Post-Quantum Cryptography #3

Prof. Claude Crépeau
McGill University

Post-Quantum Cryptography

- ~~Finite Fields~~ based cryptography
  - ~~Codes~~
  - ~~Multi-variate Polynomials~~
- Integers based cryptography
  - Approximate Integer GCD
  - Lattices

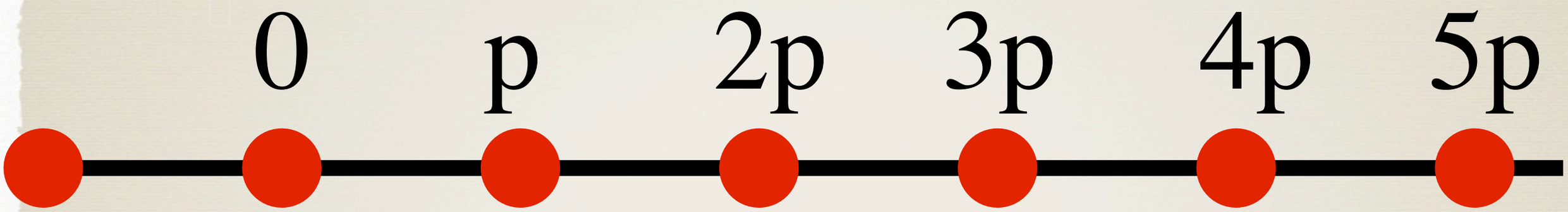
Integer based cryptography

PART 2

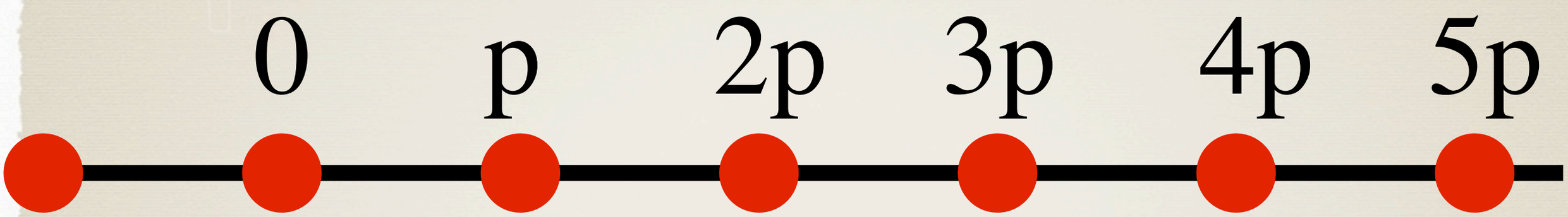
Approximate Integer GCD based crypto

§

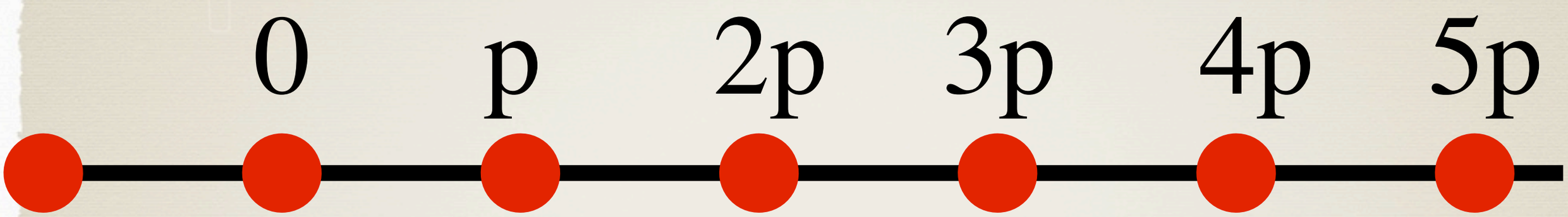
Approximate Integer GCD

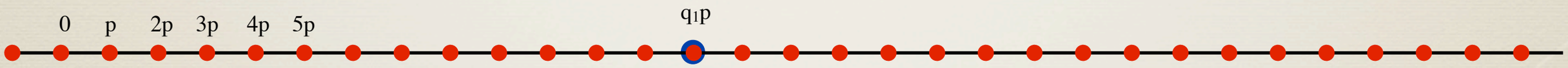


Approximate Integer GCD

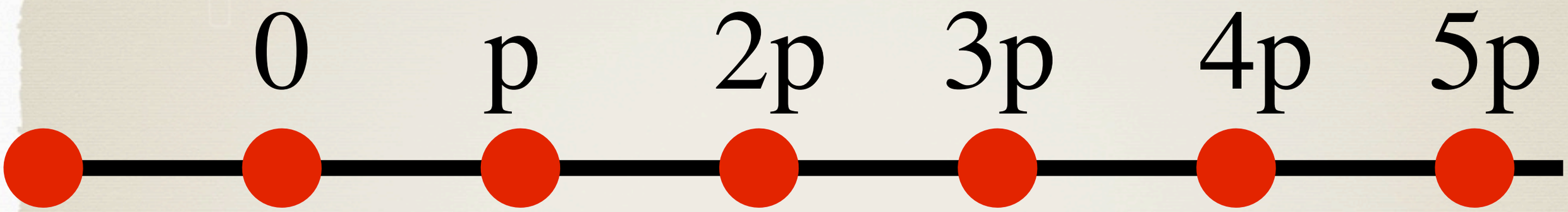


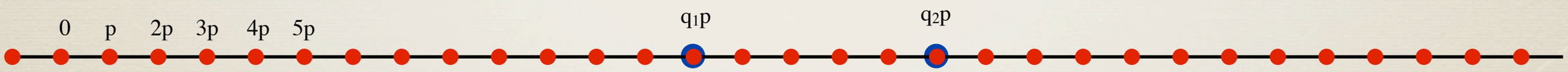
Approximate Integer GCD



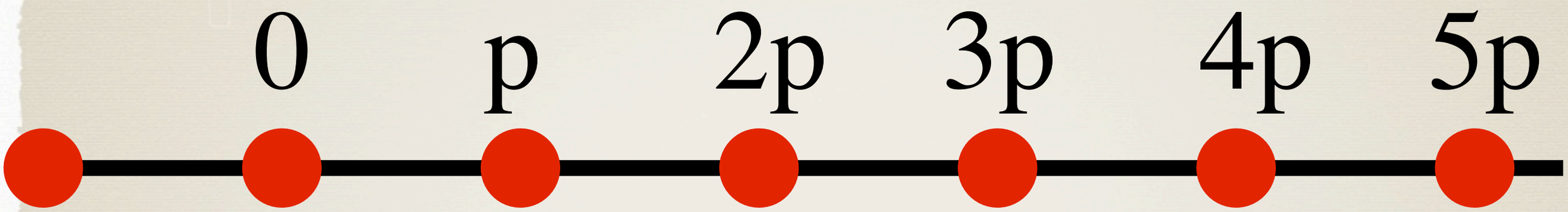


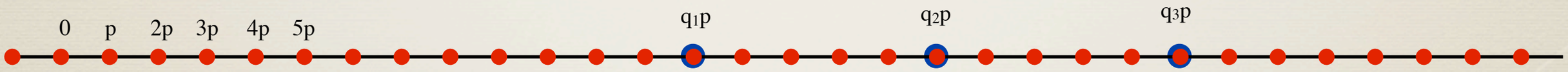
Approximate Integer GCD



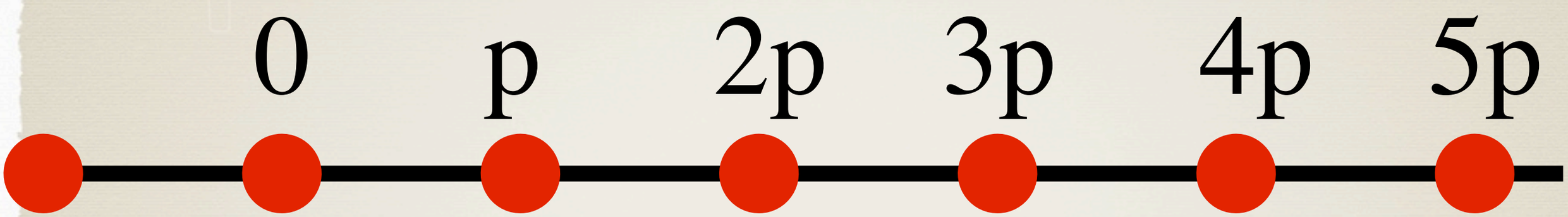


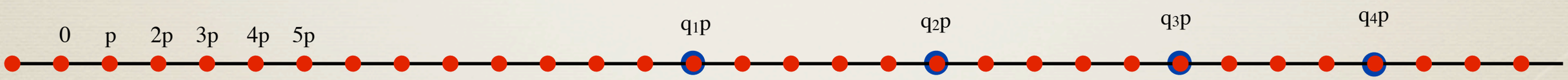
Approximate Integer GCD



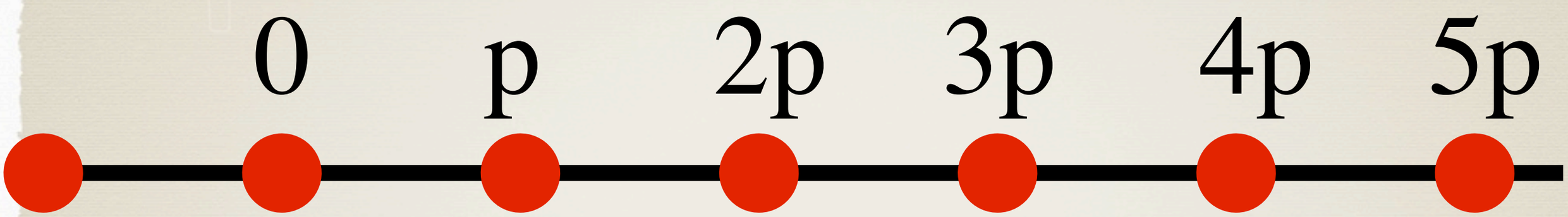


Approximate Integer GCD





Approximate Integer GCD



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

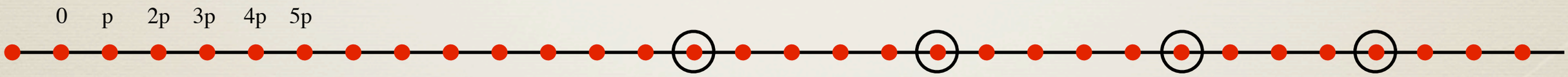
Approximate Integer GCD

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

Approximate Integer GCD

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

Approximate Integer GCD

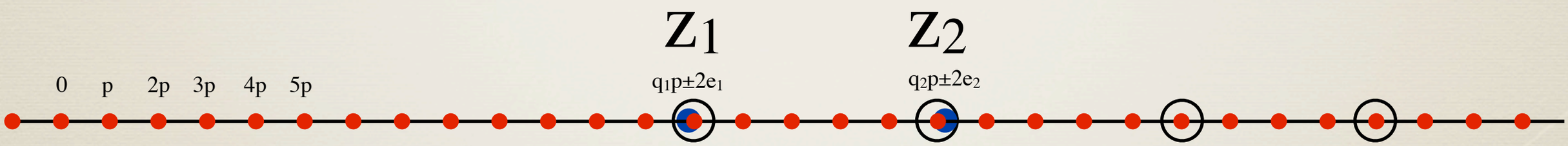


$$GCD(x_1, x_2, x_3, x_4) = p$$

Approximate Integer GCD

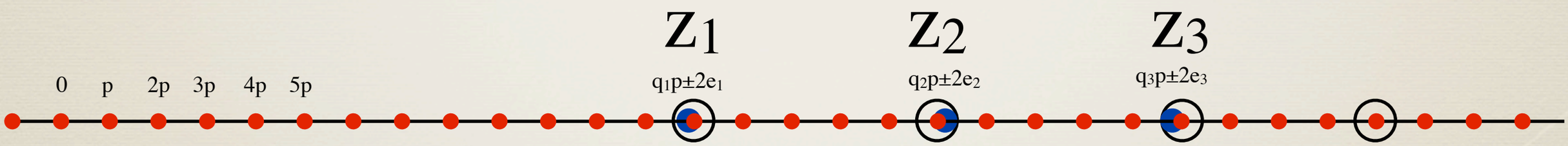
$$\mathrm{GCD}(x_1, x_2, x_3, x_4) = p$$

Approximate Integer GCD



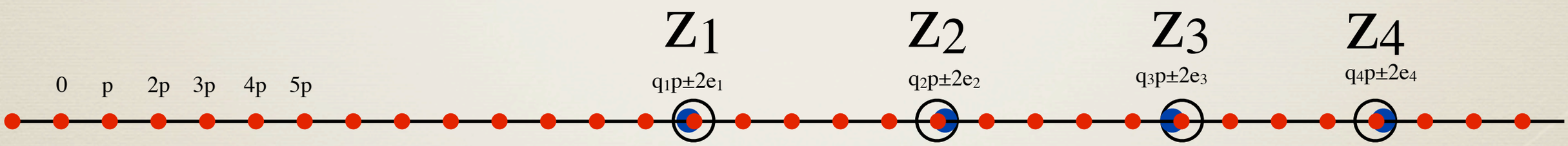
$$\mathrm{GCD}(x_1, x_2, x_3, x_4) = p$$

Approximate Integer GCD



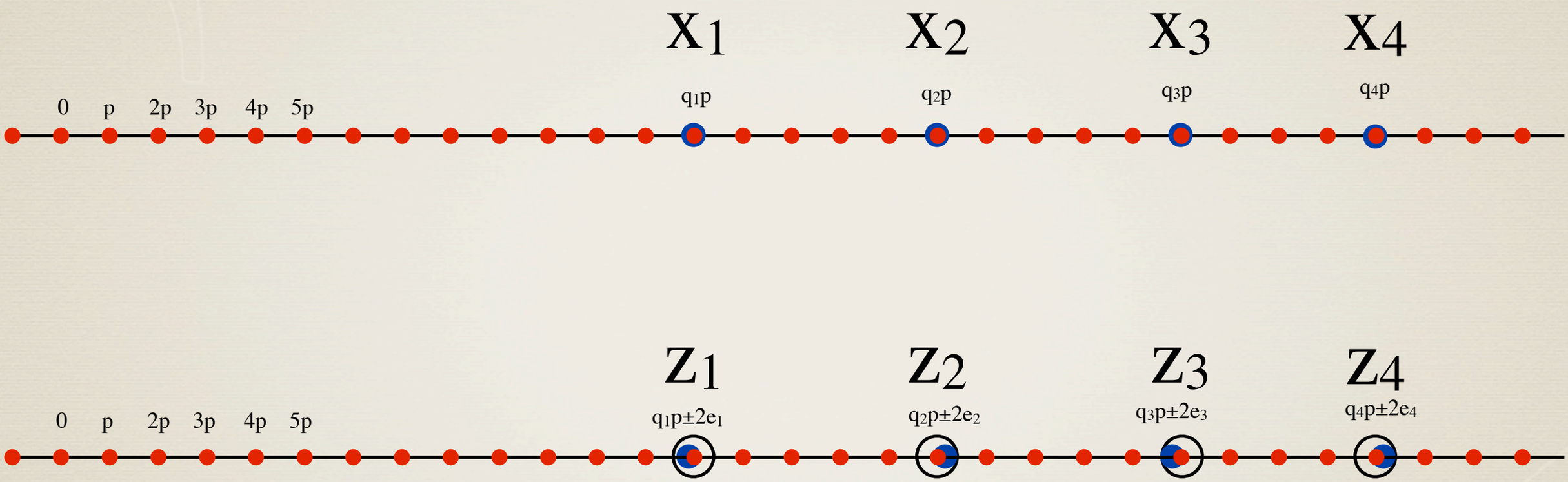
$$\mathrm{GCD}(x_1, x_2, x_3, x_4) = p$$

