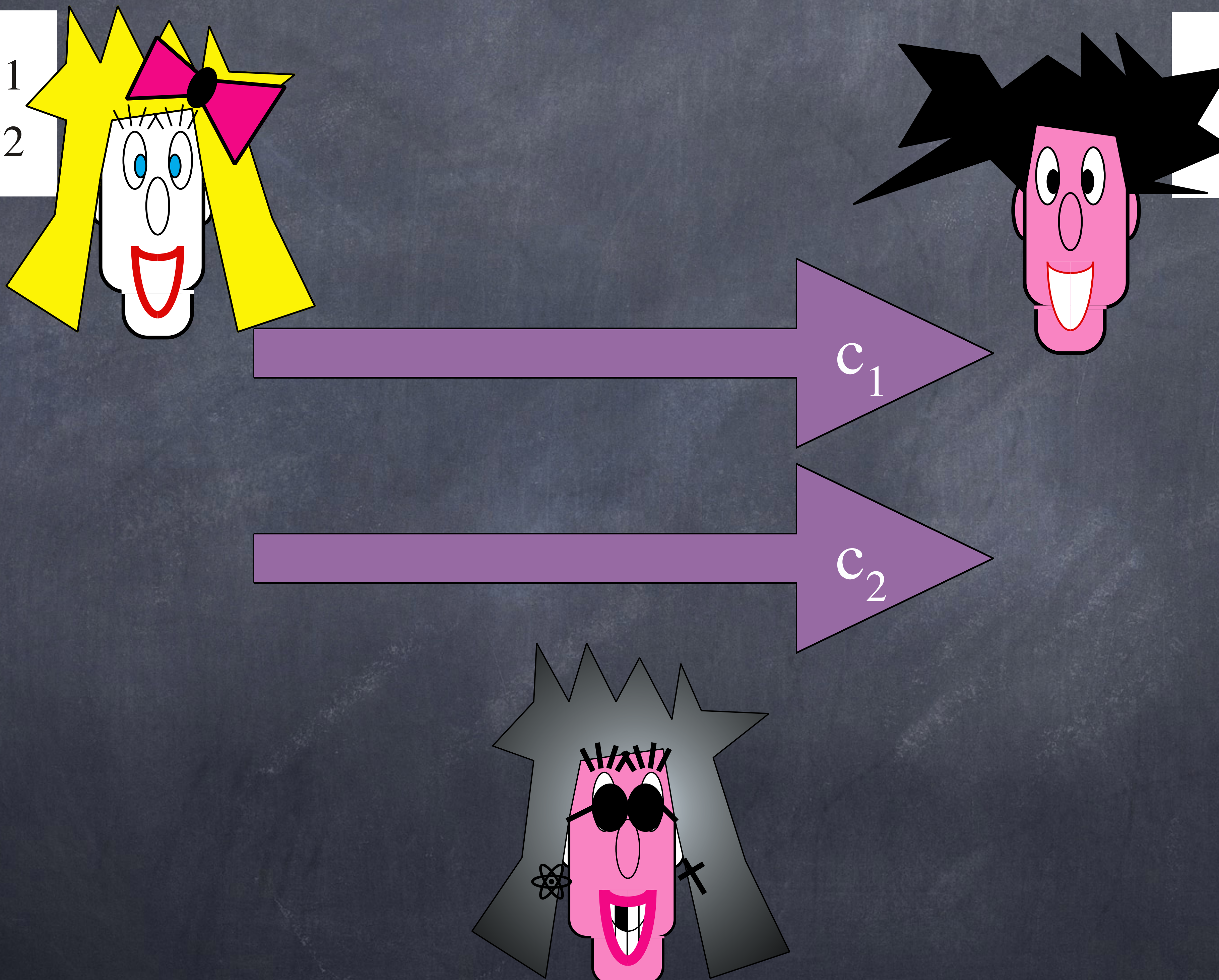
[redacted]

[redacted]



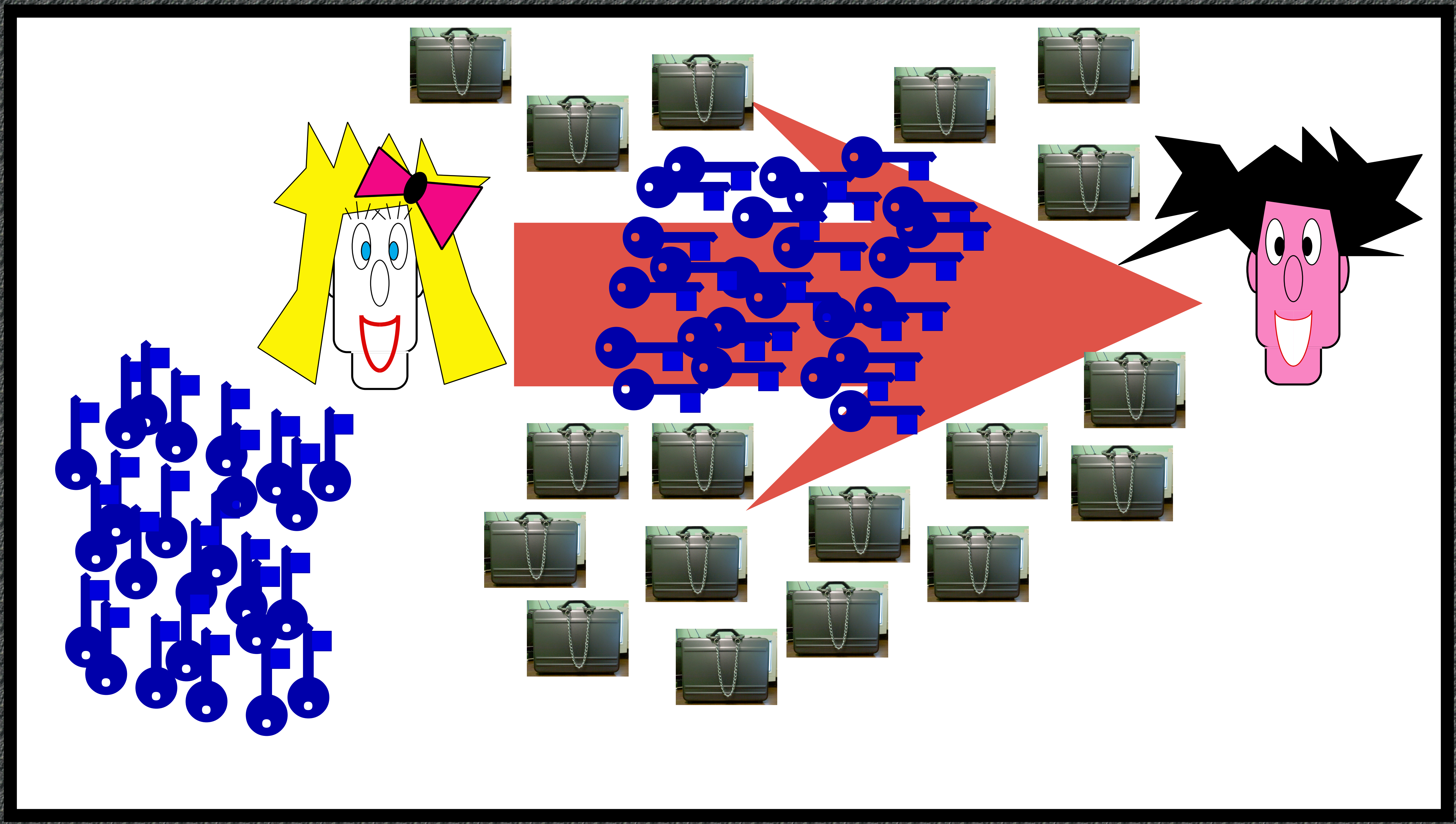
VERNAM's One-Time Pad

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



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

$c_1 \oplus c_2 = m_1 \oplus m_2$

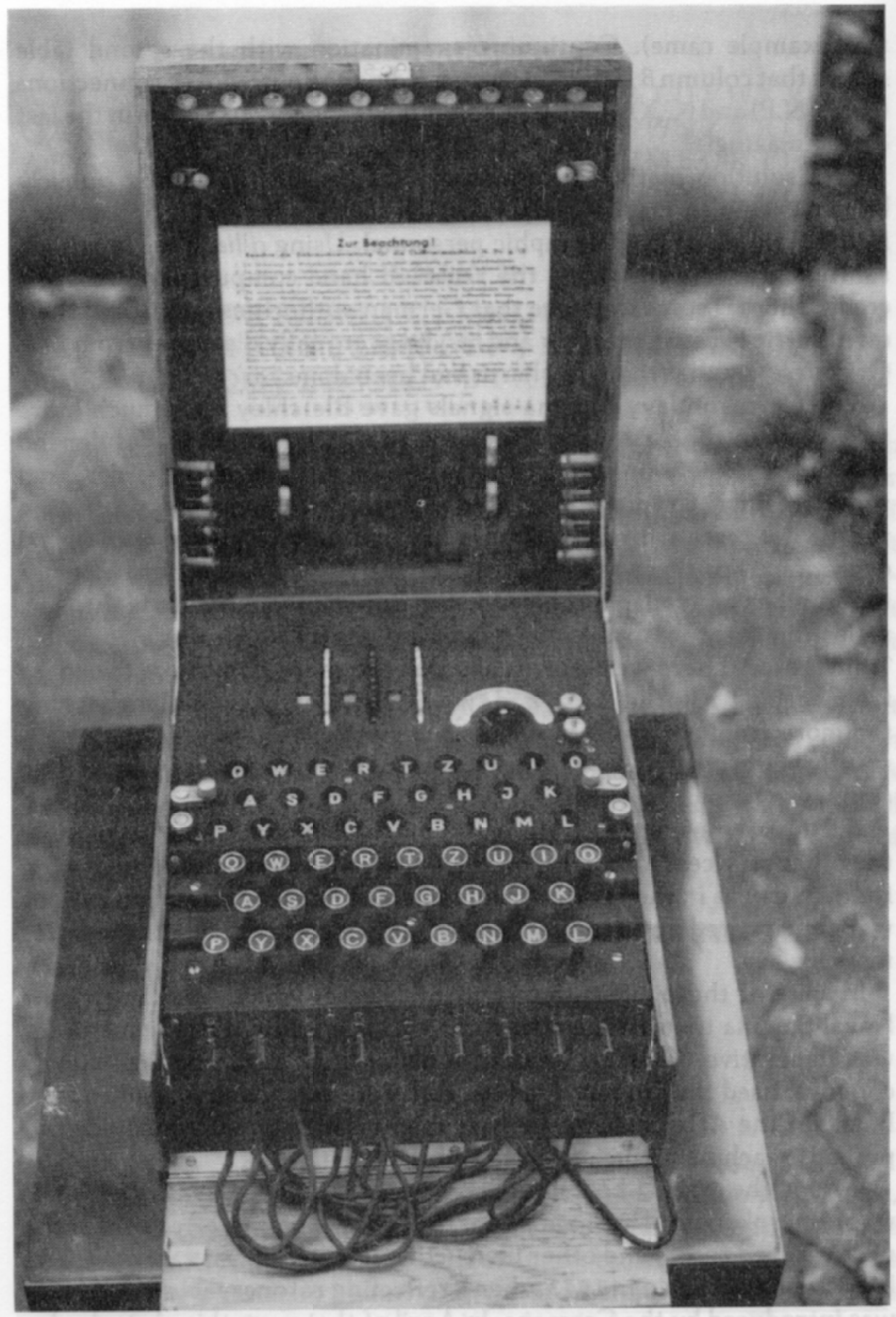


Complexity

Theoretical

Cryptography

The Enigma Machine

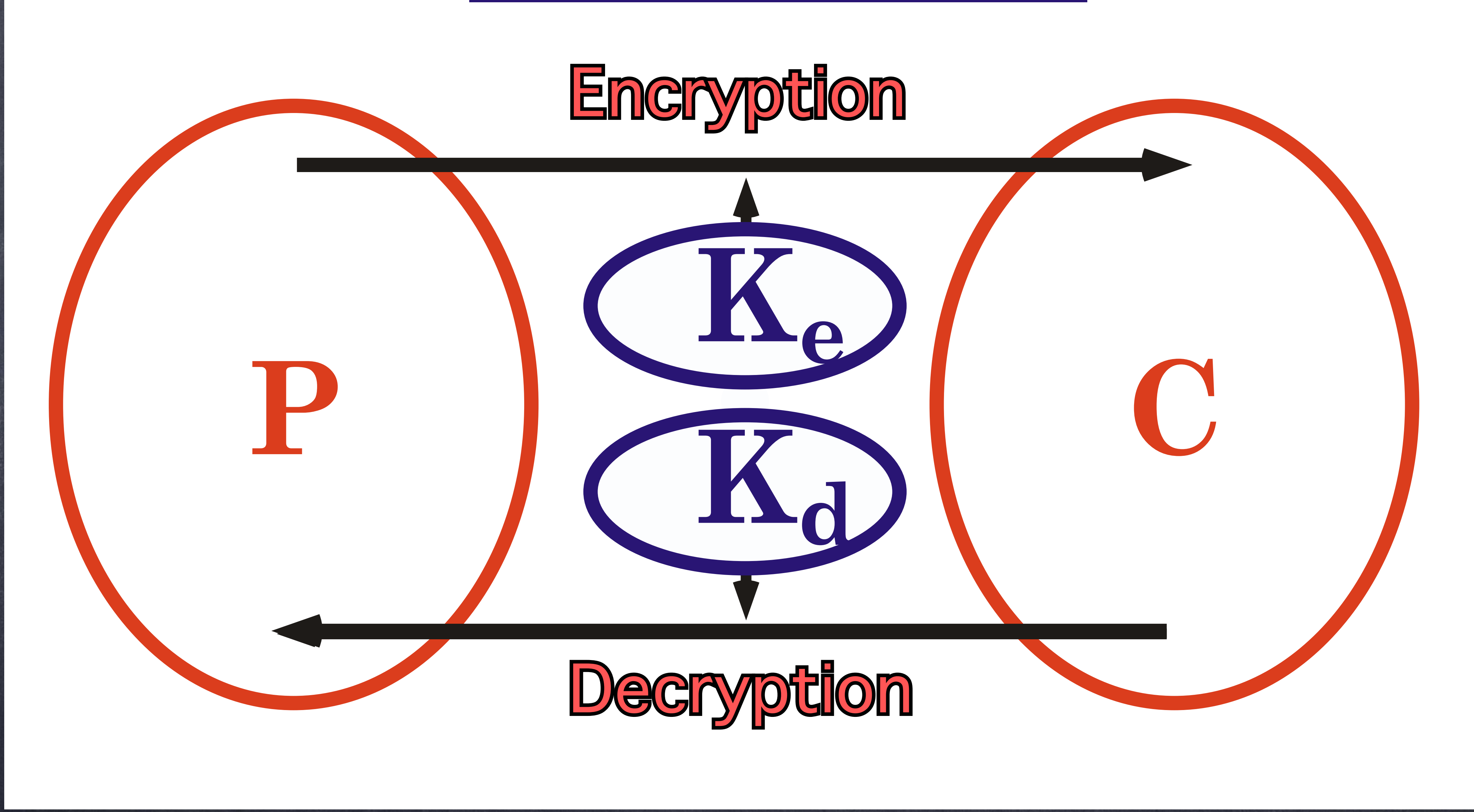


GERMAN ARMY MILITARY ENIGMA. THIS MODEL WAS THE MOST WIDELY USED VERSION OF THE GERMAN WAR-TIME ENIGMAS.

Arthur Scherbius

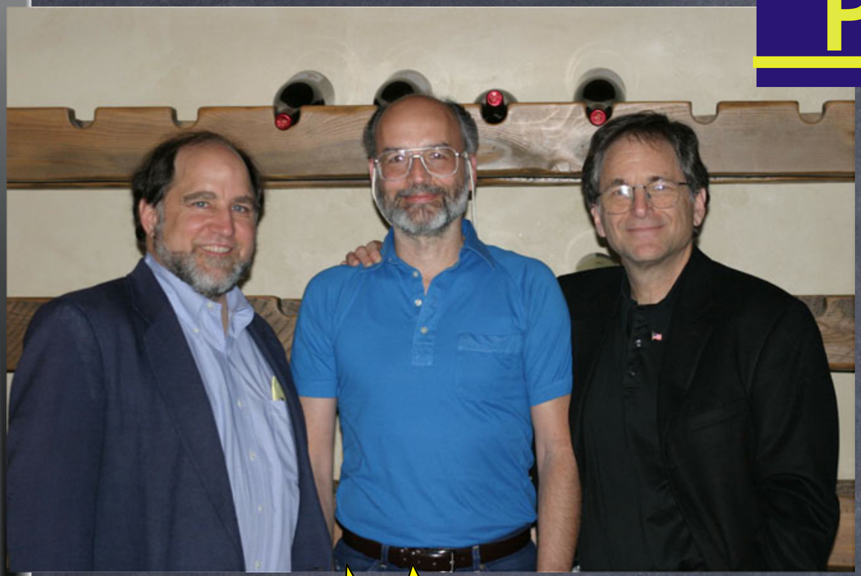
1930's

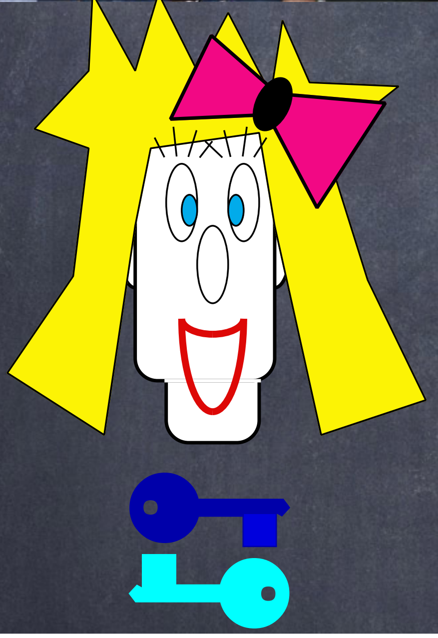
Asymmetric Encryption
(Public-Key Cryptography)



Complexity Theoretical Security

Public-Key Cryptography





Public-Key Cryptography

Public-Key Cryptography

Fast Exponentiation

- **Input:** base x and exponent e.
- **Output:** x^e. // *** warning $|x^e| \sim e|x|$ ***

$y = 1$
WHILE e>0 **DO**
  IF e%2 = 1 **THEN** $y = xy$
  e = e/2; $x = x^2$
return y

- running time is $O(e|x|^2) = O(2^{|e|}|x|^2)$

Fast Modular Exponentiation

- **Input:** base x ,modulus N and exponent e.
- **Output:** $x^e \% N$.

y = 1
WHILE e>0 **DO**
  IF $e\%2 = 1$ **THEN** $y = xy \% N$
  e = e/2; $x = x^2 \% N$
return y

- running time is $O(|e|*|x|^2) = O(|x|^3)$

Example

Example

$5^{13} \bmod 7$

Example

$5^{13} \bmod 7$

$= 5^{8+4+1} \bmod 7$

Example

$5^{13} \bmod 7$

$= 5^{8+4+1} \bmod 7$

$= 5^8 \times 5^4 \times 5^1 \bmod 7$

Example

$5^{13} \bmod 7$

$= 5^{8+4+1} \bmod 7$

$= 5^8 \times 5^4 \times 5^1 \bmod 7$

$5^0, 5^1, 5^2, 5^4, 5^8$

Example

$5^{13} \bmod 7$

$= 5^{8+4+1} \bmod 7$

$= 5^8 \times 5^4 \times 5^1 \bmod 7$

$5^0, 5^1, 5^2, 5^4, 5^8$

$\equiv 1, \; 5, \; 4, \; 2, \; 4 \pmod 7$

Example

$5^{13} \bmod 7$

$= 5^{8+4+1} \bmod 7$

$= 5^8 \times 5^4 \times 5^1 \bmod 7$

$5^0, 5^1, 5^2, 5^4, 5^8$

$\equiv 1, \ 5, \ 4, \ 2, \ 4 \pmod 7$

$= 4 \times 2 \times 5 \bmod 7$

Example

$5^{13} \bmod 7$

$= 5^{8+4+1} \bmod 7$

$= 5^8 \times 5^4 \times 5^1 \bmod 7$

$5^0, 5^1, 5^2, 5^4, 5^8$

$\equiv 1,\ 5,\ 4,\ 2,\ 4 \pmod 7$

$= 4 \times 2 \times 5 \bmod 7$

$= 1 \times 5 \bmod 7$

Example

$5^{13} \bmod 7$

$= 5^{8+4+1} \bmod 7$

$= 5^8 \times 5^4 \times 5^1 \bmod 7$

$5^0,\ 5^1,\ 5^2,\ 5^4,\ 5^8$

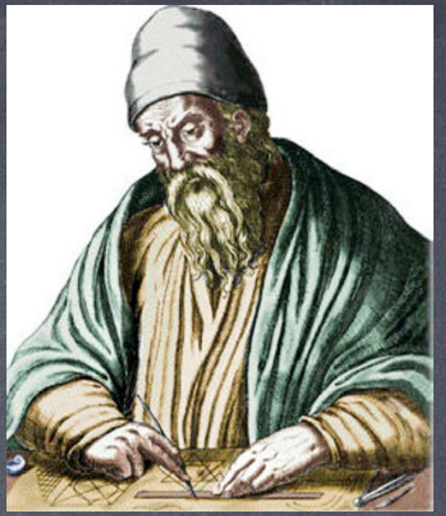
$\equiv 1,\ \ 5,\ \ 4,\ \ 2,\ \ 4 \pmod{7}$

$= 4 \times 2 \times 5 \bmod 7$

$= 1 \times 5 \bmod 7$

$= 5$

Euclid's Algorithm



Note. $\gcd(a,b) = g \rightarrow \exists_{x,y} \in \mathbb{Z}$ such that $g = ax + by$. The following recursive definition is based on the property $\gcd(a,b) = \gcd(a, b-a)$.

$$\gcd(a,b) = \begin{cases} a & \text{if } b = 0 \\ \gcd(b, a \bmod b) & \text{otherwise} \end{cases}$$

Euclidian Algorithm

- **Input**: integers a,b.
- **Output**: g,x,y such that g=GCD(a,b).