Approximate Integer GCD



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

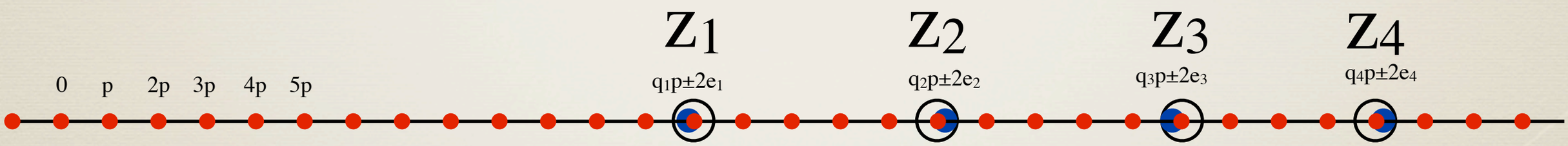
Approximate Integer GCD



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

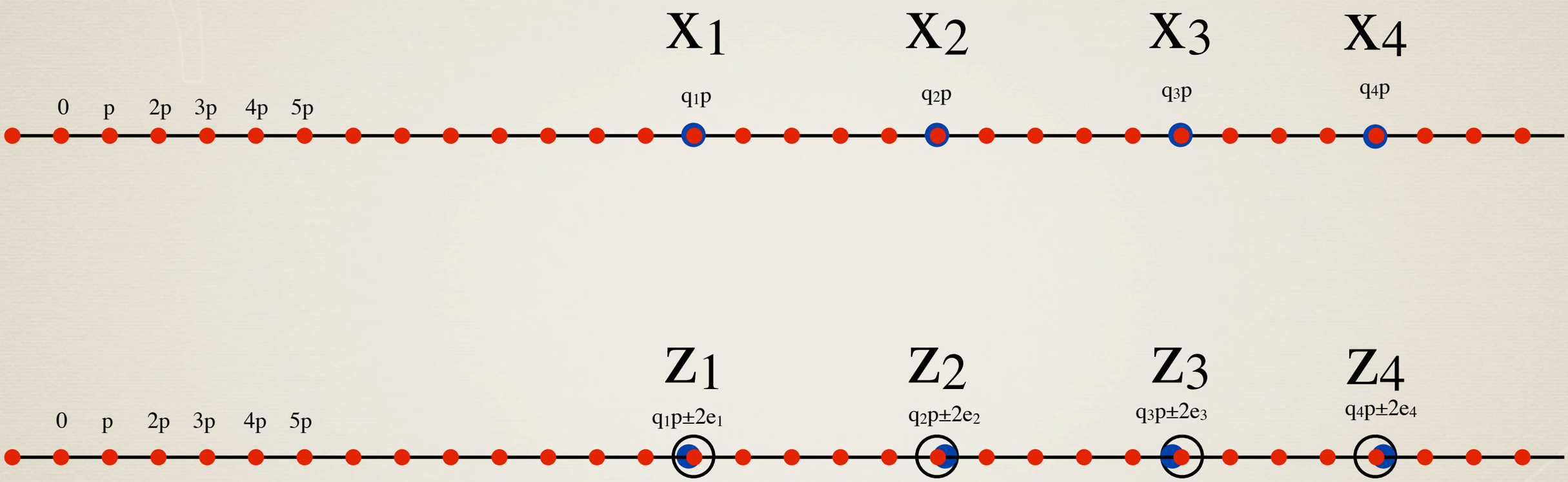
$$GCD(z_1,z_2,z_3,z_4) = 1$$

Approximate Integer GCD



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

Approximate Integer GCD



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

AIGCD : find p from z_1,z_2,z_3,z_4 ?

Approximate Integer GCD

z_1 z_2 z_3 . . . z_{k-1} z_k

z_0

Approximate Integer GCD

z_1 z_2 z_3 . . . z_{k-1} z_k

z_0

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

Approximate Integer GCD

z_1 z_2 z_3 . . . z_{k-1} z_k

z_0

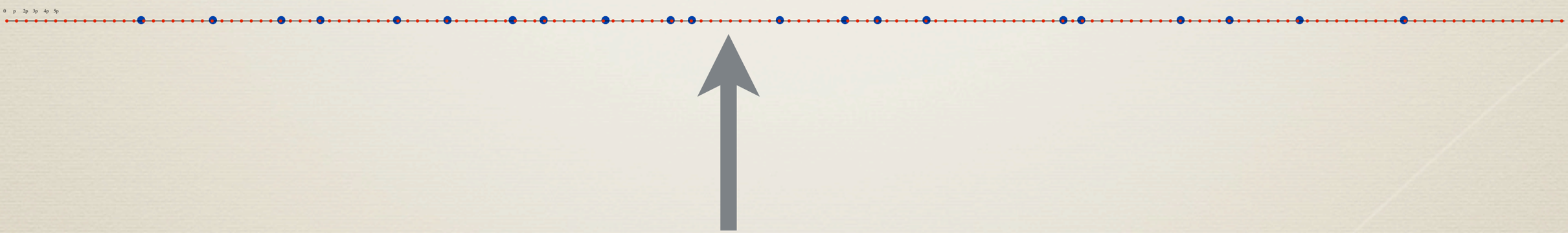
$$\sum_{1 \le i \le k} s_i z_i \bmod z_0$$

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

Approximate Integer GCD

$z_1 \quad z_2 \quad z_3 \quad . \quad . \quad . \quad z_{k-1} \quad z_k$

z_0



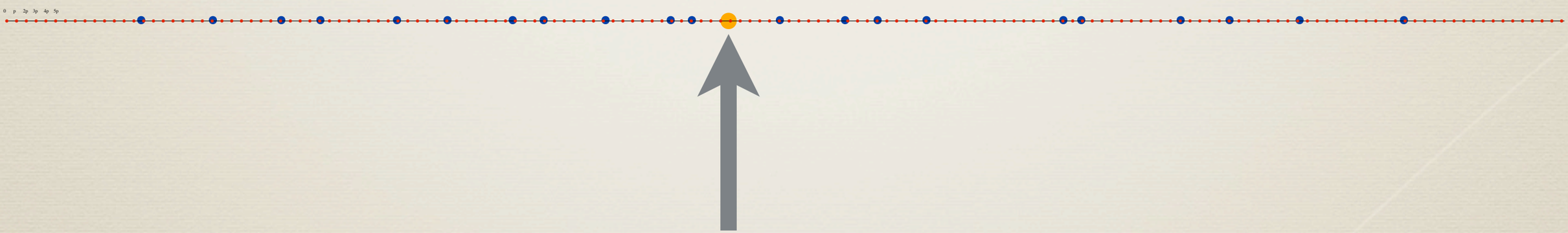
$$\sum_{1 \le i \le k} s_i z_i \bmod z_0$$

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

Approximate Integer GCD

$z_1 \quad z_2 \quad z_3 \quad . \quad . \quad . \quad z_{k-1} \quad z_k$

z_0



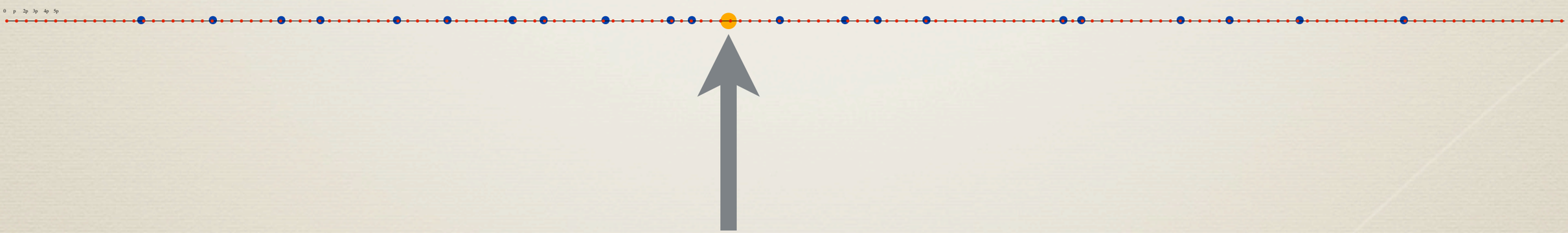
$$\sum_{1 \le i \le k} s_i z_i \bmod z_0$$

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

Approximate Integer GCD

$z_1 \quad z_2 \quad z_3 \quad . \quad . \quad . \quad z_{k-1} \quad z_k$

z_0



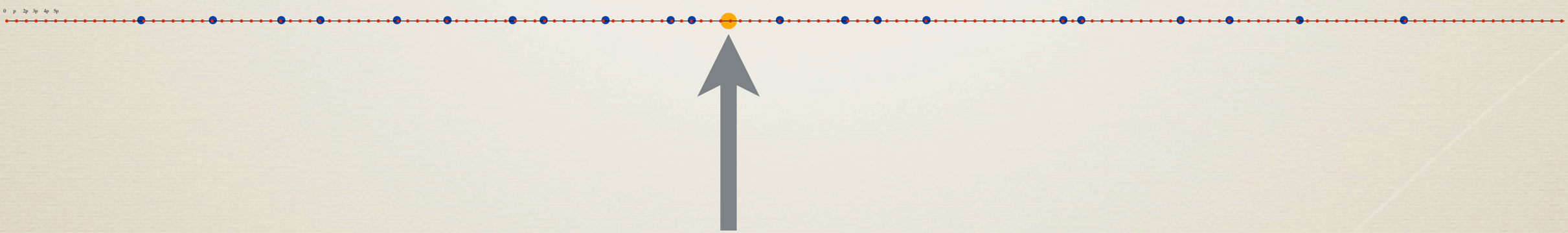
$$\sum_{1 \le i \le k} s_i z_i \bmod z_0 \approx \sum_{1 \le i \le k} s_i x_i \bmod x_0$$

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

Approximate Integer GCD

$z_1 \quad z_2 \quad z_3 \quad . \quad . \quad . \quad z_{k-1} \quad z_k$

z_0



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

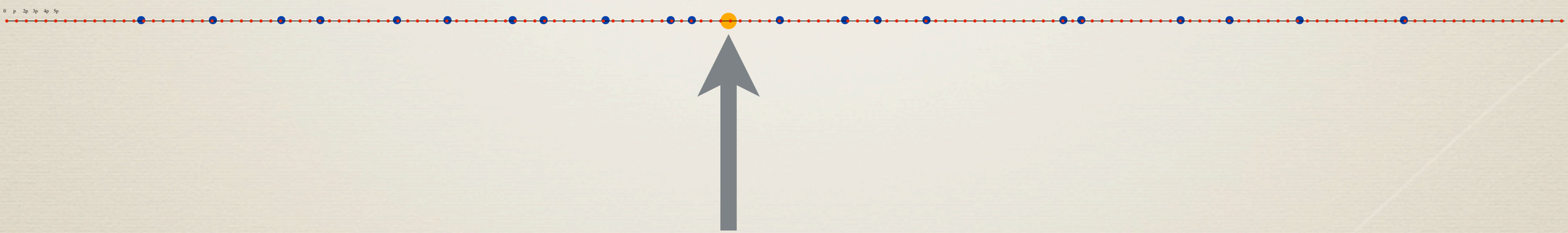
$$s_i \in \{0,1\}$$

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

Approximate Integer GCD

z_1 z_2 z_3 . . . z_{k-1} z_k

z_0



$$\left| \sum_{1 \le i \le k} s_i z_i \bmod z_0 - \left(\sum_{1 \le i \le k} s_i q_i \bmod q_0 \right) \times p \right| \le 4k|e_{max}|$$

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

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

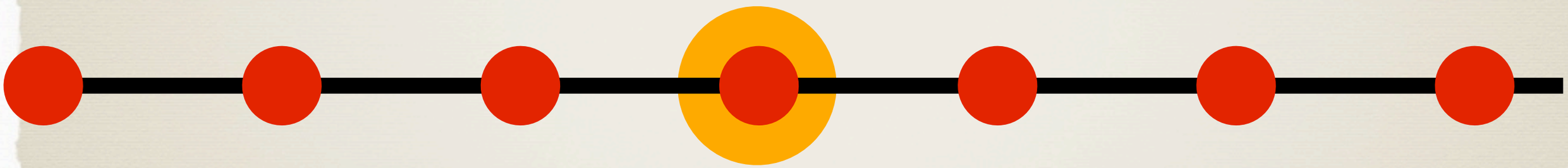
$$\Omega(s)=\sum_{1\leq i\leq k} s_i z_i \bmod z_0$$

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

Approximate Integer GCD

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

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

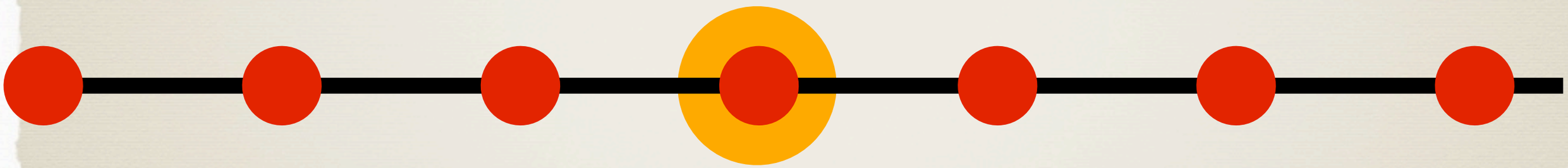


$$|e_{max}| \leq \partial \ll p/8k$$

Approximate Integer GCD

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

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



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

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

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

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

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

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

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

AIGCD encryption

SK : p

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

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

$$enc(b) = \Omega(s)+2e+b$$
$$s \in_U \{0,1\}^n$$
$$e \in_U [-\partial' \ldots +\partial']$$

$$dec(c) = c - p[c/p] \bmod 2$$
$$= \text{parity of error}$$

Oblivious Transfer from Weakly Random-Self-Reducible Encryption

by Claude Crépeau
School of Computer Science
McGill University
joint work with Raza Ali Kazmi

(0)
FOREWORD

2008

Oblivious Transfer via McEliece's PKC and Permuted Kernels

K. Kobara[1], Kirill Morozov[1] and R. Overbeck[2]

[1] RCIS, AIST
{k-kobara,kirill.morozov}@aist.go.jp
[2] TU-Darmstadt,
Department of Computer Science,
Cryptography and Computer Algebra Group.
overbeck@cdc.informatik.tu-darmstadt.de

Oblivious Transfer Based on the McEliece Assumptions

Rafael Dowsley[1], Jeroen van de Graaf[2], Jörn Müller-Quade[3],
and Anderson C.A. Nascimento[1]

2008

Oblivious Transfer via McEliece's PKC and Permuted Kernels

K. Kobara[1], Kirill Morozov[1] and R. Overbeck[2]

[1] RCIS, AIST
{k-kobara,kirill.morozov}@aist.go.jp
[2] TU-Darmstadt,
Department of Computer Science,
Cryptography and Computer Algebra Group.
overbeck@cdc.informatik.tu-darmstadt.de

Oblivious Transfer Based on the McEliece Assumptions

Rafael Dowsley[1], Jeroen van de Graaf[2], Jörn Müller-Quade[3],
and Anderson C.A. Nascimento[1]