  $g = a$; $g' = b$;
  WHILE $g'>0$ **DO**
  $k = g/g'$
  $g'' = g - kg'$; // $g'' = g \% g'$
  $g = g'$;
  $g' = g''$;
  return g

- running time is $O(|a|*|b|)$

EXAMPLE

EXAMPLE

gcd(7,19)

EXAMPLE

gcd(7,19)

= gcd(19,7)

EXAMPLE

$\gcd(7,19)$

$= \gcd(19,7)$

$= \gcd(7, 19 \bmod 7)$

EXAMPLE

$\gcd(7,19)$

$= \gcd(19,7)$

$= \gcd(7,19 \bmod 7)$

$= \gcd(7,5)$

EXAMPLE

$\gcd(7,19)$
$= \gcd(19,7)$
$= \gcd(7,19 \bmod 7)$
$= \gcd(7,5)$
$= \gcd(5,7 \bmod 5)$

EXAMPLE

$\gcd(7,19)$
$= \gcd(19,7)$
$= \gcd(7,19 \bmod 7)$
$= \gcd(7,5)$
$= \gcd(5,7 \bmod 5)$
$= \gcd(5,2)$

EXAMPLE

gcd(7,19)

= gcd(19,7)

= gcd(7,19 mod 7)

= gcd(7,5)

= gcd(5,7 mod 5)

= gcd(5,2)

= gcd(2,5 mod 2)

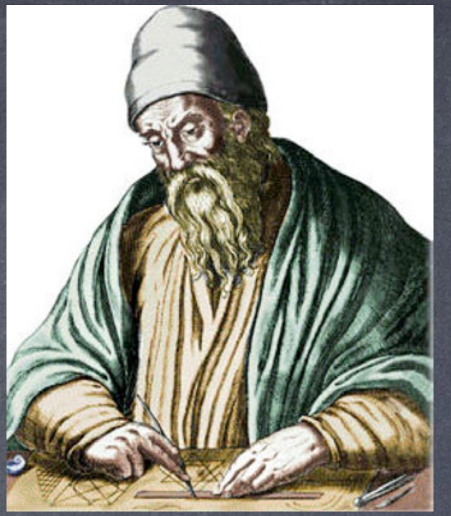
EXAMPLE

gcd(7,19)
= gcd(19,7)
= gcd(7,19 mod 7)
= gcd(7,5)
= gcd(5,7 mod 5)
= gcd(5,2)
= gcd(2,5 mod 2)
= gcd(2,1)

EXAMPLE

$\gcd(7,19)$
$= \gcd(19,7)$
$= \gcd(7,19 \bmod 7)$
$= \gcd(7,5)$
$= \gcd(5,7 \bmod 5)$
$= \gcd(5,2)$
$= \gcd(2,5 \bmod 2)$
$= \gcd(2,1)$
$= \gcd(1,2 \bmod 1) = \gcd(1,0) = 1$

Euclid's Extended Algorithm



The idea behind the following iterative algorithm is to maintain in each iteration the relations $g = ax + by$ and $g' = ax' + by'$ while reducing the value of g.

At the end of the algorithm, the value of g is $\gcd(a, b)$. The final value of x is such that $ax \equiv g \pmod{b}$ and by symmetry, the final value of y is such that $by \equiv g \pmod{a}$. When $\gcd(a, b) = 1$, we find that x is the multiplicative inverse of a modulo b and that y is the multiplicative inverse of b modulo a.

Extended Euclidian Algorithm

- **Input**: integers a,b.
- **Output**: g,x,y such that g=GCD(a,b) and $g = ax+by$.

$g = a$; $g' = b$; $x = 1$; $y = 0$; $x' = 0$; $y' = 1$;
WHILE g'>0 **DO**
  $k = g/g'$
  $g'' = g-kg'$; $x'' = x-kx'$; $y'' = y-ky'$;   // $g'' = g \% g'$
  $g = g'$; $x = x'$; $y = y'$;
  $g' = g''$; $x' = x''$; $y' = y''$;
return g,x,y

- running time is $O(|a|*|b|)$

EXAMPLE

EXAMPLE

gcd(7,19)

$x=1, y=0, x'=0, y'=1$

EXAMPLE

gcd(7,19) $x=1, y=0, x'=0, y'=1$

= gcd(19,7) $x=0, y=1, x'=1, y'=0$

EXAMPLE

gcd(7,19) $x=1, y=0, x'=0, y'=1$

= gcd(19,7) $x=0, y=1, x'=1, y'=0$

= gcd(7,19 mod 7)

EXAMPLE

$\gcd(7,19)$ $x=1, y=0, x'=0, y'=1$

$= \gcd(19,7)$ $x=0, y=1, x'=1, y'=0$

$= \gcd(7, 19 \bmod 7)$

$= \gcd(7,5)$ $x=1, y=0, x'=-2, y'=1$

EXAMPLE

$\gcd(7,19)$ $\quad x=1, y=0, x'=0, y'=1$

$= \gcd(19,7)$ $\quad x=0, y=1, x'=1, y'=0$

$= \gcd(7, 19 \bmod 7)$

$= \gcd(7,5)$ $\quad x=1, y=0, x'=-2, y'=1$

$= \gcd(5, 7 \bmod 5)$

EXAMPLE

$\gcd(7,19)$	$x=1, y=0, x'=0, y'=1$
$= \gcd(19,7)$	$x=0, y=1, x'=1, y'=0$
$= \gcd(7, 19 \bmod 7)$	
$= \gcd(7,5)$	$x=1, y=0, x'=-2, y'=1$
$= \gcd(5, 7 \bmod 5)$	
$= \gcd(5,2)$	$x=-2, y=1, x'=3, y'=-1$

EXAMPLE

gcd(7,19) $\quad x=1, y=0, x'=0, y'=1$

= gcd(19,7) $\quad x=0, y=1, x'=1, y'=0$

= gcd(7,19 mod 7)

= gcd(7,5) $\quad x=1, y=0, x'=-2, y'=1$

= gcd(5,7 mod 5)

= gcd(5,2) $\quad x=-2, y=1, x'=3, y'=-1$

= gcd(2,5 mod 2)

EXAMPLE

$\gcd(7,19)$	$x=1, y=0, x'=0, y'=1$
$= \gcd(19,7)$	$x=0, y=1, x'=1, y'=0$
$= \gcd(7, 19 \bmod 7)$	
$= \gcd(7,5)$	$x=1, y=0, x'=-2, y'=1$
$= \gcd(5, 7 \bmod 5)$	
$= \gcd(5,2)$	$x=-2, y=1, x'=3, y'=-1$
$= \gcd(2, 5 \bmod 2)$	
$= \gcd(2,1)$	$x=3, y=-1, x'=-8, y'=3$

EXAMPLE

gcd(7,19)	x=1,y=0,x′=0,y′=1
= gcd(19,7)	x=0,y=1,x′=1,y′=0
= gcd(7,19 mod 7)	
= gcd(7,5)	x=1,y=0,x′=-2,y′=1
= gcd(5,7 mod 5)	
= gcd(5,2)	x=-2,y=1,x′=3,y′=-1
= gcd(2,5 mod 2)	
= gcd(2,1)	x=3,y=-1,x′=-8,y′=3
= gcd(1,2 mod 1)	

EXAMPLE

$\gcd(7,19)$	$x=1, y=0, x'=0, y'=1$
$= \gcd(19,7)$	$x=0, y=1, x'=1, y'=0$
$= \gcd(7, 19 \bmod 7)$	
$= \gcd(7,5)$	$x=1, y=0, x'=-2, y'=1$
$= \gcd(5, 7 \bmod 5)$	
$= \gcd(5,2)$	$x=-2, y=1, x'=3, y'=-1$
$= \gcd(2, 5 \bmod 2)$	
$= \gcd(2,1)$	$x=3, y=-1, x'=-8, y'=3$
$= \gcd(1, 2 \bmod 1)$	
$= \gcd(1,0) = 1$	$x=-8, y=3, x'=19, y'=-7$

EXAMPLE

EXAMPLE

gcd(7,19)

EXAMPLE

gcd(7,19)

= gcd(1,0) = 1

$x=-8, y=3, x'=19, y'=-7$

EXAMPLE

$\gcd(7,19)$

$= \gcd(1,0) = 1$

$x=-8, y=3, x'=19, y'=-7$

EXAMPLE

gcd(7,19)

= gcd(1,0) = 1 $x=-8, y=3, x'=19, y'=-7$

thus

EXAMPLE

gcd(7,19)

= gcd(1,0) = 1 $\quad$ $x=-8, y=3, x'=19, y'=-7$

thus

$$1 = -8 \times 7 + 3 \times 19$$

EXAMPLE

$\gcd(7,19)$

$= \gcd(1,0) = 1$ $x=-8, y=3, x'=19, y'=-7$

thus

$$1 = -8 \times 7 + 3 \times 19$$

and

EXAMPLE

gcd(7,19)

= gcd(1,0) = 1 $\quad$ $x=-8, y=3, x'=19, y'=-7$

thus

$$1 = -8 \times 7 + 3 \times 19$$

and

$$7^{-1} \bmod 19 = 11 = -8 \bmod 19$$

EXAMPLE

gcd(7,19)

= gcd(1,0) = 1 $\quad$ x=-8,y=3,x'=19,y'=-7

thus

$$1 = -8 \times 7 + 3 \times 19$$

and

$$7^{-1} \bmod 19 = 11 = -8 \bmod 19$$

$$19^{-1} \bmod 7 = 3 = 3 \bmod 7$$

Primality Testing

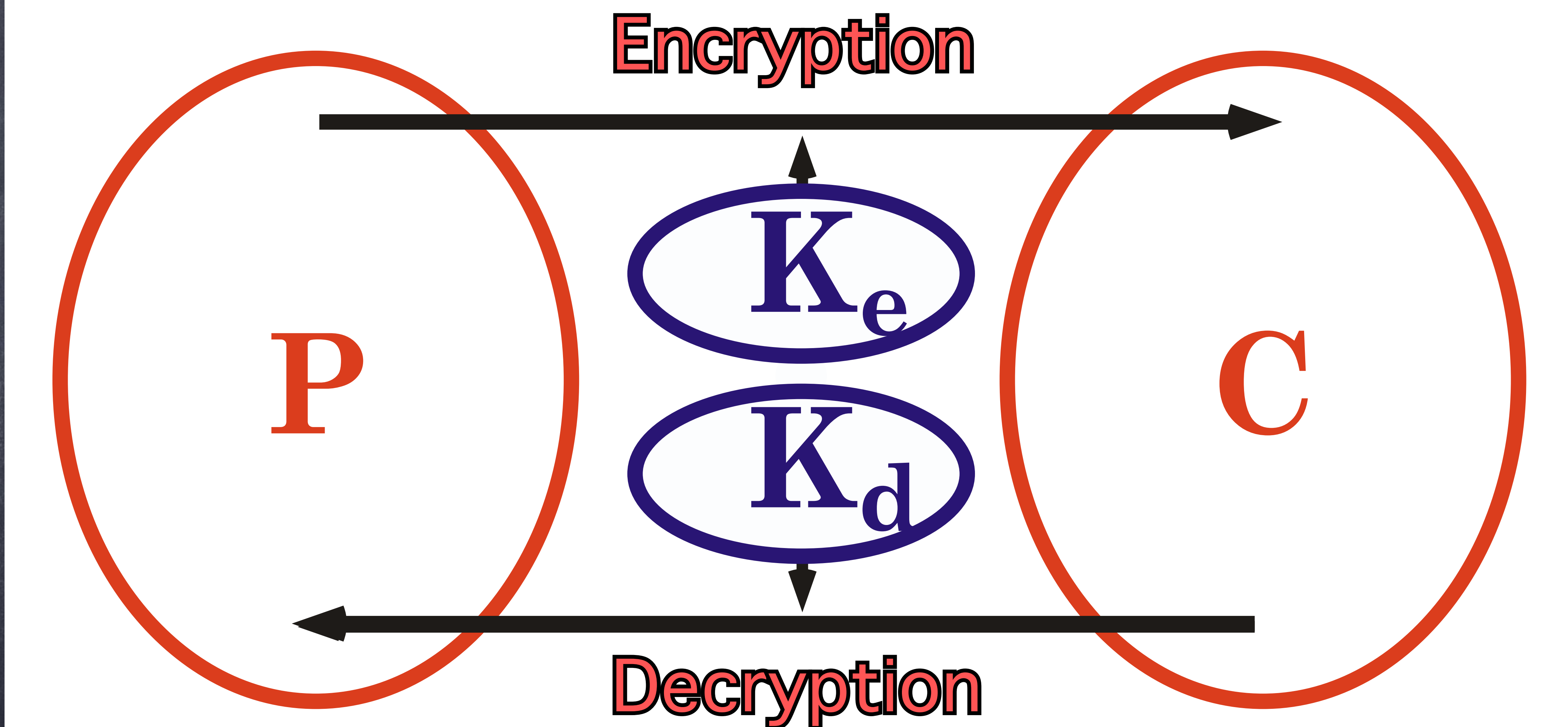
- **Input**: base a, modulus N.
- **Output**: Is N a base-a pseudo-prime? .

IF GCD(a,N) > 1 **THEN return False**
set $s \geq 0$ and t (odd) s.t. $N\text{-}1 = t2^s$
$x = a^2 \,\%N$; $y = N\text{-}1$
FOR i = 1 **TO** s
  IF x = 1 AND y = N-1 **THEN return True**
  $y = x$; $x = x^2 \,\%N$
return False

- running time is $O(|N|^4)$

Asymmetric Encryption

(Public-Key Cryptography)



Complexity Theoretical Security

Definitions

DEFINITION A public-key encryption scheme *is a tuple of PPT algorithms* (**Gen,Enc,Dec**) *s.t. :*

1. *The* key generation algorithm **Gen** *takes as input the security parameter* 1^n *and outputs a pair of keys* (***pk,sk***)*. We refer to the first of these as the* public key *and the second as the* private key*. We assume for convenience that* ***pk*** *and* ***sk*** *each have length at least* ***n****, and that* ***n*** *can be determined from* ***pk, sk****.*

Definitions

2. *The* encryption algorithm **Enc** *takes as input a public key* ***pk*** *and a message* ***m*** *from some underlying plaintext space. It outputs a ciphertext* ***c****, and we write this as*

$$c \leftarrow \mathbf{Enc}_{pk}(m).$$

3. *The* decryption algorithm **Dec** *takes as input a private key* ***sk*** *and a ciphertext* ***c****, and outputs a message* ***m*** *or a special symbol* $\perp$ *denoting failure. We assume without loss of generality that* **Dec** *is deterministic, and write this as*

$$m \coloneqq \mathbf{Dec}_{sk}(c).$$

Definitions

It is required that there exists a negligible function **negl** *such that for every* **n**, *every* **(*pk*,*sk*)** *output by* **Gen(1*ⁿ*)**, *and every message* **m** *in the appropriate underlying plaintext space, it holds that*

$$\Pr[\,\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) \neq m\,] \leq \mathsf{negl}(n).$$

RSA Encryption

Public inventors

Private inventors

Ellis, Cocks, Williamson

1970's

RSA Key Generation

- **Input**: Security parameter 1^n.
- **Output**: N, e, d of size n.

$N \leftarrow p*q$ // for some large random primes p, q
$\varphi(N) = (p\text{-}1)*(q\text{-}1)$
choose randomly e such that GCD(e , $\varphi(N)$)=1
compute d = e^{-1} mod $\varphi(N)$
return (N,e,d)

In Cocks' variation, e=*N* and therefore $d=N^{-1}$ mod $\varphi(N)$.

RSA Encryption

- **Gen**: on input 1^n run GenRSA(1^n) and obtain (N,e,d). Let $\langle N,e \rangle$ be the public-key and $\langle d \rangle$ the private key.
- **Enc**: on input $\langle N,e \rangle$ and a message $0<m<N$ compute
$$c = m^e \bmod N$$
- **Dec**: on input $\langle d \rangle$ and a ciphertext $0<c<N$ compute
$$m = c^d \bmod N$$

The RSA Assumption

- The **RSA** problem can be described informally as follows: given a modulus N, an integer (exponent) $e > 0$ that is relatively prime to $\varphi(N)$, and an element $y \in \mathbb{Z}_N^*$, compute $\sqrt[e]{y} \bmod N$;

  Given N,e,y find x such that $x^e = y \bmod N$.

The RSA Assumption

The **RSA** experiment $\textbf{RSA-inv}_{A,\text{GenRSA}}(n)$:

1. *Run* $\textbf{GenRSA}(1^n)$ *to obtain* (N, e, d).

2. *Choose* $y \leftarrow \mathbb{Z}_N^*$.

3. *A is given* N, e, y, *and outputs* $x \in \mathbb{Z}_N^*$.

4. *The output of the experiment is defined to be* **1** *if* $y = x^e \bmod N$, *and* **0** *otherwise.*

The RSA Assumption

DEFINITION We say that the **RSA** problem is hard relative to **GenRSA** *if for all probabilistic polynomial-time algorithms A there exists a negligible function* **negl** *such that*

$$\Pr[\,\mathsf{RSA\text{-}inv}_{A,\mathsf{GenRSA}}(n) = 1\,] \leq \mathsf{negl}(n).$$