2008

Oblivious Transfer via McEliece's PKC and Permuted Kernels

K. Kobara[1], Kirill Morozov[1] and R. Overbeck[2]

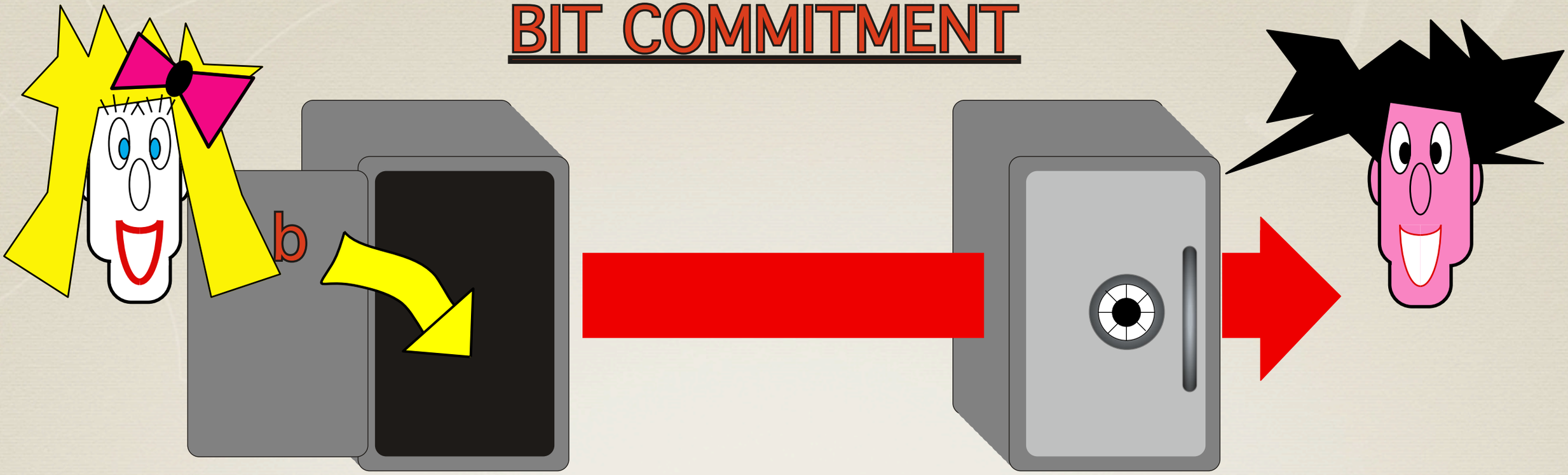
[1] RCIS, AIST
{k-kobara,kirill.morozov}@aist.go.jp
[2] TU-Darmstadt,
Department of Computer Science,
Cryptography and Computer Algebra Group.
overbeck@cdc.informatik.tu-darmstadt.de

Oblivious Transfer Based on the McEliece Assumptions

Rafael Dowsley[1], Jeroen van de Graaf[2], Jörn Müller-Quade[3],
and Anderson C.A. Nascimento[1]