Quantum Factoring

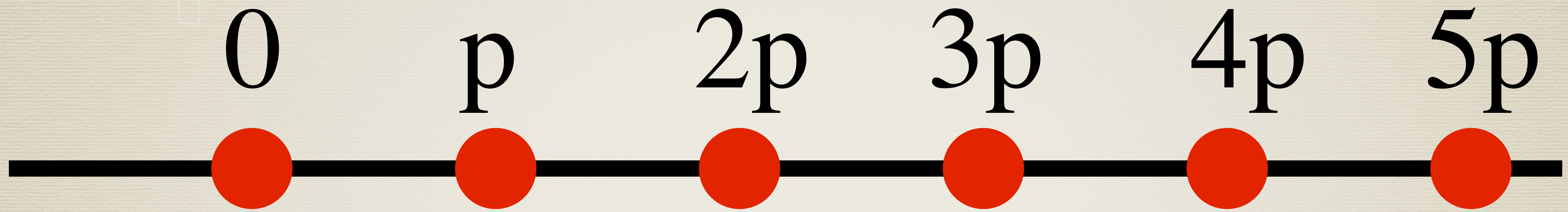
1990’s

Approximate Integer GCD Based Cryptography

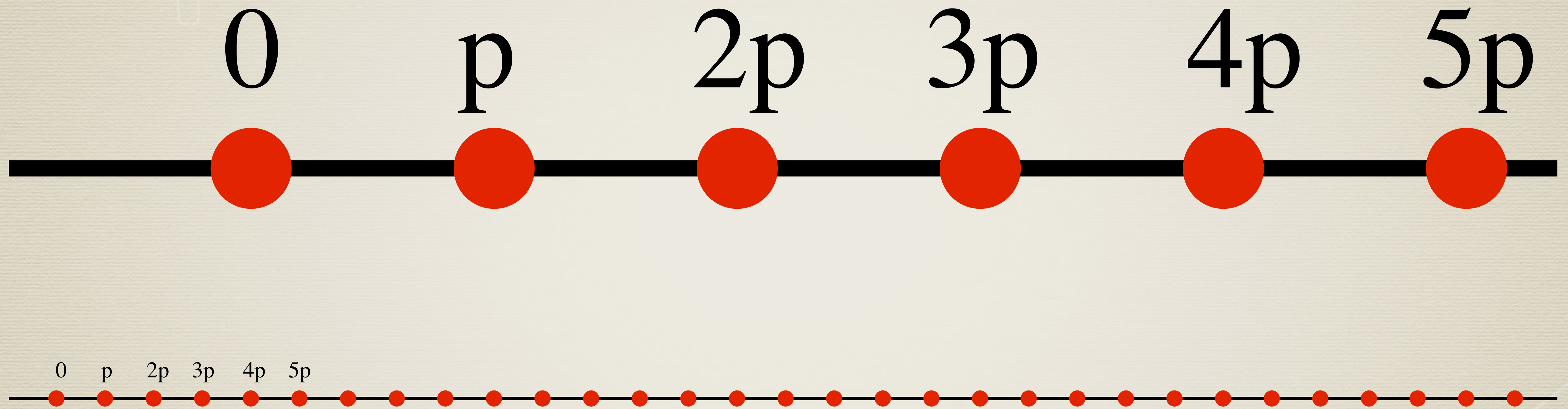
§

2010's

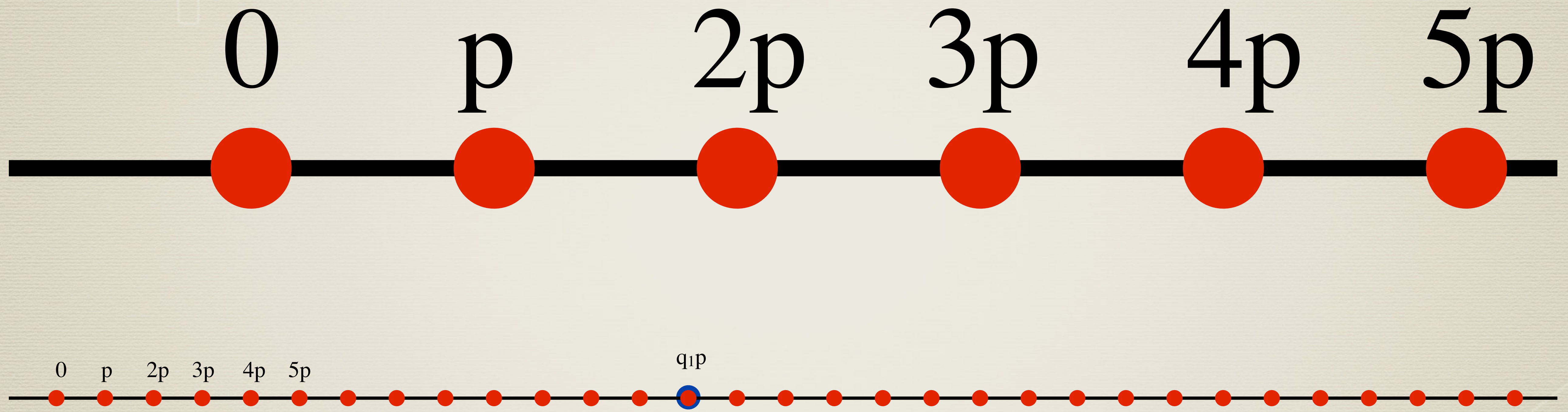
Approximate Integer GCD



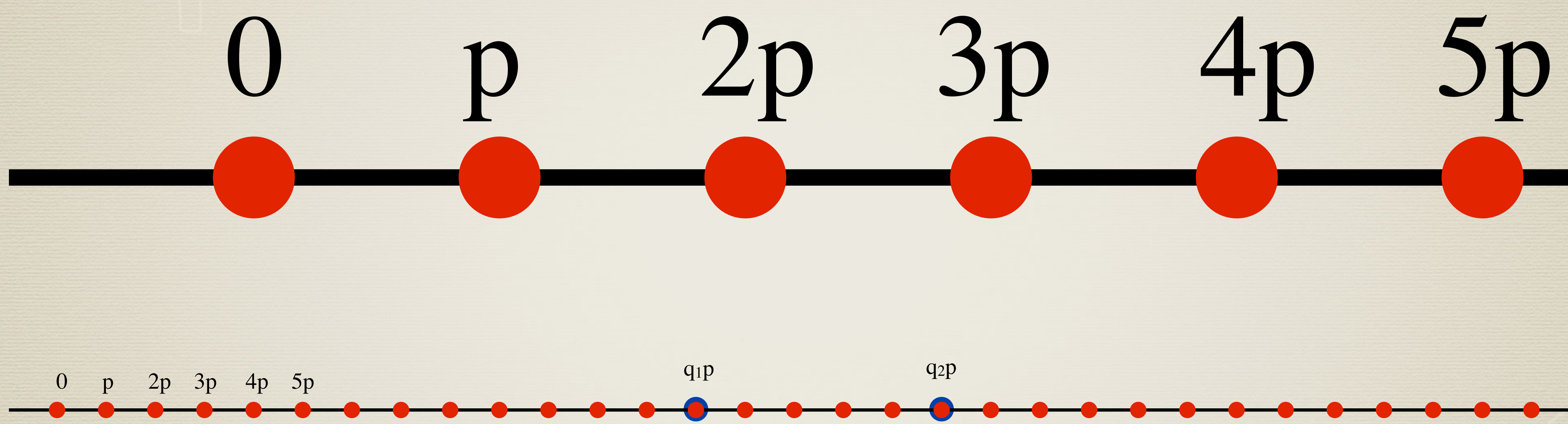
Approximate Integer GCD



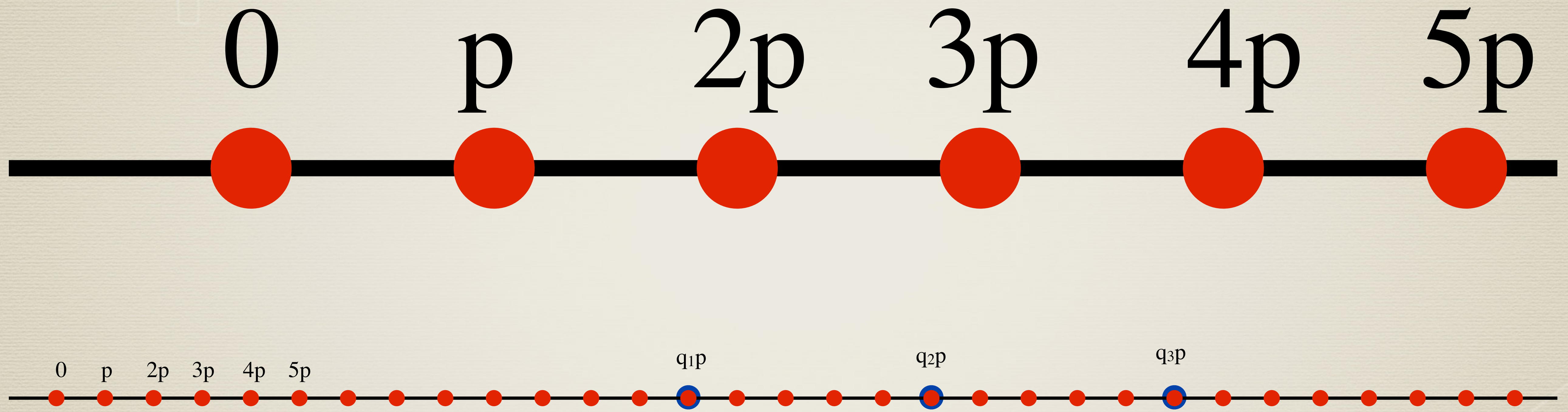
Approximate Integer GCD



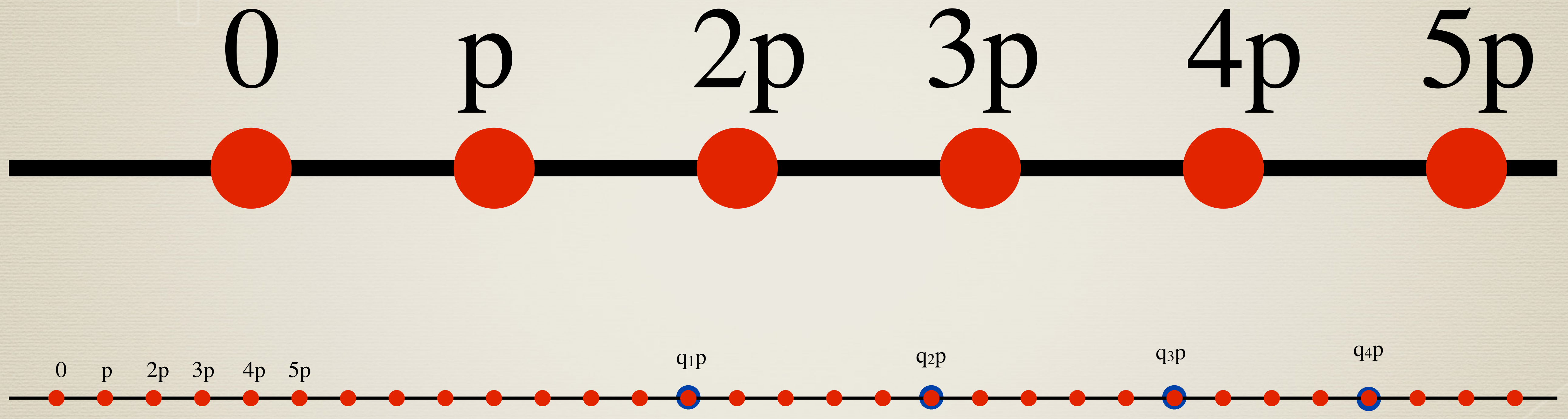
Approximate Integer GCD



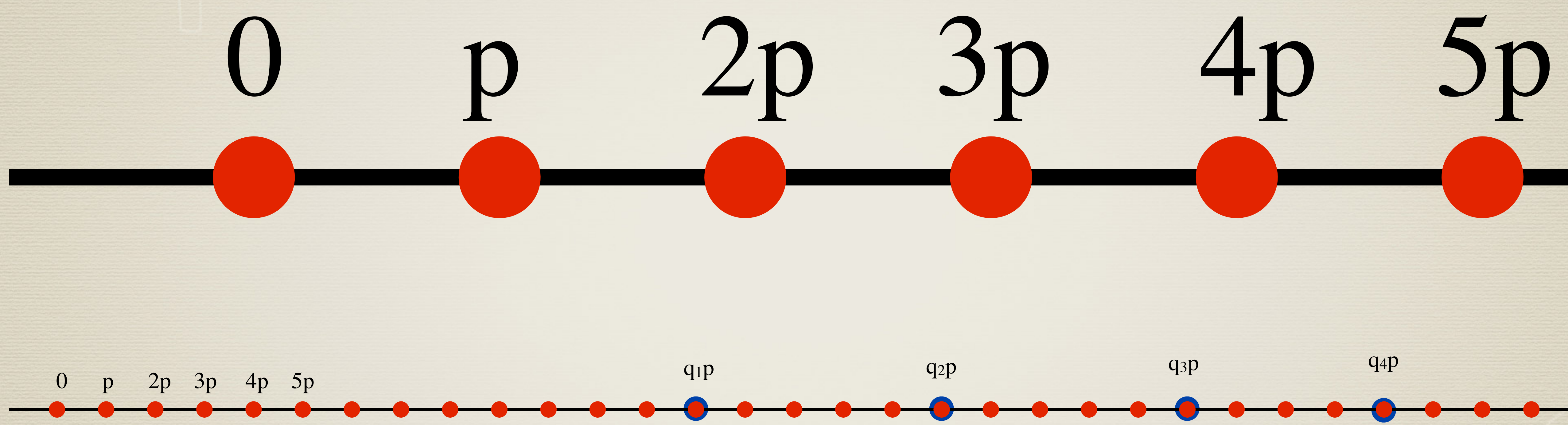
Approximate Integer GCD



Approximate Integer GCD

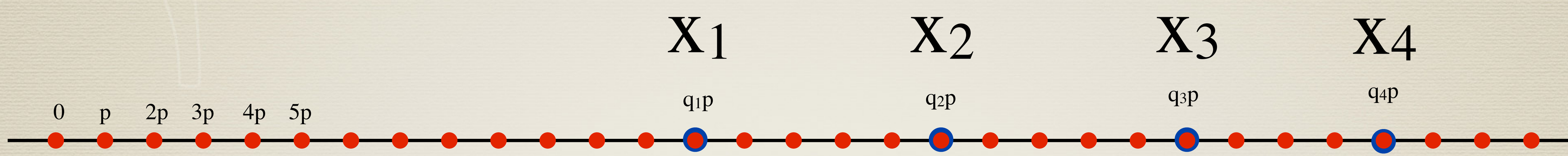


Approximate Integer GCD



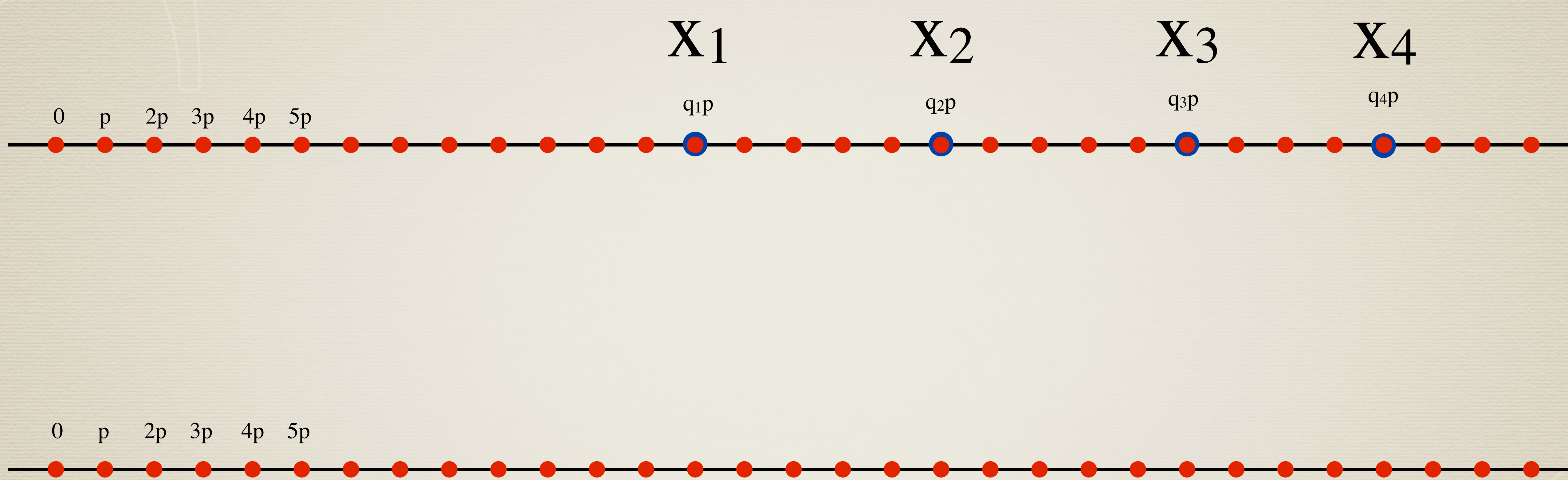
$$GCD(q_1p, q_2p, q_3p, q_4p) = p$$

Approximate Integer GCD



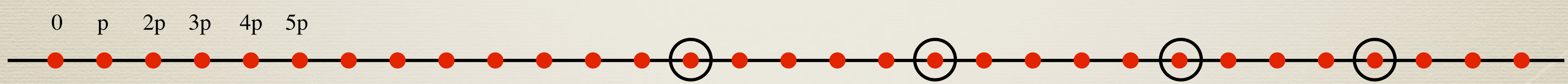
$$\mathrm{GCD}(x_1, x_2, x_3, x_4) = p$$

Approximate Integer GCD



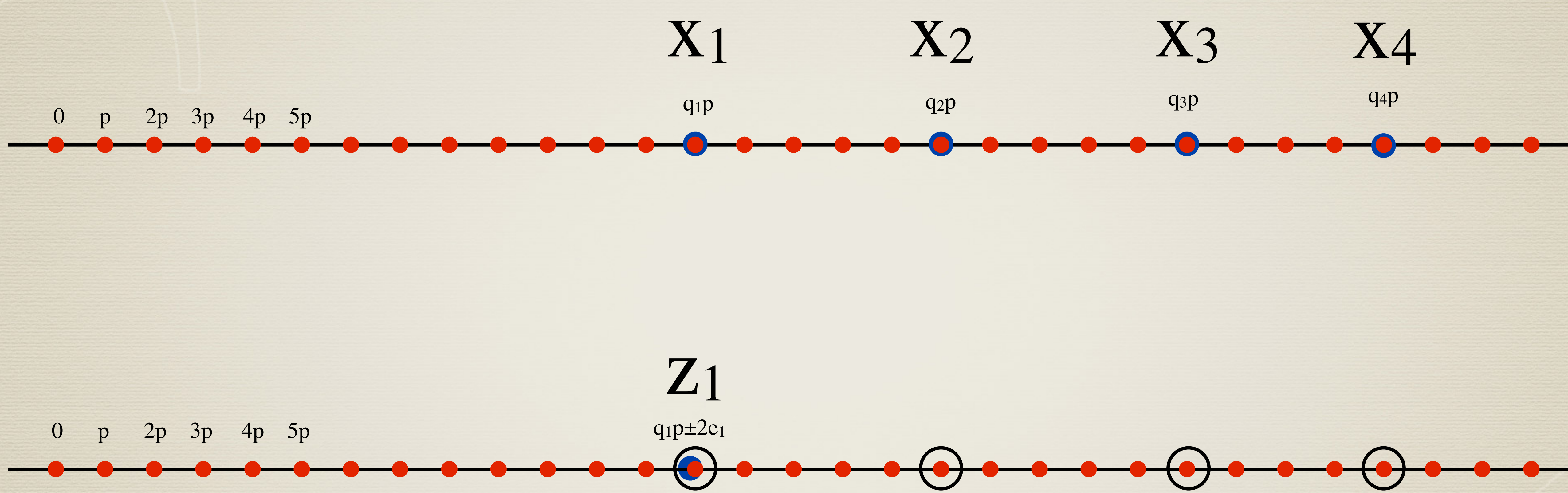
$$GCD(x_1,x_2,x_3,x_4) = p$$

Approximate Integer GCD



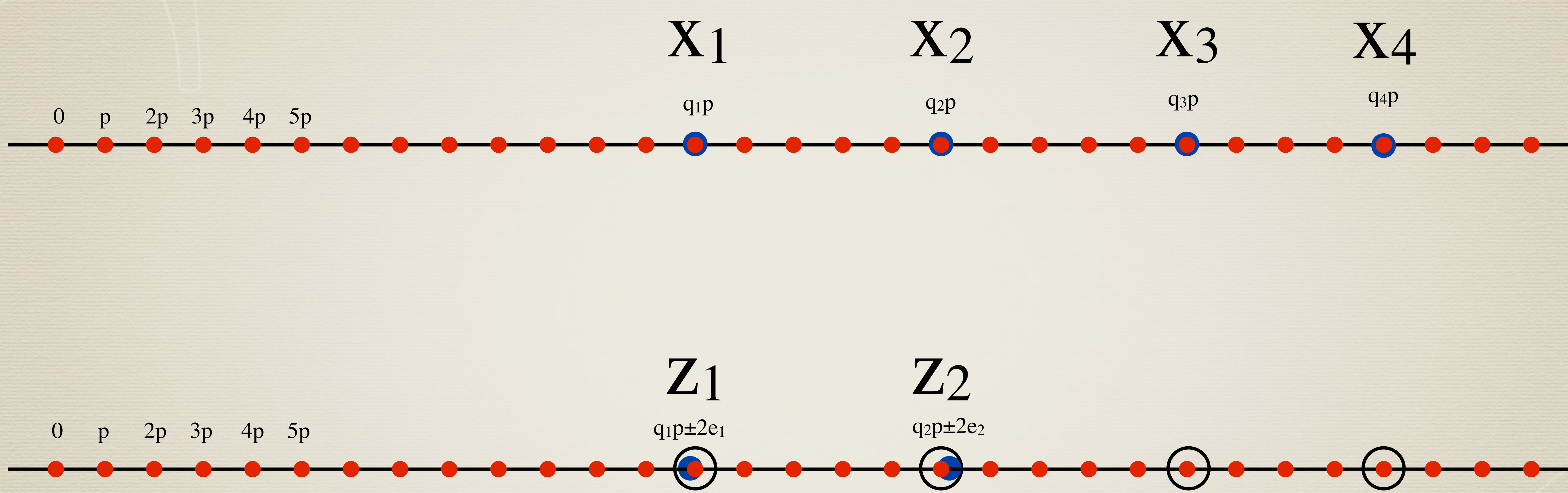
$$\mathrm{GCD}(x_1, x_2, x_3, x_4) = p$$

Approximate Integer GCD



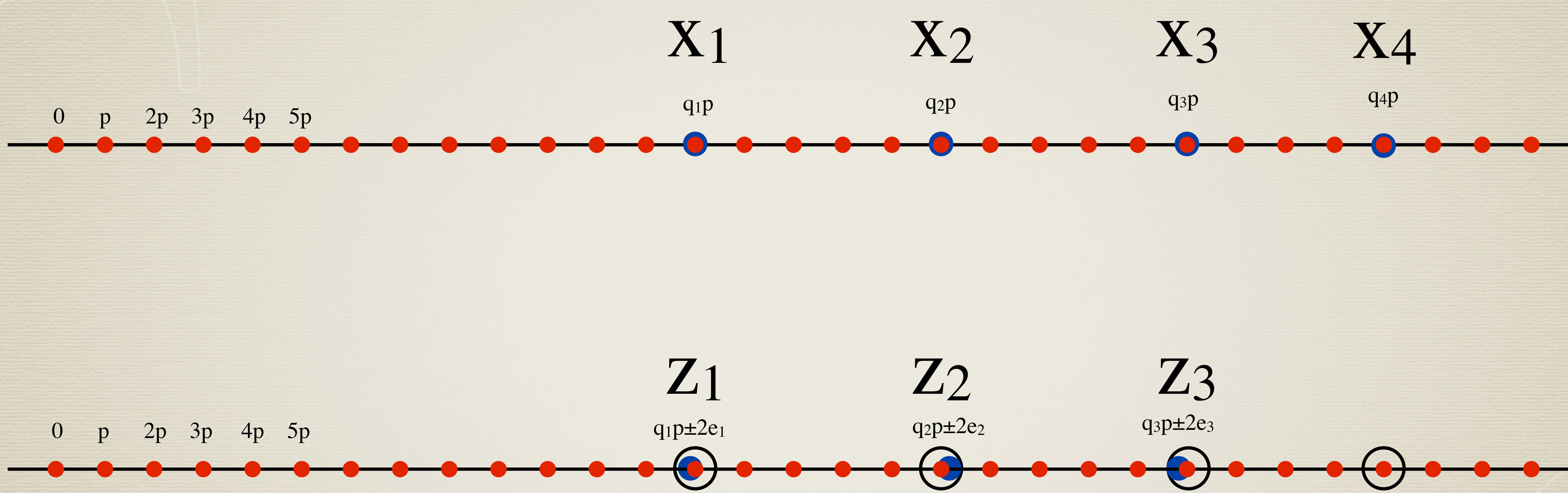
$$GCD(x_1, x_2, x_3, x_4) = p$$

Approximate Integer GCD



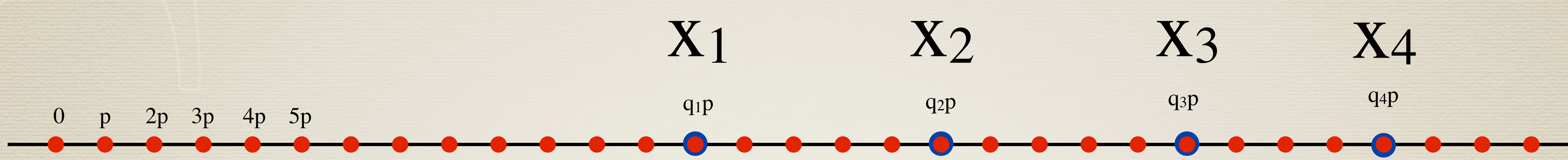
$$\mathrm{GCD}(x_1, x_2, x_3, x_4) = p$$

Approximate Integer GCD



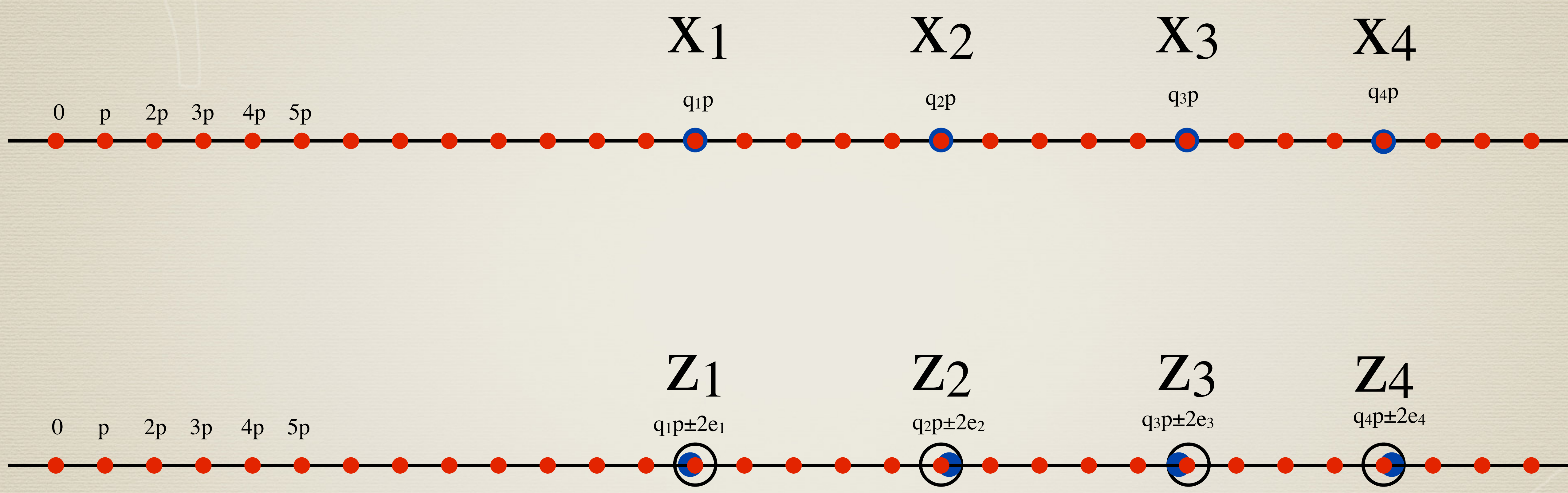
$$\mathrm{GCD}(x_1, x_2, x_3, x_4) = p$$

Approximate Integer GCD



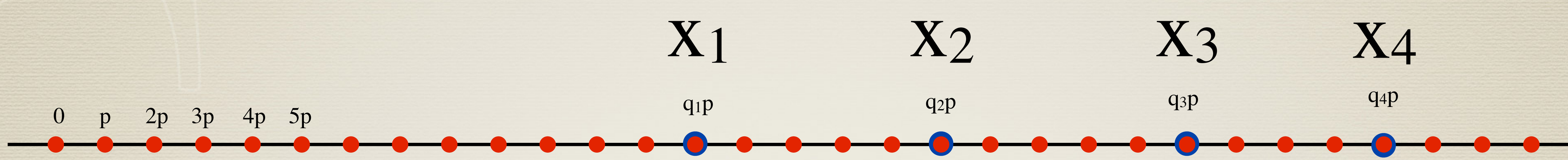
$$\mathrm{GCD}(x_1, x_2, x_3, x_4) = p$$