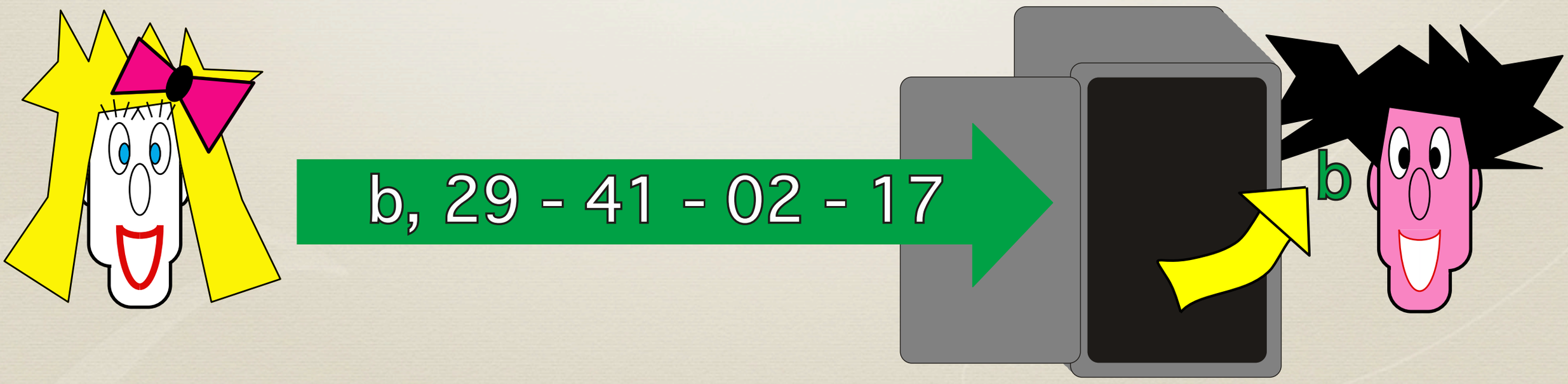
(1)
Two-Party
Cryptographic Protocols

BIT COMMITMENT

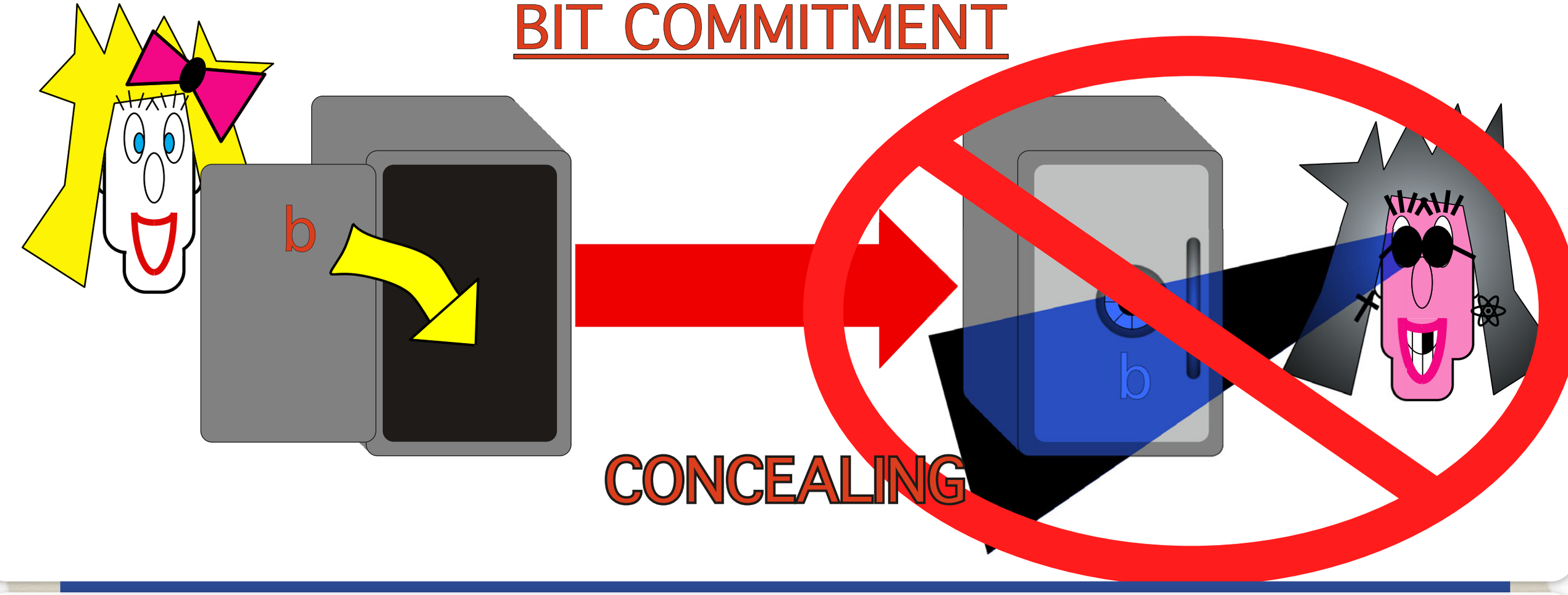


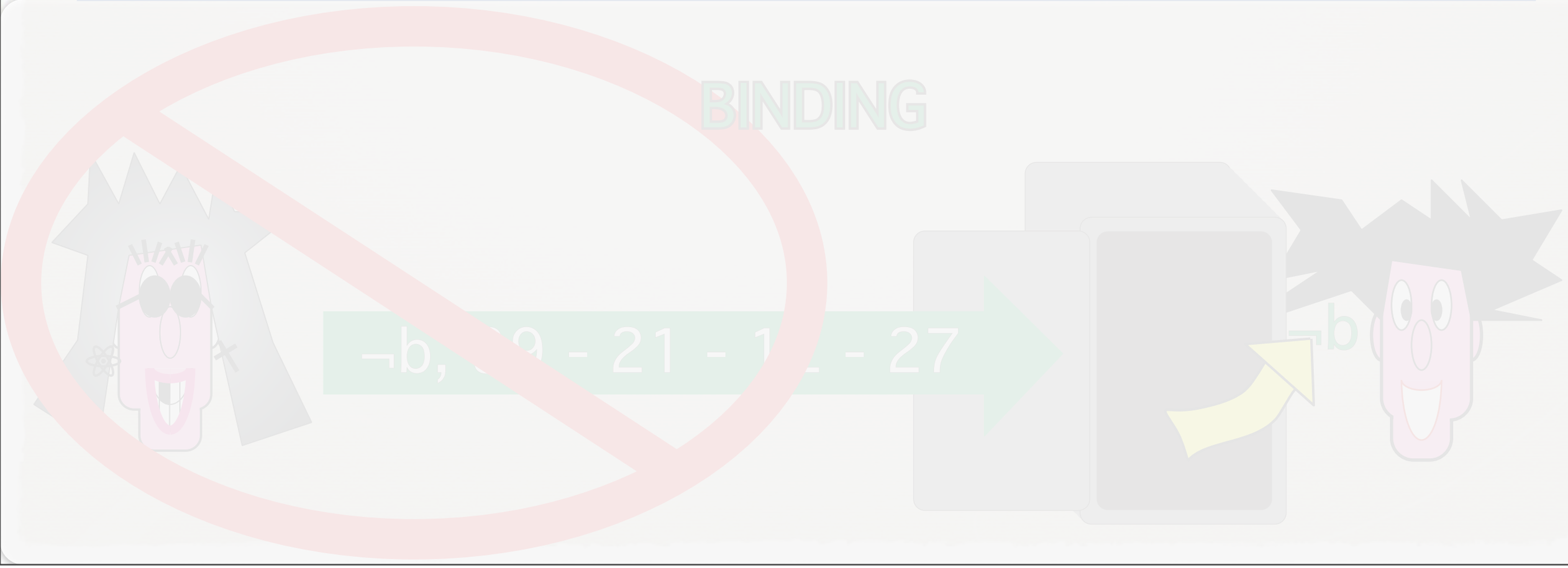
COMMIT

UNVEIL

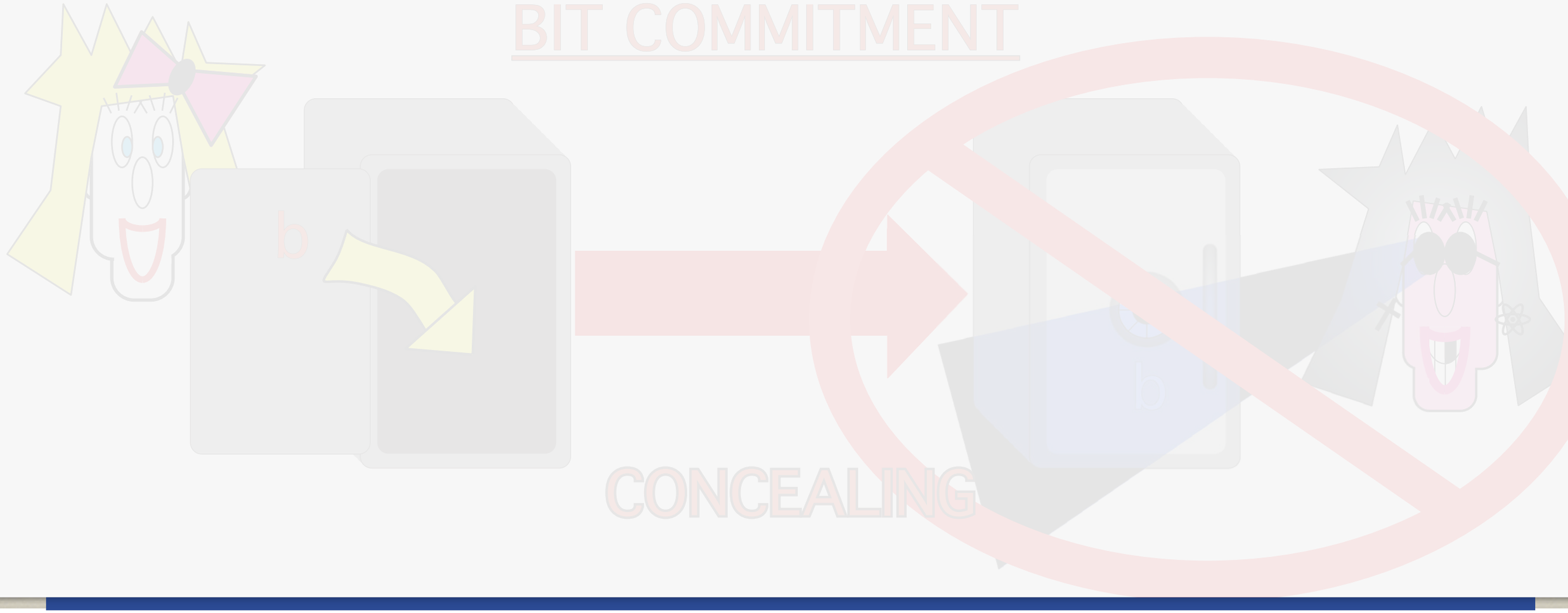


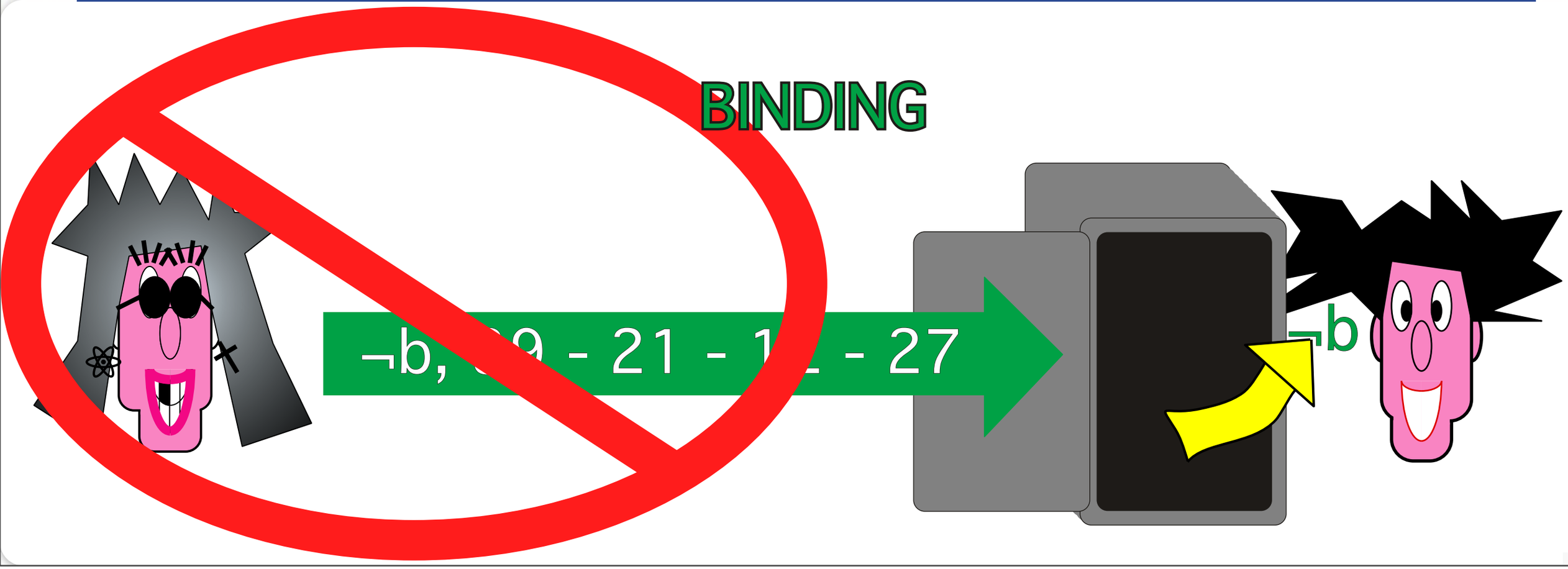
BIT COMMITMENT



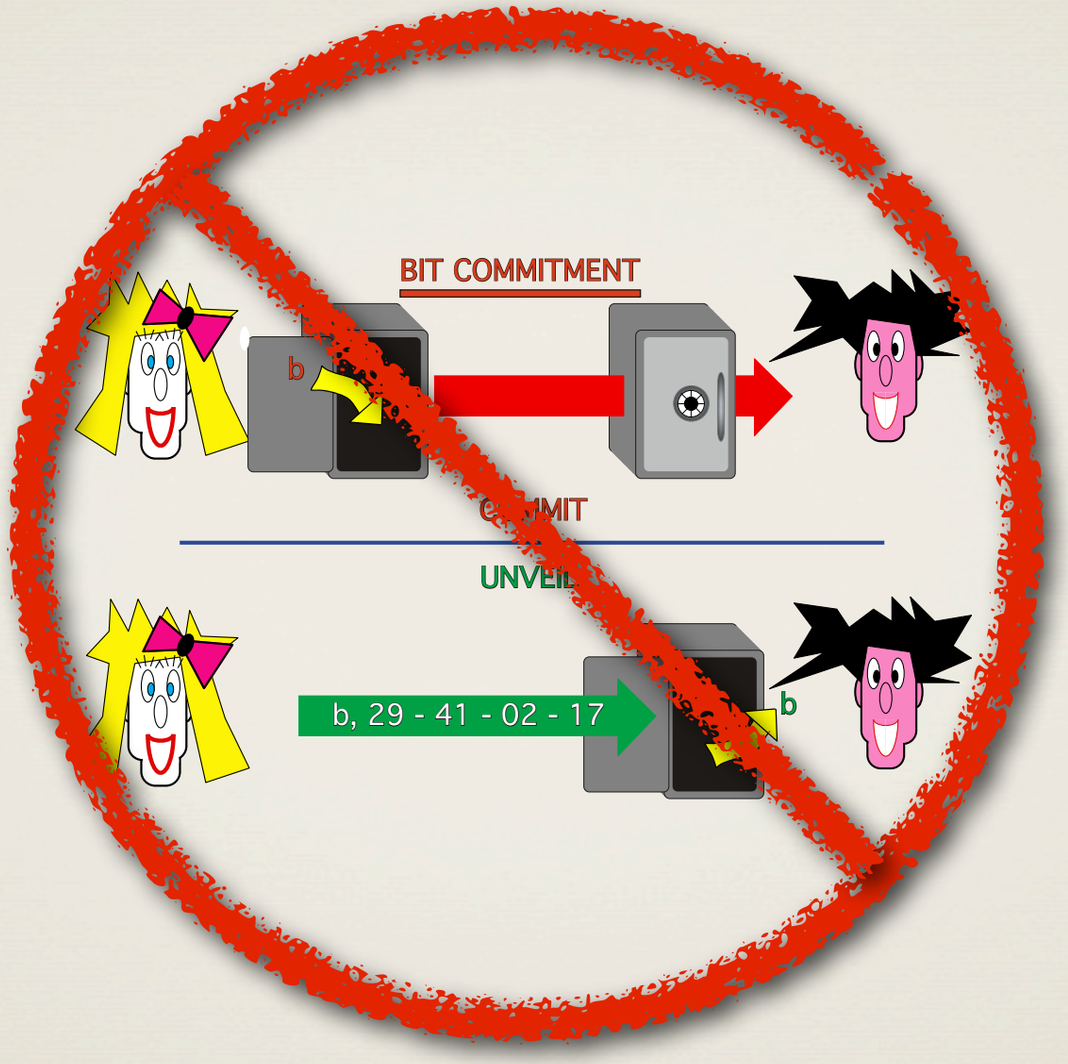


BIT COMMITMENT



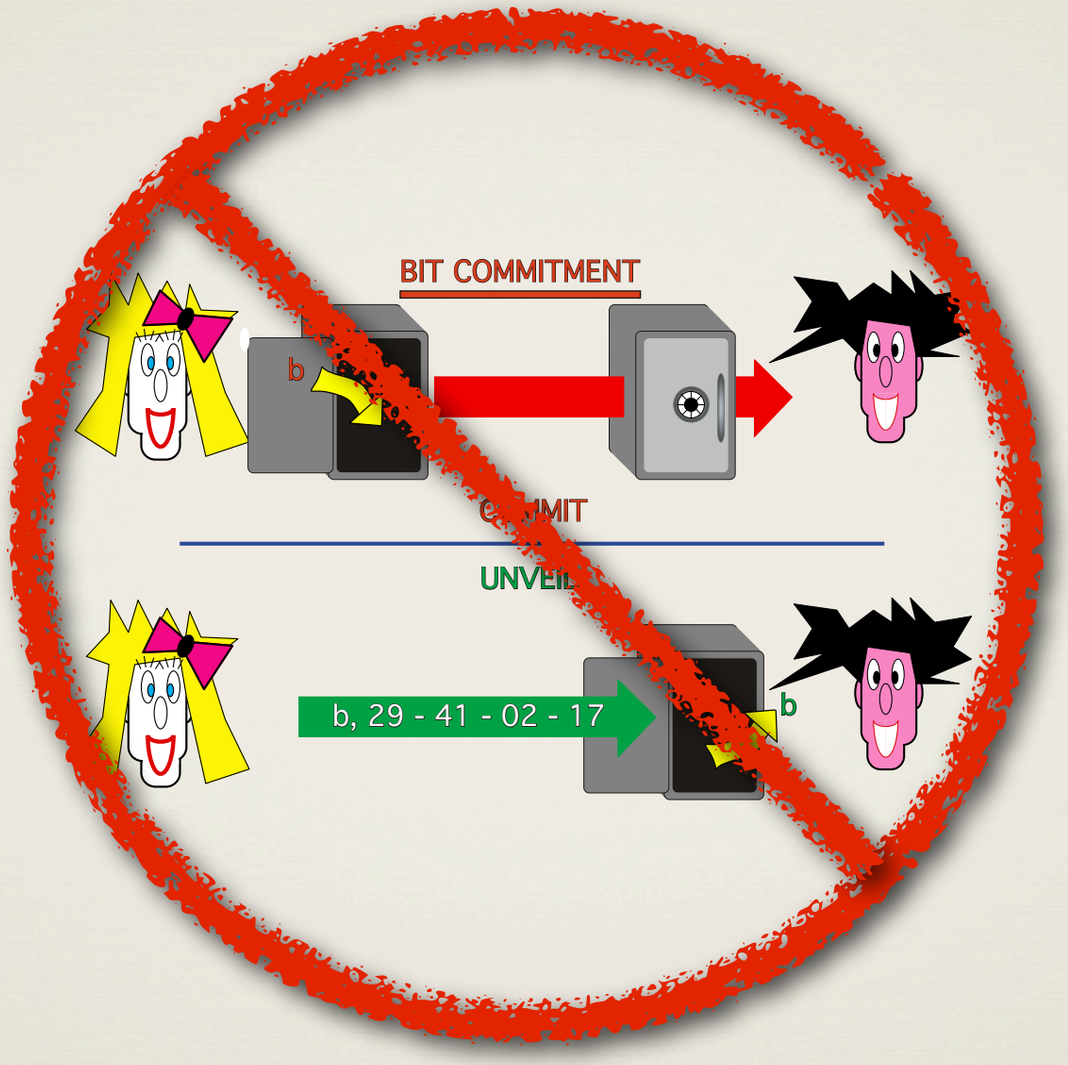


Classical



Folklore

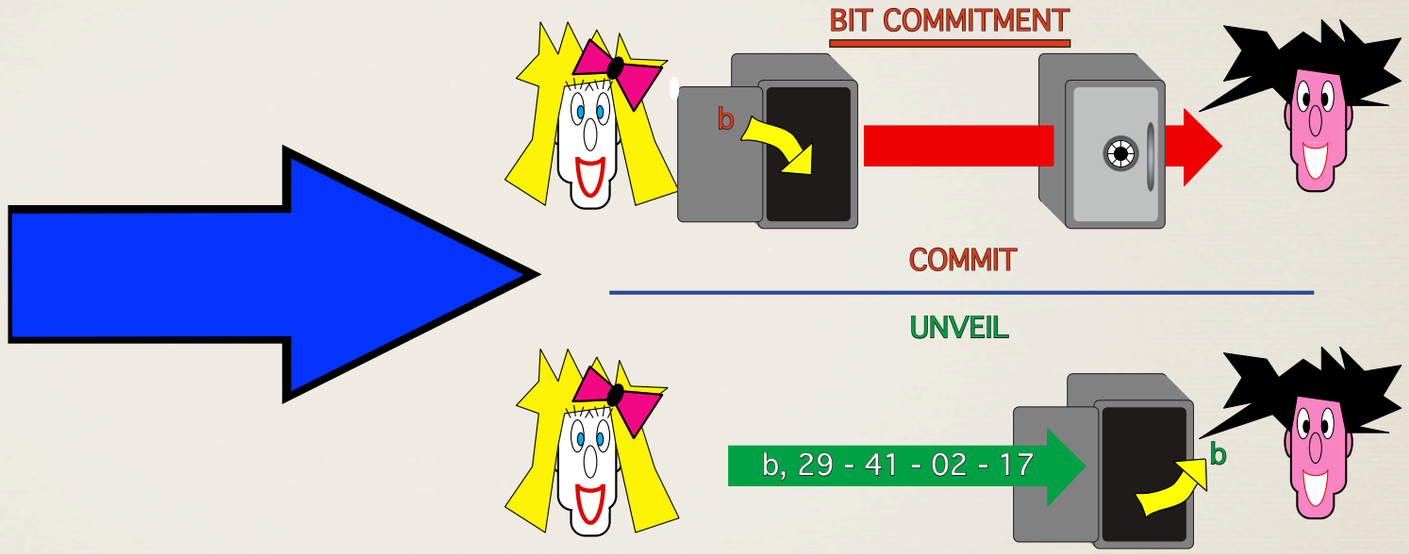
Quantum



Mayers, Lo-Chau

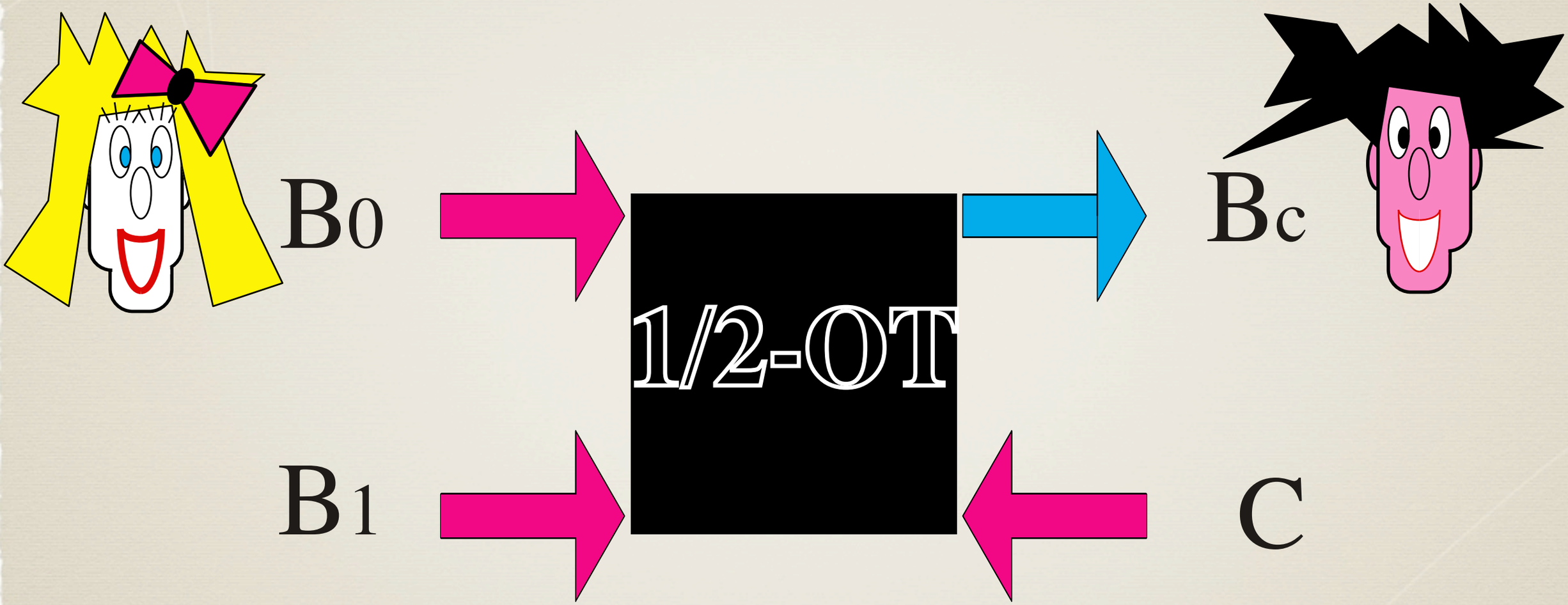
Classical

One-Way Function

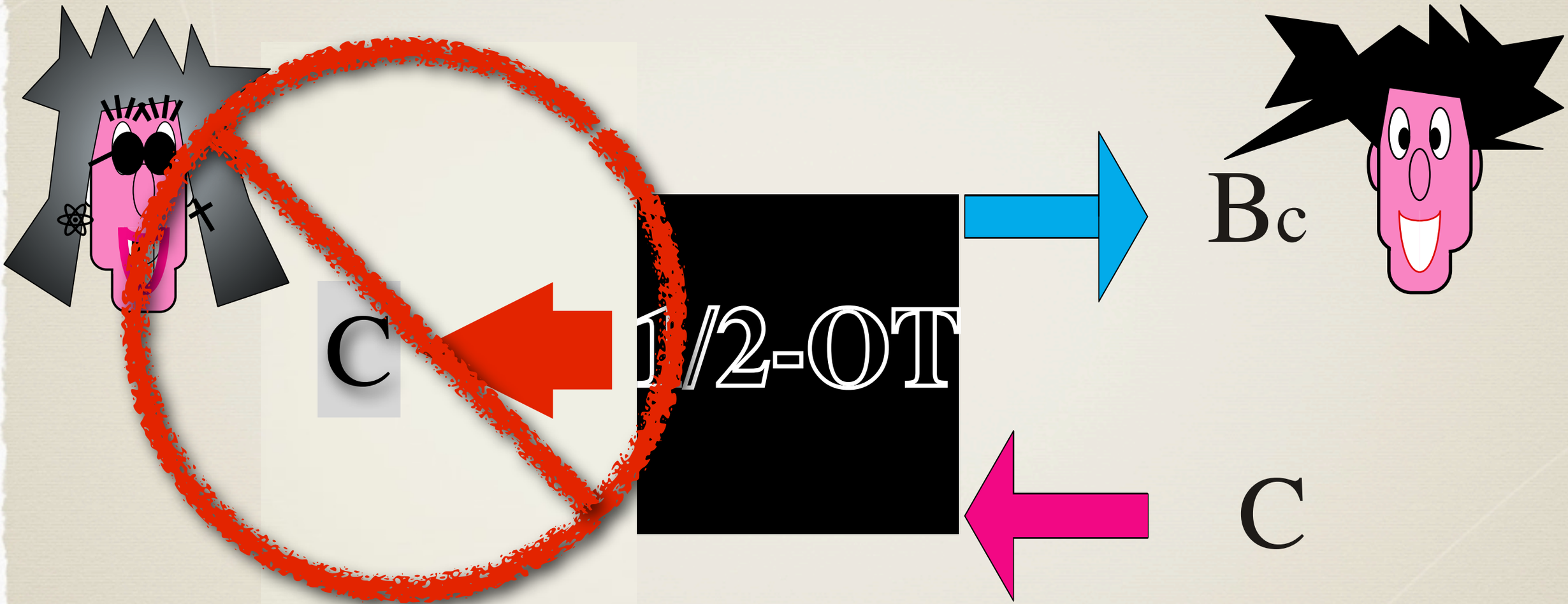


[HILL90]

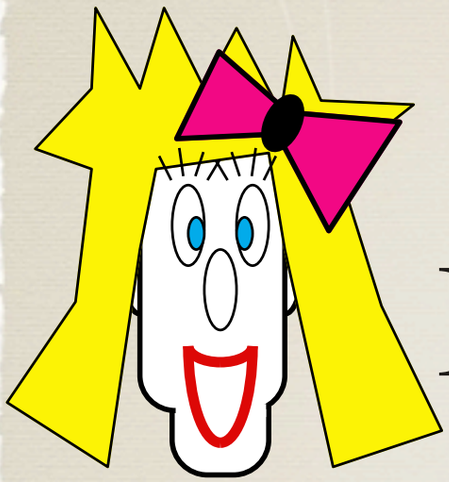
Oblivious Transfer



Oblivious Transfer

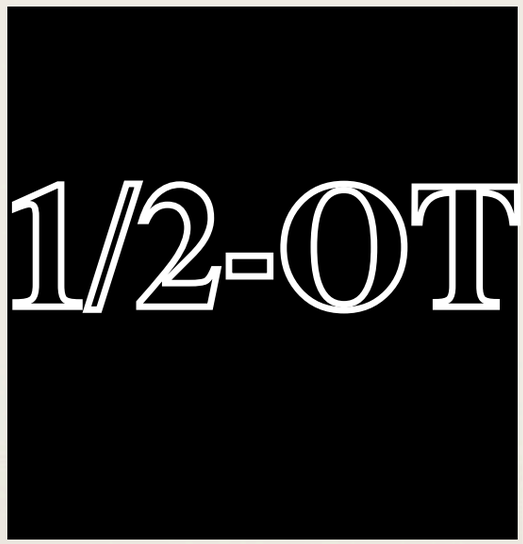


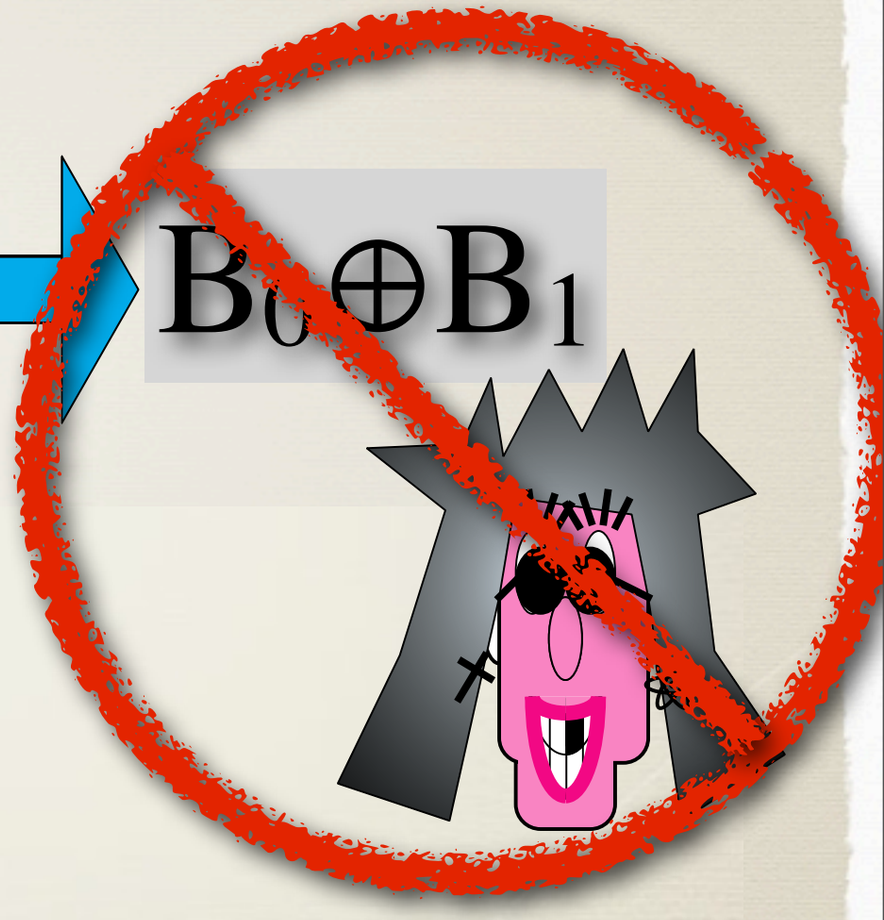
Oblivious Transfer



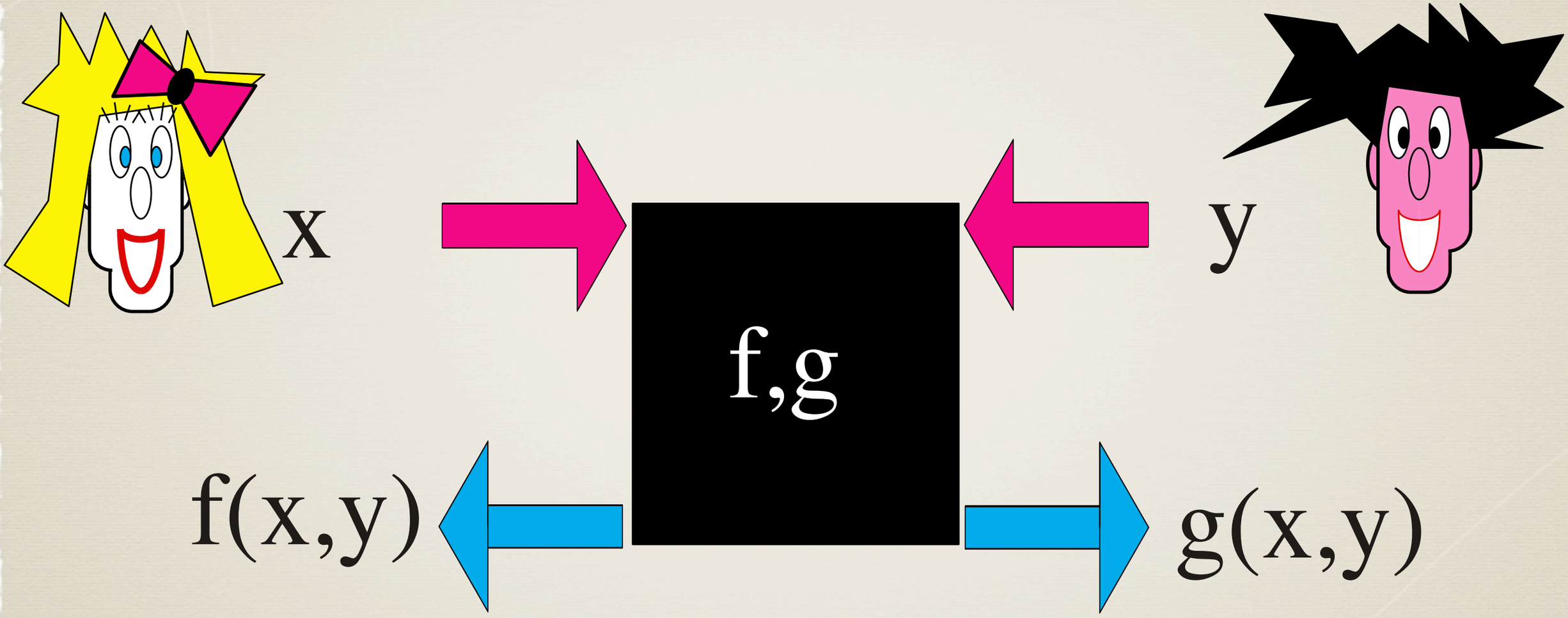
B_0

B_1

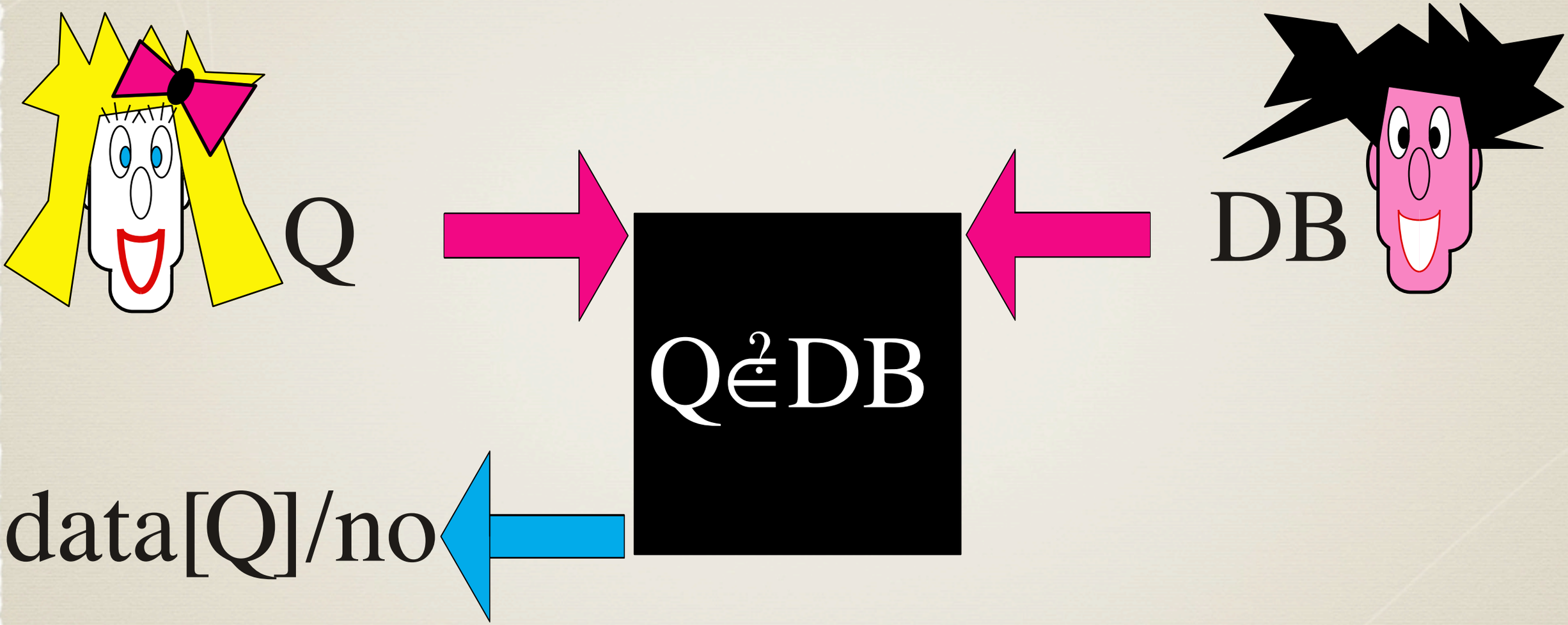




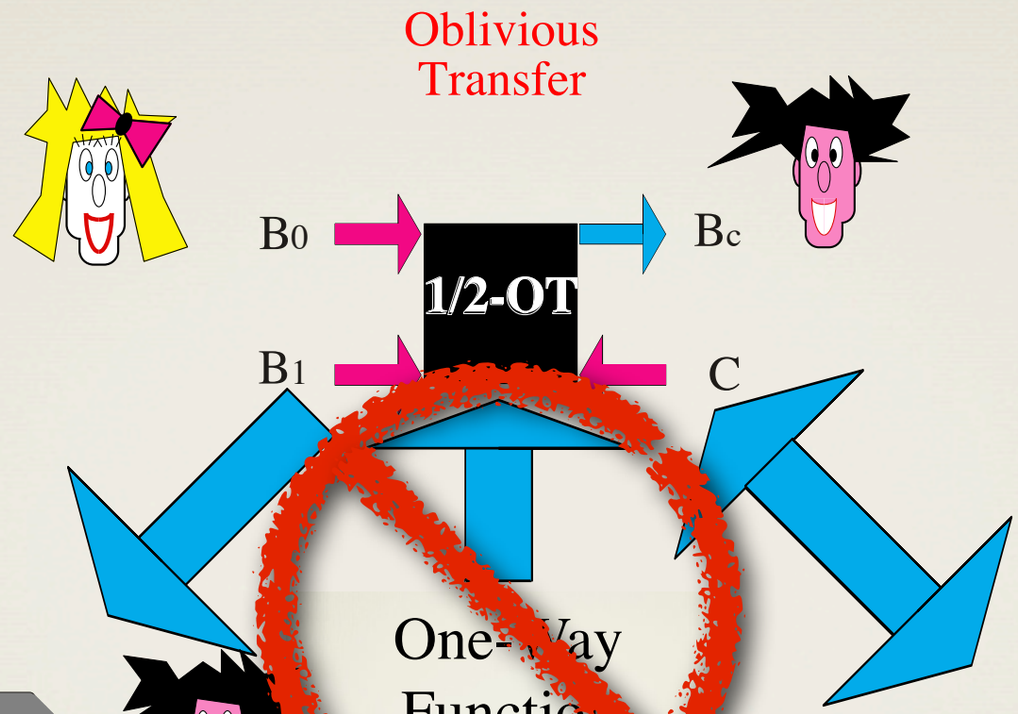
Oblivious Function Evaluation

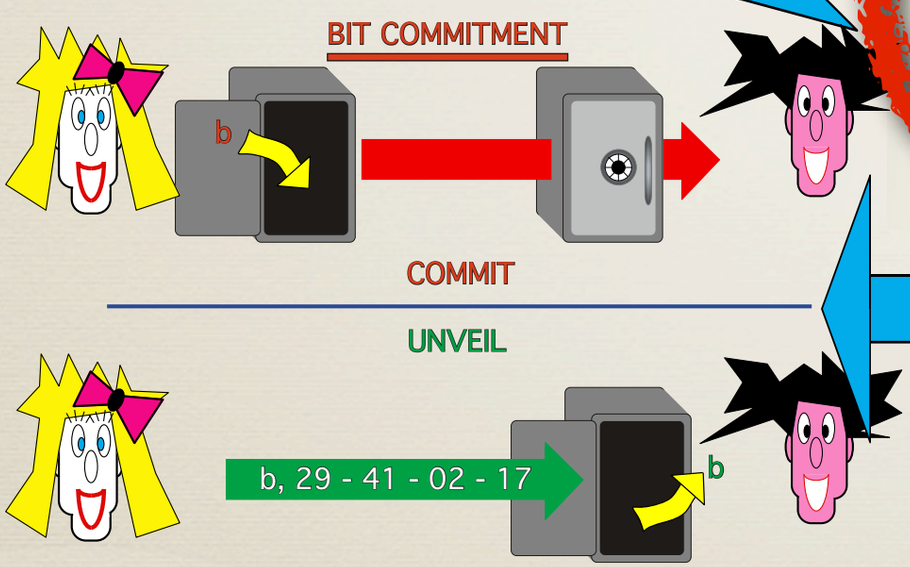


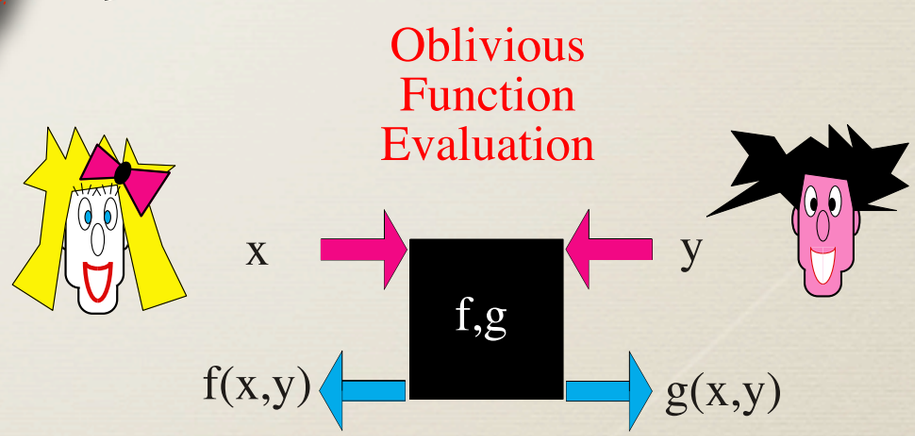
Oblivious Database Query



Classically

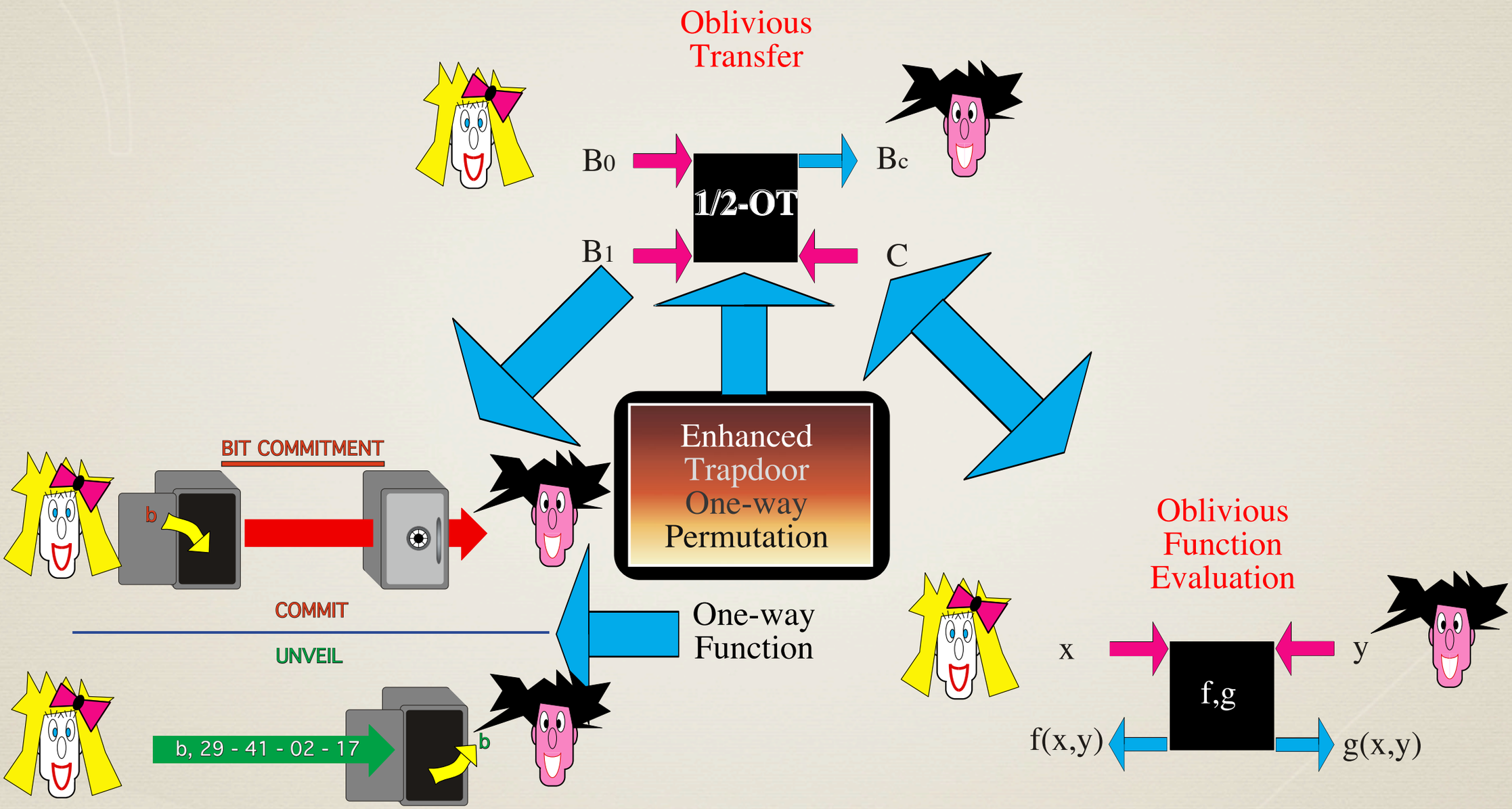






(2)
Secure OT
Implementations

Classically

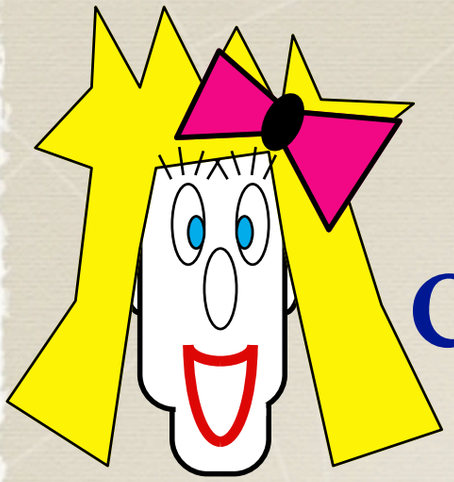


[EGL85]
[GMW87]

* INGREDIENTS:

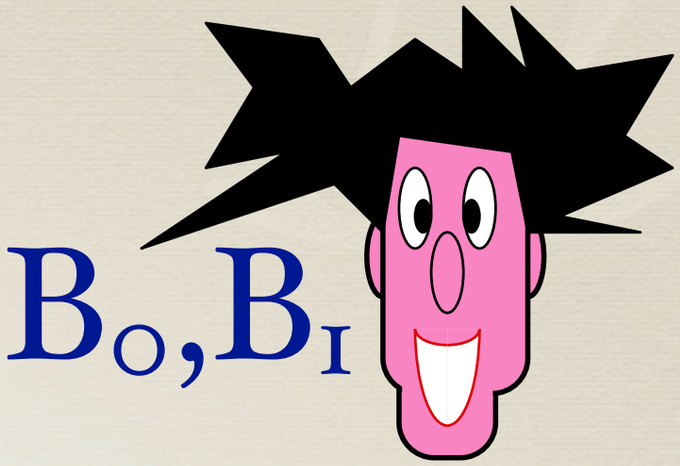
Public-Key Cryptosystem (TOWP)
(enc, dec)

and hard-core predicate π



c=o

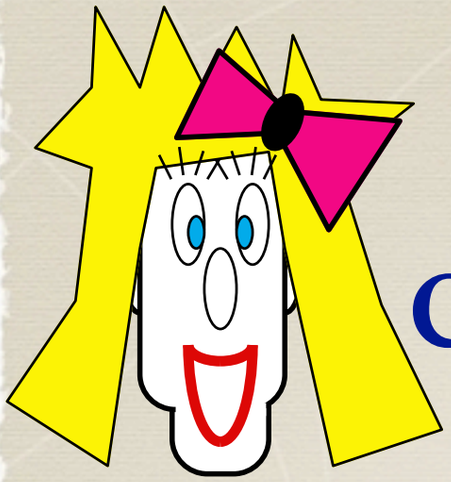
[EGL85]
[GMW87]



B_o, B_I

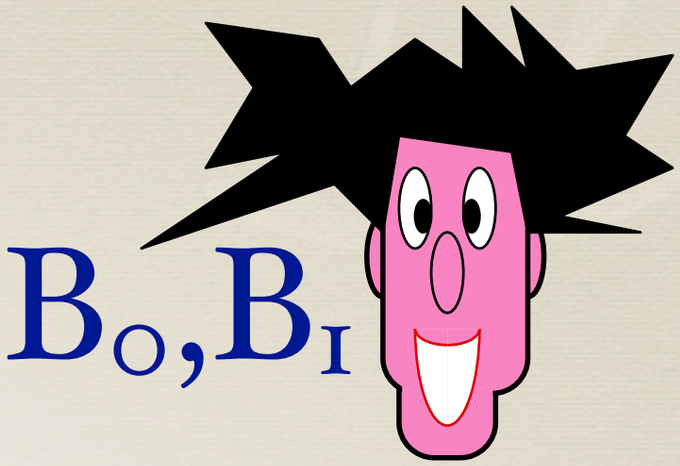
$enc_B(®_o)$ ⇒ U_o

$®_I$ ⇒ U_I



c=o

[EGL85]
[GMW87]