Approximate Integer GCD



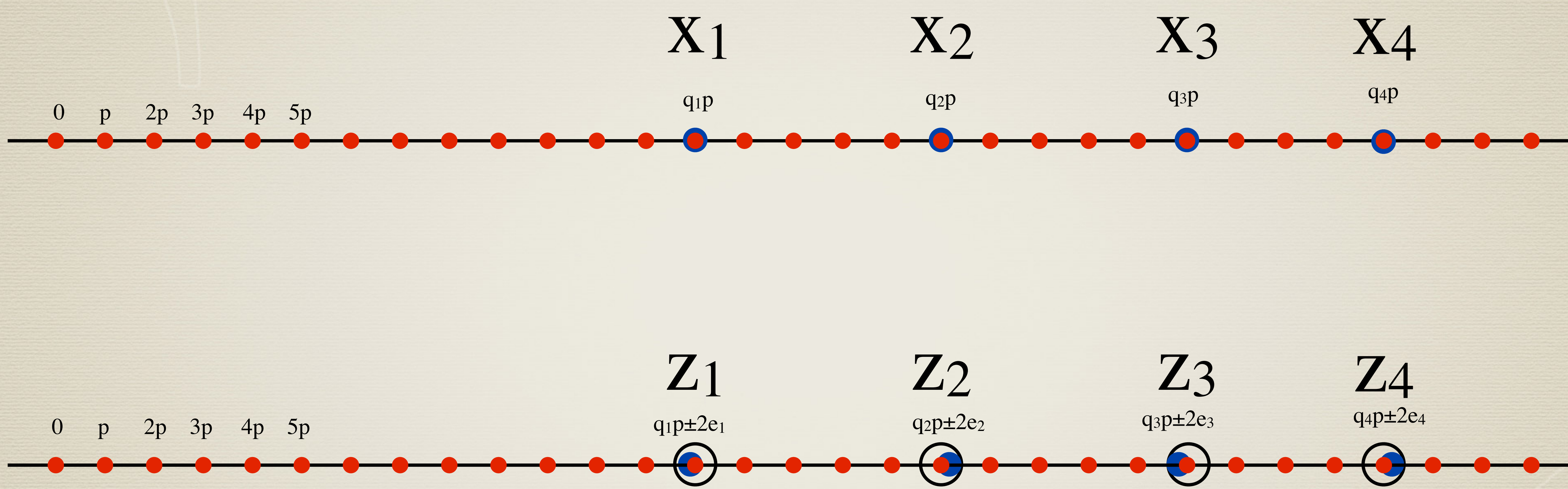
$$\mathrm{GCD}(x_1, x_2, x_3, x_4) = p$$
$$\mathrm{GCD}(z_1, z_2, z_3, z_4) = 1$$

Approximate Integer GCD



$$\mathrm{GCD}(x_1, x_2, x_3, x_4) = p$$

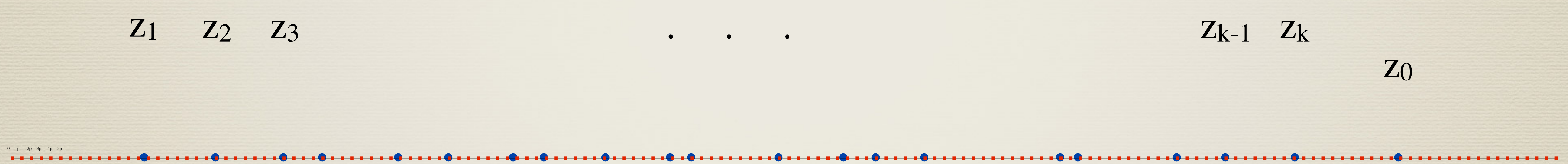
Approximate Integer GCD



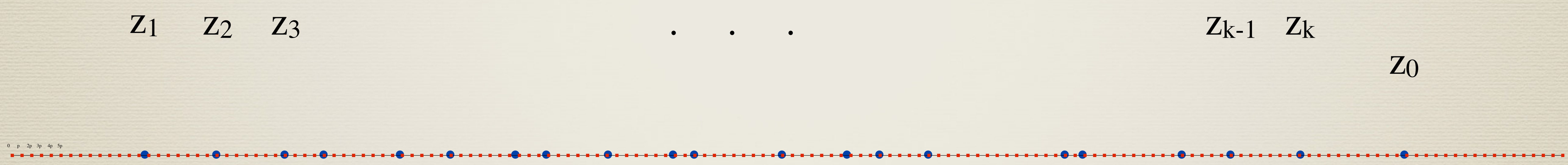
$$GCD(x_1,x_2,x_3,x_4) = p$$

<u>AIGCD</u> : find p from z_1,z_2,z_3,z_4 ?

Approximate Integer GCD

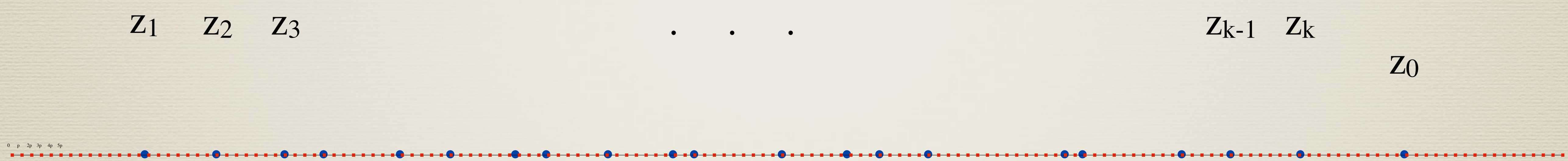


Approximate Integer GCD



$$\sum_{1\le i\le k} s_i z_i \bmod z_0$$

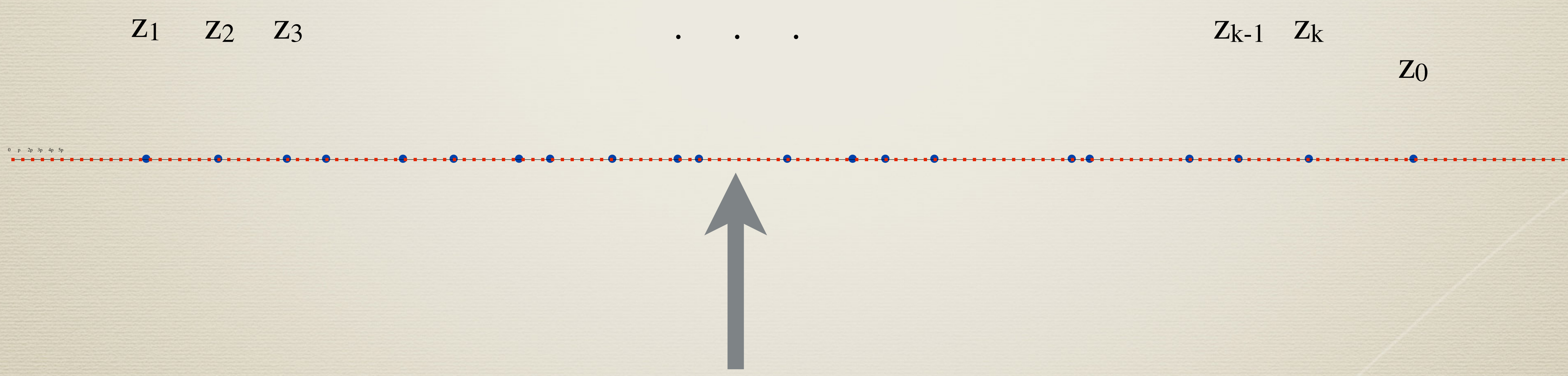
Approximate Integer GCD



$$\sum_{1 \le i \le k} s_i z_i \bmod z_0$$

$$s_i \in \{0,1\}$$

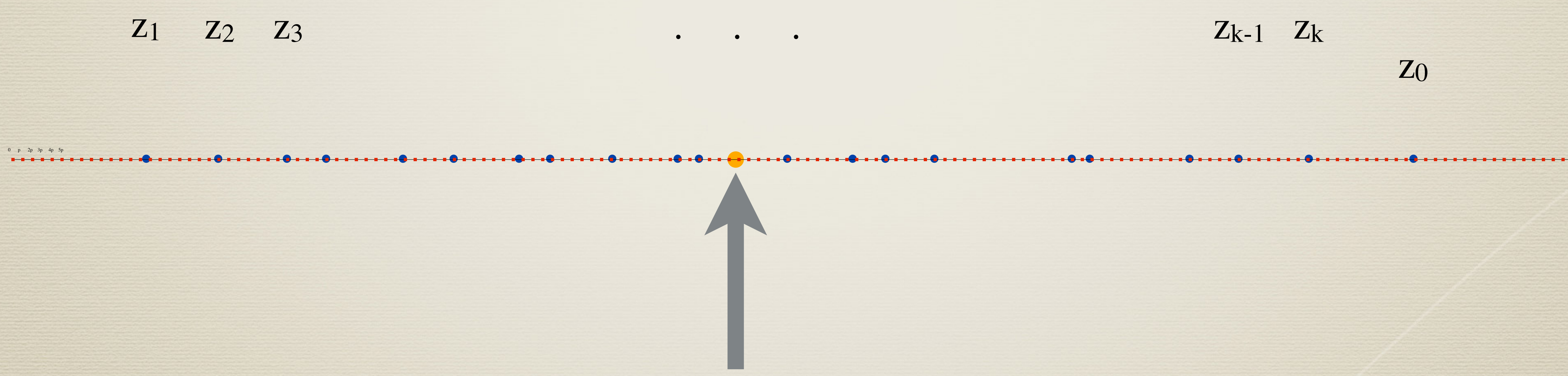
Approximate Integer GCD



$$\sum_{1 \le i \le k} s_i z_i \bmod z_0$$

$$s_i \in \{0,1\}$$

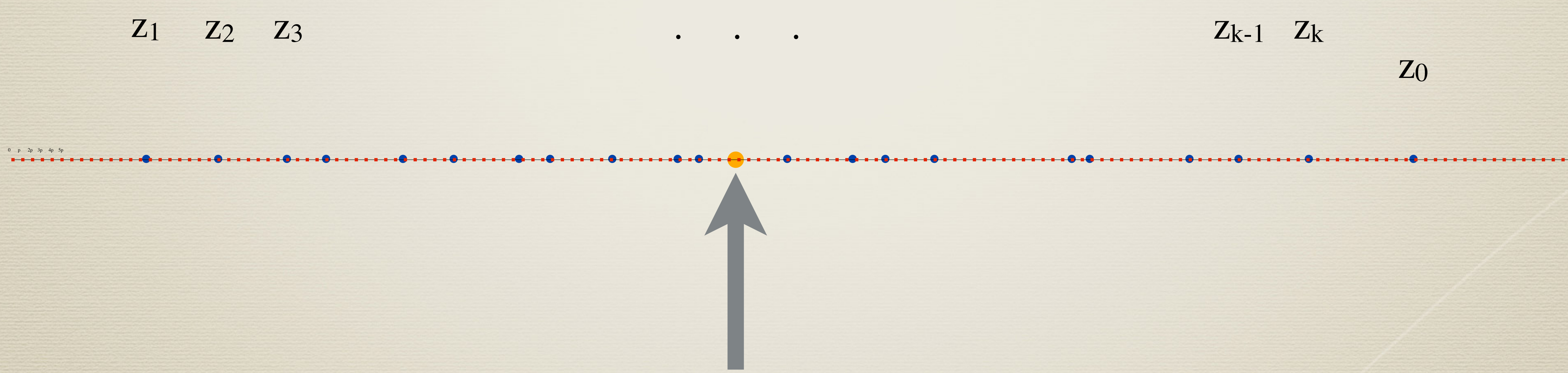
Approximate Integer GCD



$$\sum_{1 \le i \le k} s_i z_i \bmod z_0$$

$$s_i \in \{0,1\}$$

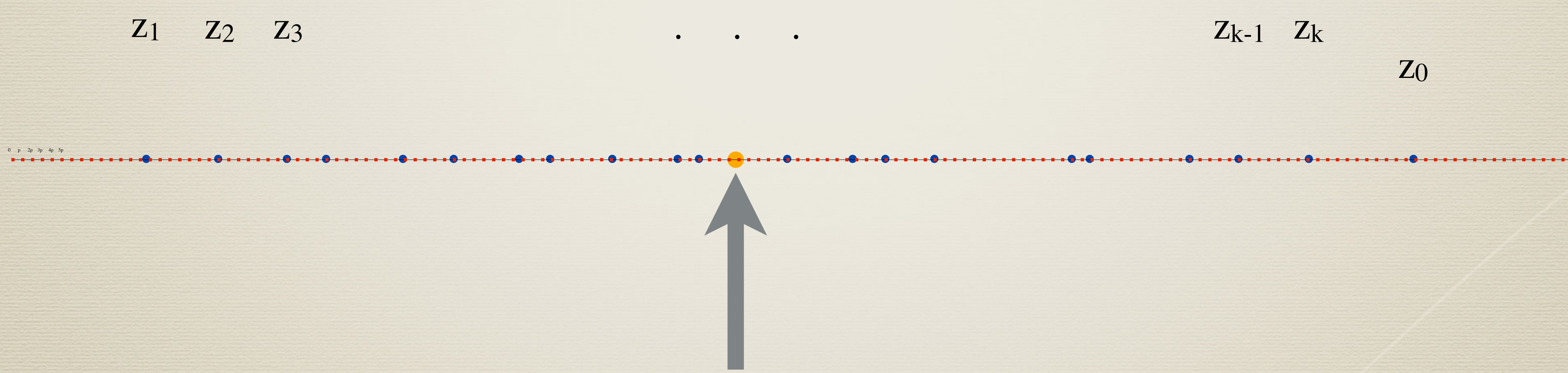
Approximate Integer GCD



$$\sum_{1\le i\le k} s_i z_i \bmod z_0 \approx \sum_{1\le i\le k} s_i x_i \bmod x_0$$