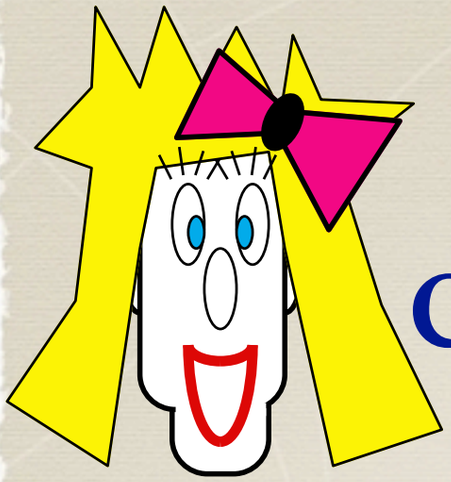
B_o, B_I

$enc_B(®_o)$ ⇒ U_o

$®_I$ ⇒ U_I

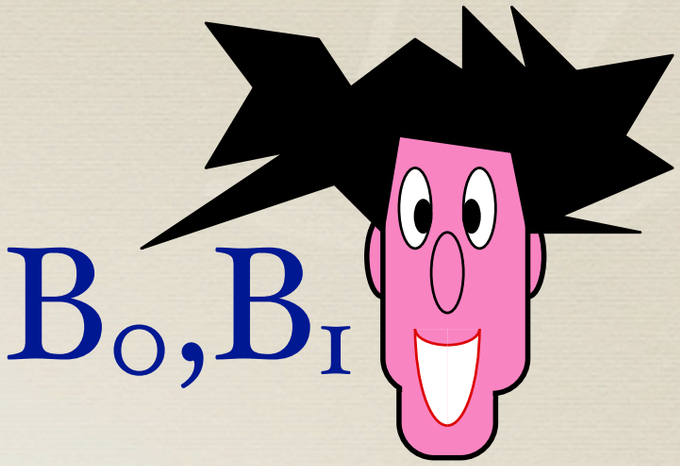
$Z_o \Leftarrow \pi(dec_B(U_o)) \oplus B_o$

$Z_I \Leftarrow \pi(dec_B(U_I)) \oplus B_I$



c=o

[EGL85]
[GMW87]



B_o, B_I

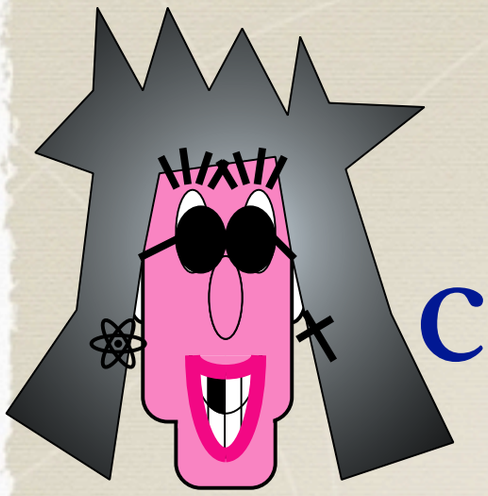
$enc_B(®_o)$ ⇒ U_o

$®_I$ ⇒ U_I

Z_o ⇐ $\pi(dec_B(U_o)) \oplus B_o$

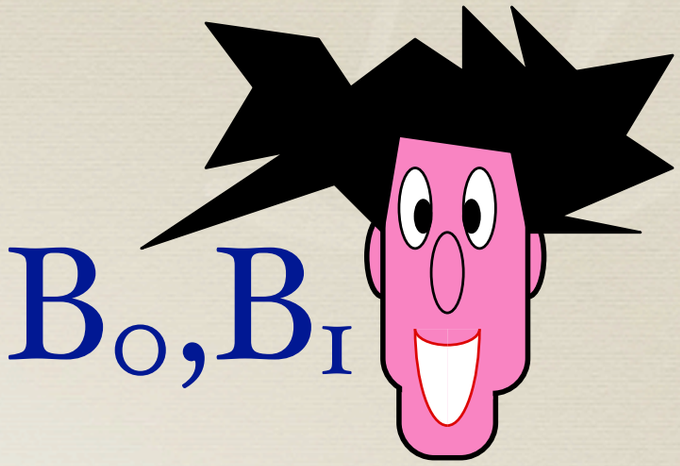
Z_I ⇐ $\pi(dec_B(U_I)) \oplus B_I$

$B_o = \pi(®_o) \oplus Z_o$



c=?

[EGL85]
[GMW87]



B_O, B_I

$enc_B(®_O)$ ⇒ U_O

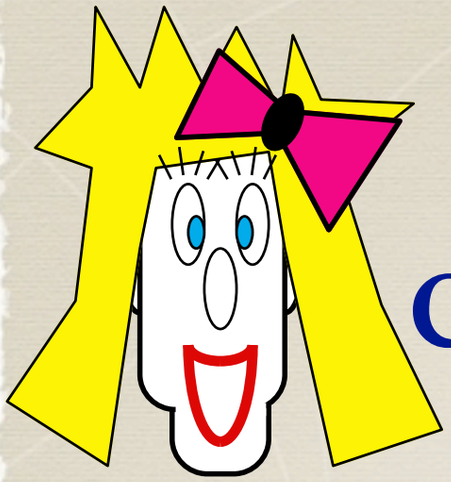
$enc_B(®_I)$ ⇒ U_I

Z_O ⇐ $\pi(dec_B(U_O)) \oplus B_O$

Z_I ⇐ $\pi(dec_B(U_I)) \oplus B_I$

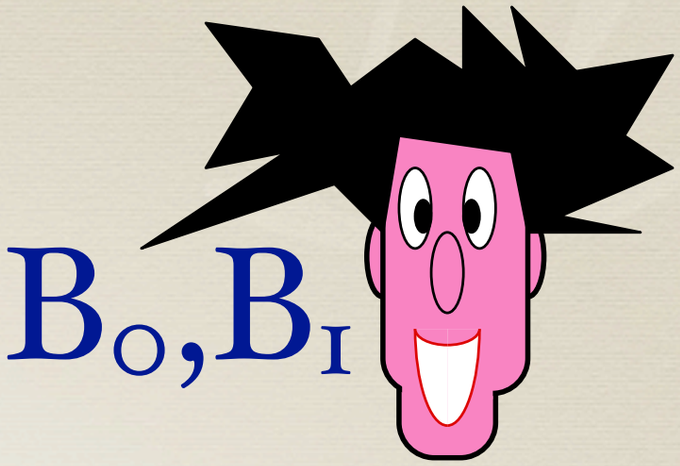
$B_O = \pi(®_O) \oplus Z_O$

$B_I = \pi(®_I) \oplus Z_I$



c=O

[GMW87]



B_O, B_I

$enc_B(®_O)$ ⇒ U_O

$®_I$ ⇒ U_I

Z_O ⇐ $\pi(dec_B(U_O)) \oplus B_O$

Z_I ⇐ $\pi(dec_B(U_I)) \oplus B_I$

Use ZK proofs to make sure both parties follow protocol.

[Goldreich02]

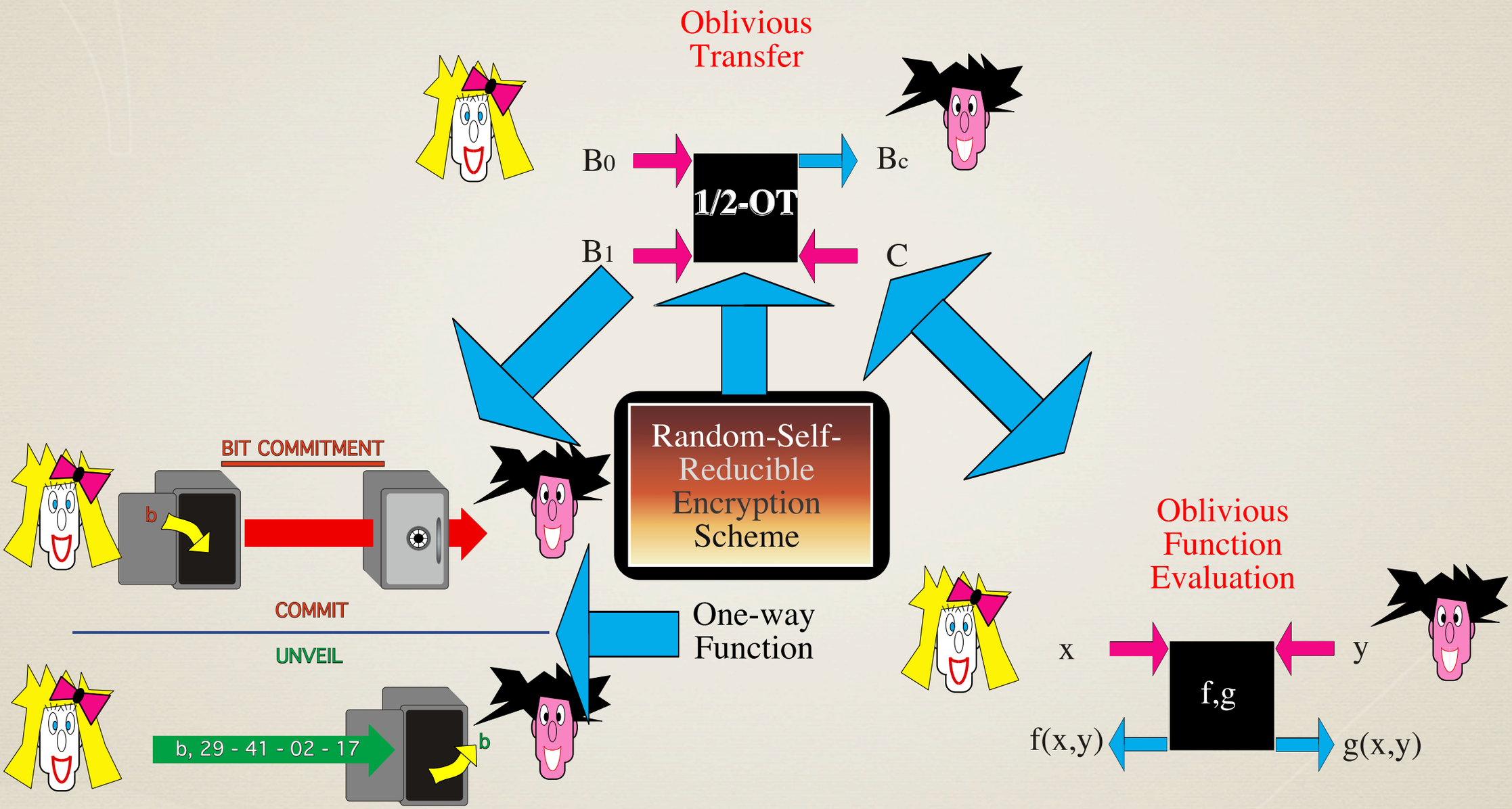
Definition (*Enhanced* TOWP)

A TOWP **enc** is *enhanced* if there exists a PPT algorithm to select random elements from the image of **enc** without knowledge of the corresponding pre-image.

OT Implementations

- RSA
- Factoring
- Elliptic Curves
- ElGammal

Classically



[GM84]

[BCR86]

[GM84]
[BCR86]

* INGREDIENTS:
  Public-Key Cryptosystem (enc, dec)
  and hard-core predicate π

[GM84]
[BCR86]

* INGREDIENTS:
  Public-Key Cryptosystem (enc, dec)
  and hard-core predicate π

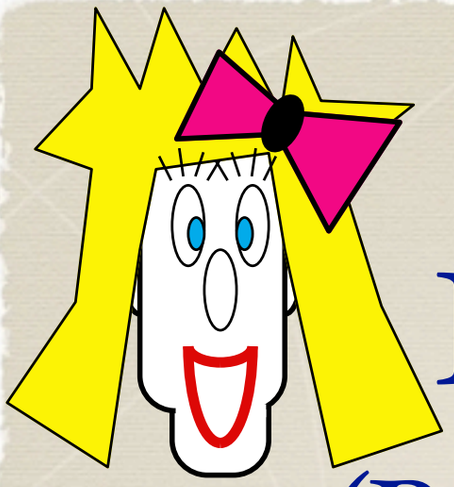
* Define $enc' = enc(\pi^{-1}(\bullet))$, $dec' = \pi(dec(\bullet))$
  we need (enc′,dec′) be semantically sec.

[GM84]
[BCR86]

* INGREDIENTS:
  Public-Key Cryptosystem (enc, dec)
  and hard-core predicate π

* Define $enc' = enc(\pi^{-1}(\bullet))$, $dec' = \pi(dec(\bullet))$
  we need (enc′,dec′) be semantically sec.

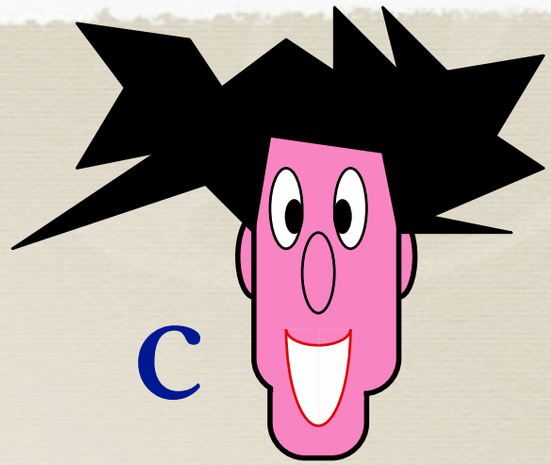
* We also need (enc,dec) to be RSR.



B_O, B_I

[GM84]

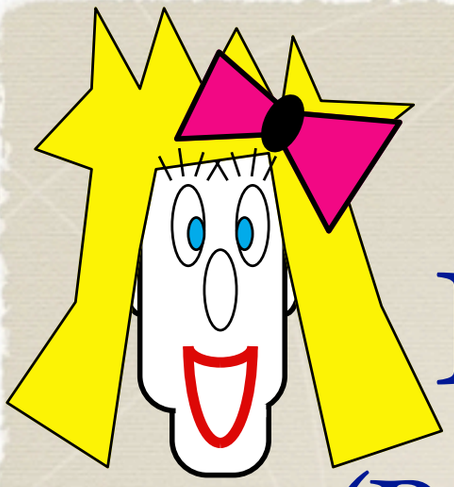
[BCR86]



c

$m_O = \pi^{-1}(B_O)$

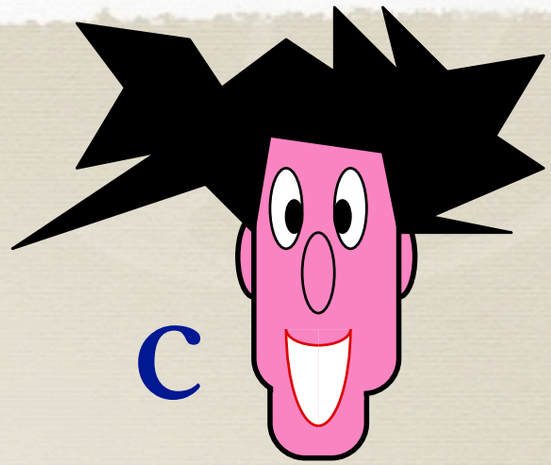
$m_I = \pi^{-1}(B_I)$



B_O, B_I

[GM84]

[BCR86]



c

$m_O = \pi^{-1}(B_O)$

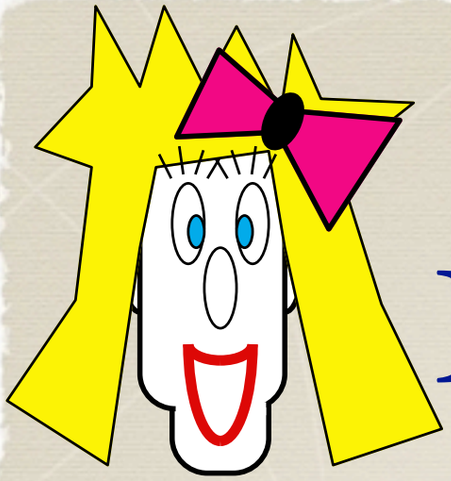
$enc_A(m_O)$

U_O

$m_I = \pi^{-1}(B_I)$

$enc_A(m_I)$

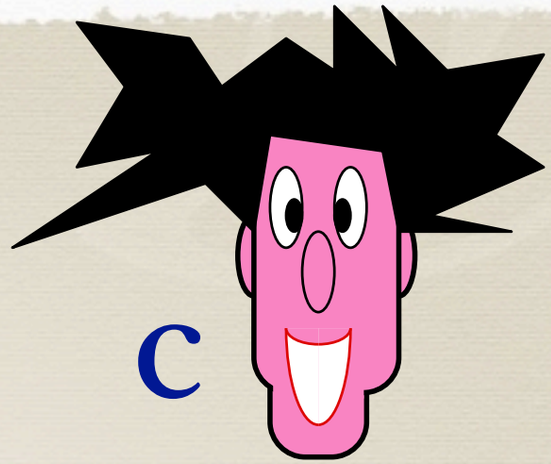
U_I



B_O, B_I

[GM84]

[BCR86]



c

$enc_A(m_O)$ → U_O

$enc_A(m_I)$ → U_I

Z ← $RE_A(®, U_C)$

Random Self-Reducible

[AL83]

[3] D. Angluin and D. Lichtenstein, "Provable Security of Cryptosystems: a Survey", Technical Report TR-288, Yale University, October 1983.

✳ **Definition (RSR cryptosystem)**

A cryptosystem (enc,dec) is *Random Self-Reducible* if there exists a pair of PPT algorithms (RE,RD) such that for all ®,m (re-encrypt,re-decrypt)

RD(®,dec(RE(®,enc(m))))=m

and

RE(®,enc(m)) is a uniform ciphertext

when ® is uniform.

[AL83]

[3] D. Angluin and D. Lichtenstein, "Provable Security of Cryptosystems: a Survey", Technical Report TR-288, Yale University, October 1983.

✳ **Definition (RSR cryptosystem)**

A cryptosystem (enc,dec) is *Random Self-Reducible* if there exists a pair of PPT algorithms (RE,RD) such that for all ®,m (re-encrypt,re-decrypt)

RD(®,dec(RE(®,enc(m))))=m

and

RE(®,enc(m)) is a uniform ciphertext

when ® is uniform.

[AL83]

[3] D. Angluin and D. Lichtenstein, "Provable Security of Cryptosystems: a Survey", Technical Report TR-288, Yale University, October 1983.

✳ **Definition (RSR cryptosystem)**

A cryptosystem (enc,dec) is *Random Self-Reducible* if there exists a pair of PPT algorithms (RE,RD) such that for all ®,m (re-encrypt,re-decrypt)

RD(®,dec(RE(®,enc(m))))=m

and

RE(®,enc(m)) is a uniform ciphertext

when ® is uniform.

[AL83]

[3] D. Angluin and D. Lichtenstein, "Provable Security of Cryptosystems: a Survey", Technical Report TR-288, Yale University, October 1983.

✳ **Definition (RSR cryptosystem)**

A cryptosystem (enc,dec) is *Random Self-Reducible* if there exists a pair of PPT algorithms (RE,RD) such that for all ®,m (re-encrypt,re-decrypt)

$$RD(®,dec(RE(®,enc(m))))=m$$

and

RE(®,enc(m)) is a uniform ciphertext

when ® is uniform.

[AL83]

[3] D. Angluin and D. Lichtenstein, "Provable Security of Cryptosystems: a Survey", Technical Report TR-288, Yale University, October 1983.

✳ **Definition (RSR cryptosystem)**

A cryptosystem (enc,dec) is *Random Self-Reducible* if there exists a pair of PPT algorithms (RE,RD) such that for all ®,m (re-encrypt,re-decrypt)

$$RD(®,dec(RE(®,enc(m))))=m$$

and

RE(®,enc(m)) is a uniform ciphertext

when ® is uniform.

[AL83]

[3] D. Angluin and D. Lichtenstein, "Provable Security of Cryptosystems: a Survey", Technical Report TR-288, Yale University, October 1983.

✳ **Definition (RSR cryptosystem)**

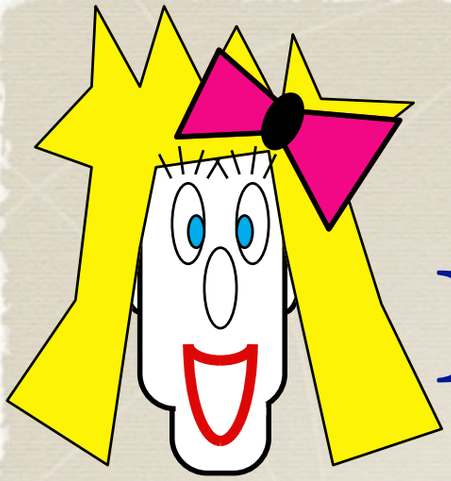
A cryptosystem (enc,dec) is *Random Self-Reducible* if there exists a pair of PPT algorithms (RE,RD) such that for all ®,m (re-encrypt,re-decrypt)

RD(®,dec(RE(®,enc(m))))=m

and

RE(®,enc(m)) is a uniform ciphertext

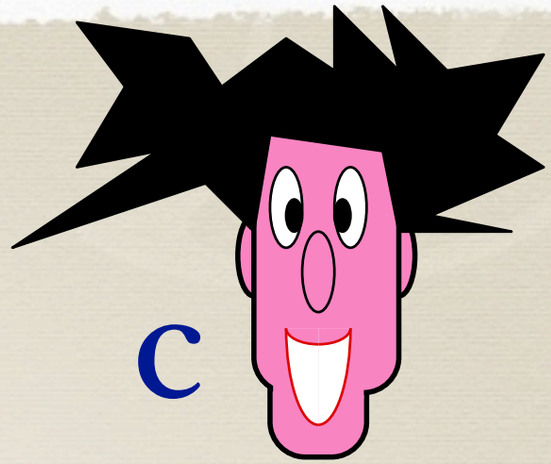
when ® is uniform.