$$s_i \in \{0,1\}$$

Approximate Integer GCD

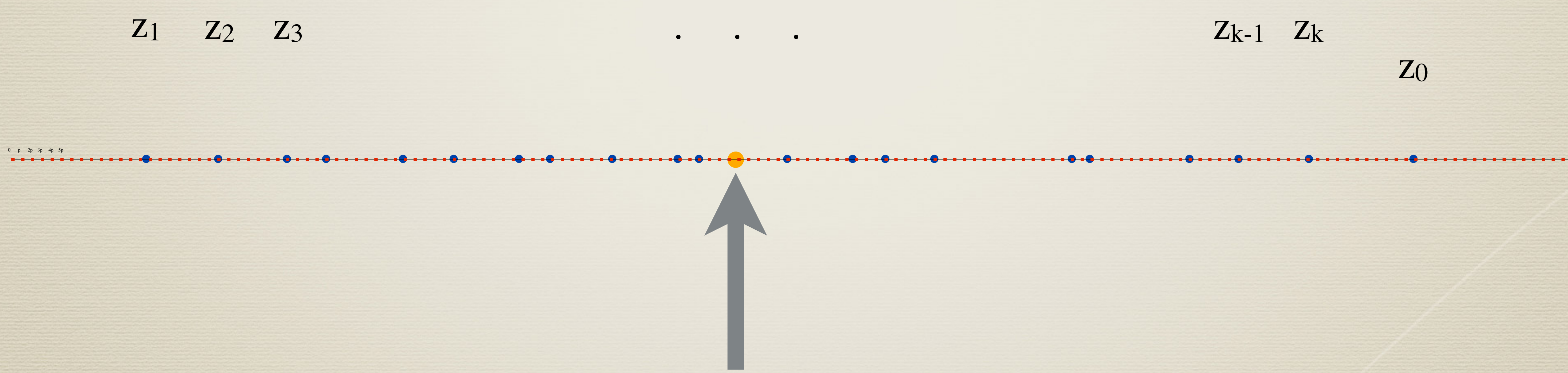


$$\sum_{1\le i\le k} s_i z_i \bmod z_0 \approx \sum_{1\le i\le k} s_i x_i \bmod x_0$$

$$\pm 2(ke_0 + \sum_{1\le i\le k} e_i)$$

$$s_i \in \{0,1\}$$

Approximate Integer GCD



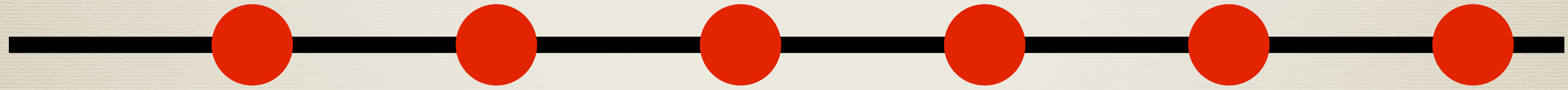
$$\left| \sum_{1\le i\le k} s_i z_i \bmod z_0 - \Big(\sum_{1\le i\le k} s_i q_i \bmod q_0\Big)\times p \right|$$

$$s_i \in \{0,1\}$$

$$\le 4k|e_{max}|$$

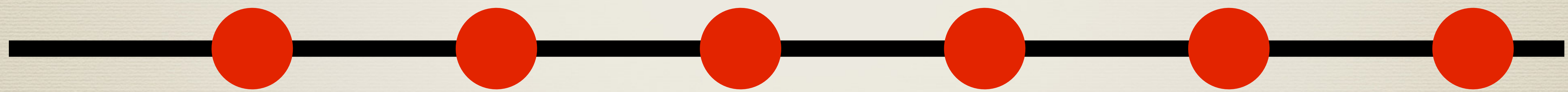
Approximate Integer GCD

Approximate Integer GCD



Approximate Integer GCD

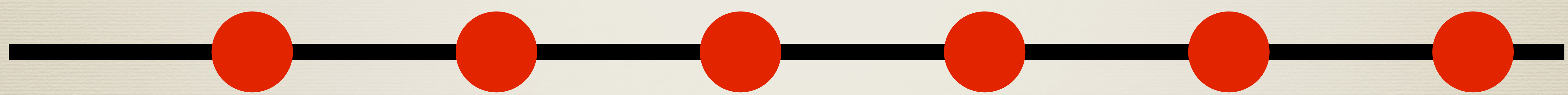
$$\Omega(s)=\sum_{1\leq i\leq k} s_i z_i \bmod z_0$$



Approximate Integer GCD

$$\Omega(s)=\sum_{1\leq i\leq k} s_i z_i \bmod z_0$$

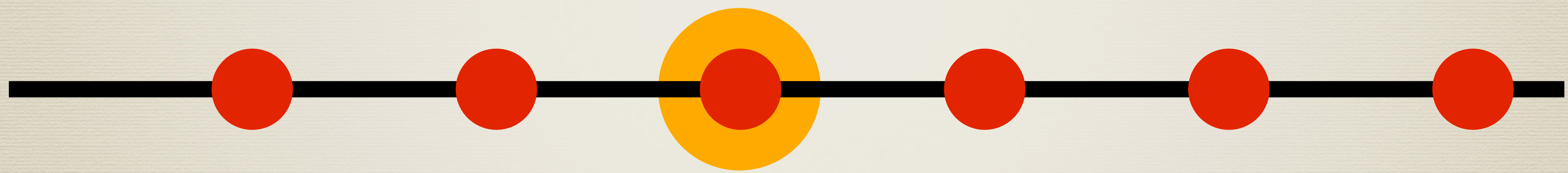
$$s\in\{0,1\}^n$$



Approximate Integer GCD

$$\Omega(s)=\sum_{1\le i\le k} s_i z_i \bmod z_0$$

$$s\in\{0,1\}^n$$

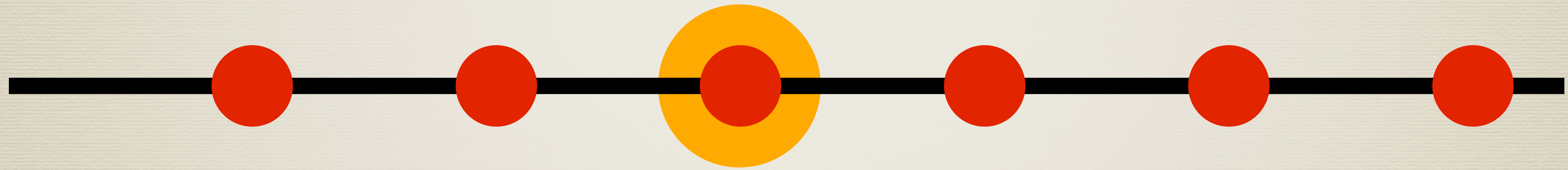


$$|e_{max}|\le \partial \ll p/8k$$

Approximate Integer GCD

$$\Omega(s)=\sum_{1\leq i\leq k} s_i z_i \bmod z_0$$

$$s\in\{0,1\}^n$$



$$|e_{max}|\leq \partial \ll p/8k$$

$\Omega(s)-p[\Omega(s)/p]$ = small even error

AIGCD encryption

AIGCD encryption

SK : p

AIGCD encryption

SK : p

PK : $z_0, z_1, z_2, \ldots, z_k$, $\partial \ll p/8k \ll \partial' \ll p/2$

AIGCD encryption

SK : p

PK : $z_0, z_1, z_2, \ldots, z_k$, $\partial \ll p/8k \ll \partial' \ll p/2$

$e_i \in_U [-\partial \ldots +\partial]$

AIGCD encryption

SK : p

PK : $z_0, z_1, z_2, \ldots, z_k$, $\partial \ll p/8k \ll \partial' \ll p/2$

$e_i \in_U [-\partial \ldots +\partial]$

$$\text{enc}(b) = \Omega(s)+2e+b$$

$s \in_U \{0,1\}^n$

$e \in_U [-\partial' \ldots +\partial']$

AIGCD encryption

SK : p

PK : $z_0, z_1, z_2, \ldots, z_k$, $\partial \ll p/8k \ll \partial' \ll p/2$

$e_i \in_U [-\partial \ldots +\partial]$

$$enc(b) = \Omega(s)+2e+b$$

$s \in_U \{0,1\}^n$

$e \in_U [-\partial' \ldots +\partial']$

$$dec(c) = c-p[c/p] \bmod 2$$

$= \text{parity of error}$

Information Assurance

- About IA at NSA
- IA Client and Partner Support
- IA News
- IA Events
- IA Mitigation Guidance
- IA Academic Outreach
- IA Business and Research
- IA Programs
  - Commercial Solutions for Classified Program
  - Global Information Grid
  - High Assurance Platform
  - Inline Media Encryptor
  - **Suite B Cryptography**
  - NSA Mobility Program

2015

Home > Information Assurance > Programs > NSA Suite B Cryptography

SEARCH

Cryptography Today

In the current global environment, rapid and secure information sharing is important to protect our Nation, its citizens and its interests. Strong cryptographic algorithms and secure protocol standards are vital tools that contribute to our national security and help address the ubiquitous need for secure, interoperable communications.

Currently, Suite B cryptographic algorithms are specified by the National Institute of Standards and Technology (NIST) and are used by NSA's Information Assurance Directorate in solutions approved for protecting classified and unclassified National Security Systems (NSS). Below, we announce preliminary plans for transitioning to quantum resistant algorithms.

PQCrypto 2016

About the Conference

- Top Page
- News
- Registration
- Stipends
- Program
- Accepted Papers
- Hot Topic Session
- Winter School
- Call for Papers
- Submit a Paper
- Proceedings
- Contacts

The Seventh International Conference on Post-Quantum Cryptography

Fukuoka, Japan, February 24-26, 2016

NSA announced preliminary plans for transitioning to quantum resistant algorithms.

NIST published a draft of the report on post-quantum cryptography NISTIR 8105.

Updates

February 24, 2016: The slides of NIST Announcement "Post-Quantum Cryptography: NIST's Plan for the Future" by Dustin Moody are now available.

February 21, 2016: Useful information concerning the winter school and the conference can be found in the **public webfolder**: https://goo.gl/UGwjua (access is read-only)

February 15, 2016: Due to high attendance, for participants, who register **on site**, only access to an **electronic version** (instead of a hardcopy) of the proceedings will be provided.

February 10, 2016: Updated programs of the conference (with hot topic session included) and the winter school are now available.

February 9, 2016: Deadline for online registration has passed. On-site registration will be available at the venue from February 22.

General Information

- Location
- Venue
- Accomodation
- Discover Fukuoka

Post-Quantum Cryptography: NIST's Plan for the Future

Dustin Moody
Post Quantum Cryptography Team
National Institute of Standards and Technology (NIST)

24 Feb 2016

Timeline

- Fall 2016 – formal Call For Proposals
- Nov 2017 – Deadline for submissions
- 3-5 years – Analysis phase
  - NIST will report its findings
- 2 years later – Draft standards ready

- Workshops
  - Early 2018 – submitter's presentations
  - One or two during the analysis phase

Winter 2016
COMP-250: Introduction to Computer Science

Lecture 14, February 25, 2016