B_O, B_I

[GM84]

[BCR86]



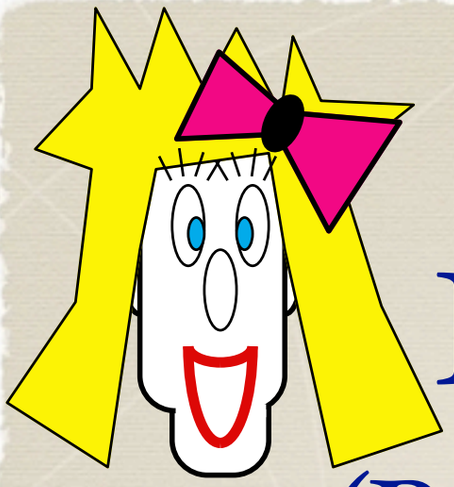
C

$enc_A(m_O)$ → U_O

$enc_A(m_I)$ → U_I

Z ← $RE_A(®, U_C)$

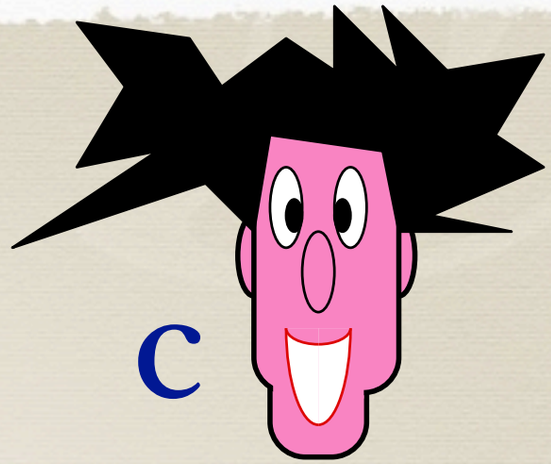
$dec_A(Z)$ → y



B_O, B_I

[GM84]

[BCR86]



C

$m_O = \pi^{-1}(B_O)$

$enc_A(m_O)$

U_O

$m_I = \pi^{-1}(B_I)$

$enc_A(m_I)$

U_I

Z

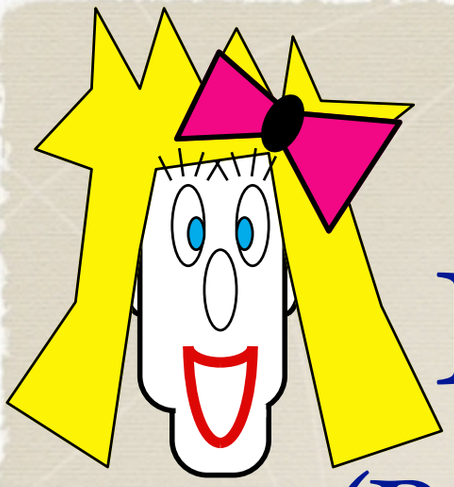
$RE_A(®, U_C)$

$dec_A(Z)$

y

$m_C = RD_A(®, y)$

$B_C = \pi(m_C)$

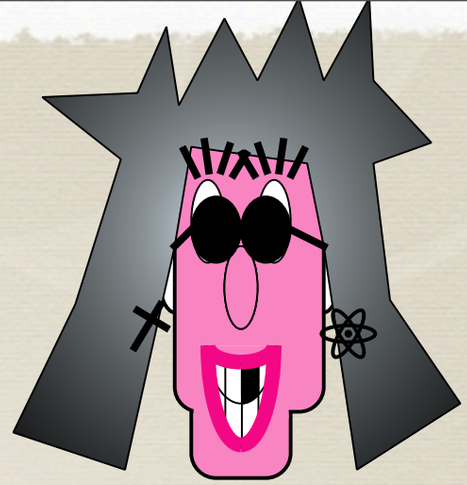


[GM84]

[BCR86]

B_O, B_I

$c = \oplus$



$m_O = \pi^{-I}(B_O)$

$enc_A(m_O)$

U_O

$m_I = \pi^{-I}(B_I)$

$enc_A(m_I)$

U_I

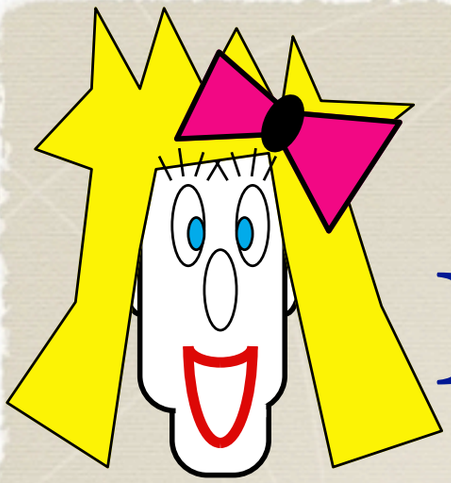
Z

$RE_A(®, U_O + U_I)$

$dec_A(Z)$

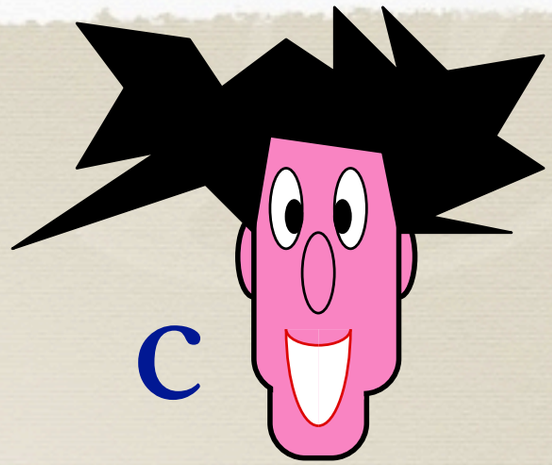
y

$B_O \oplus B_I = \pi(RD_A(®, y))$



B_O, B_I

[BCR86]



C

$enc_A(m_O)$ → U_O

$enc_A(m_I)$ → U_I

Z ← $RE_A(®, U_C)$

$dec_A(Z)$ → y

Use ZK proofs to make sure both parties follow protocol.

OT Implementations

- RSA
- Rabin/Factoring
- Goldwasser-Micali
- Paillier
- ElGammal

(3)
Post-Quantum Secure OT

Implementations

Post-Quantum OT

2008

Oblivious Transfer via McEliece's PKC and Permuted Kernels

K. Kobara[1], Kirill Morozov[1] and R. Overbeck[2]

[1] RCIS, AIST
{k-kobara,kirill.morozov}@aist.go.jp
[2] TU-Darmstadt,
Department of Computer Science,
Cryptography and Computer Algebra Group.
overbeck@cdc.informatik.tu-darmstadt.de

Oblivious Transfer Based on the McEliece Assumptions

Rafael Dowsley[1], Jeroen van de Graaf[2], Jörn Müller-Quade[3], and Anderson C.A. Nascimento[1]

2008

A Framework for Efficient and Composable Oblivious Transfer

Chris Peikert*
SRI International

Vinod Vaikuntanathan
MIT†

Brent Waters‡
SRI International

August 25, 2008

Abstract

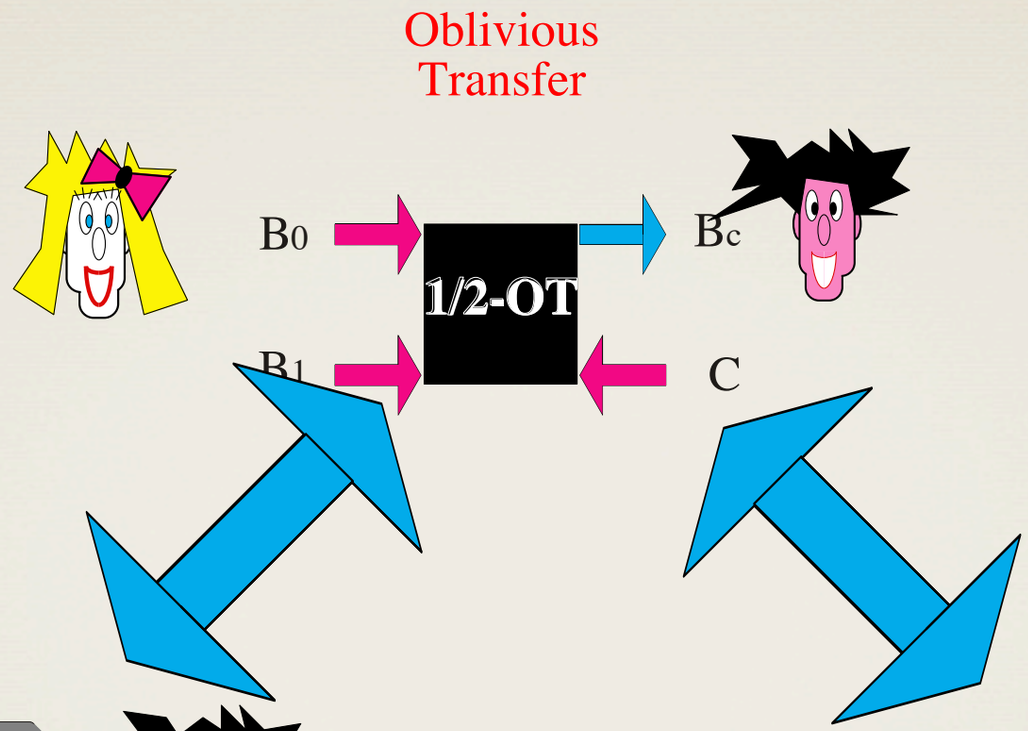
We propose a simple and general framework for constructing oblivious transfer (OT) protocols that are *efficient*, *universally composable*, and *generally realizable* under any one of a variety of standard number-theoretic assumptions, including the decisional Diffie-Hellman assumption, the quadratic residuosity and decisional composite residuosity assumptions, and *worst-case* lattice assumptions.

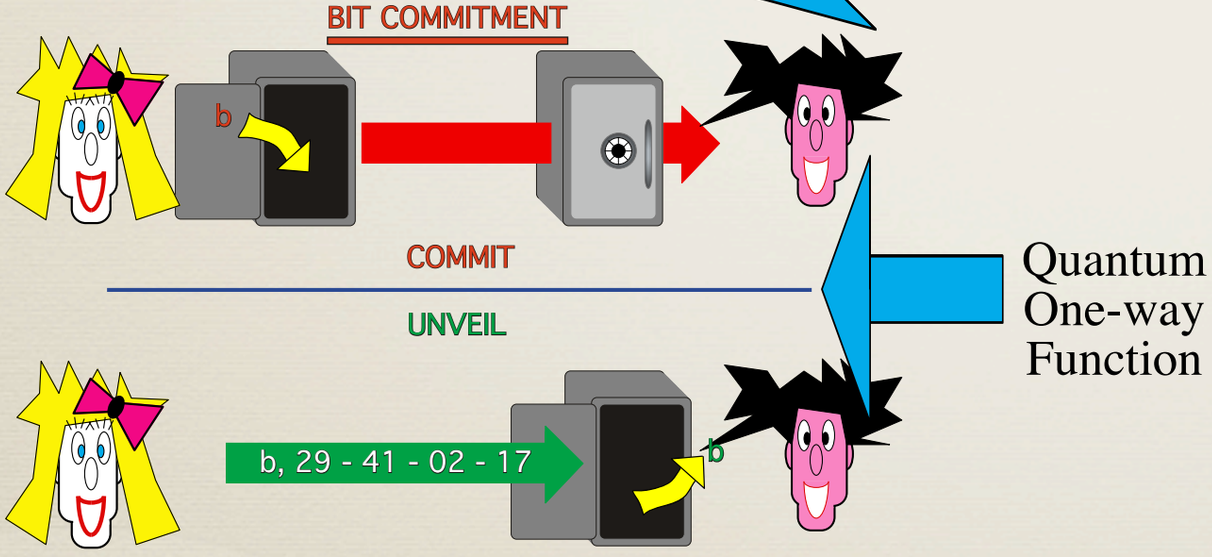
Our OT protocols are round-optimal (one message each way), quite efficient in computation and communication, and can use a single common string for an unbounded number of executions between the same sender and receiver. Furthermore, the protocols can provide *statistical* security to either the sender or the receiver, simply by changing the distribution of the common string. For certain instantiations of the protocol, even a *uniformly random* common string suffices.

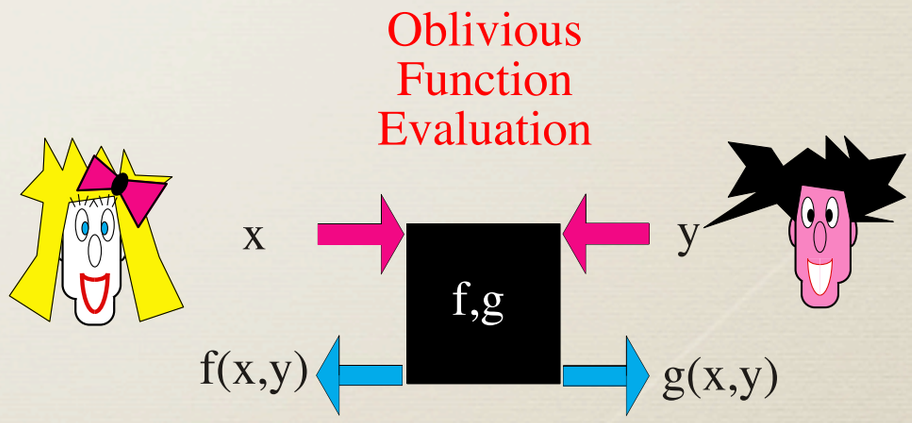
Our key technical contribution is a simple abstraction that we call a *dual-mode* cryptosystem. We implement dual-mode cryptosystems by taking a unified view of several cryptosystems that have what we call "messy" public keys, whose defining property is that a ciphertext encrypted under such a key carries *no information* (statistically) about the encrypted message.

As a contribution of independent interest, we also provide a multi-bit *amortized* version of Regev's lattice-based cryptosystem (STOC 2005) whose time and space complexity are improved by a linear factor in the security parameter n. The resulting amortized encryption and decryption times are only $\tilde{O}(n)$ bit operations per message bit, and the ciphertext expansion can be made as small as a constant; the public key size and underlying lattice assumption remain essentially the same.

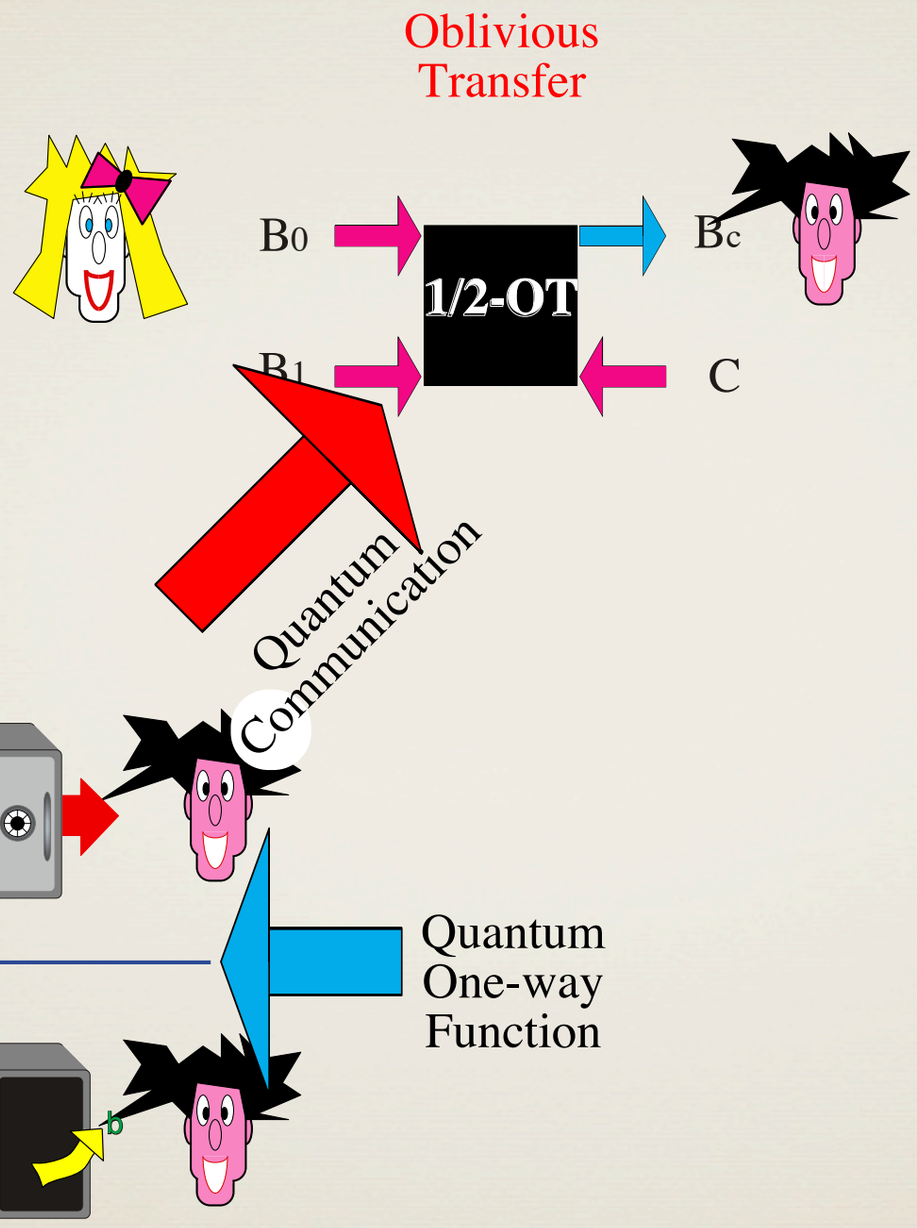
Quantumly


Oblivious
Transfer
B0
B1
1/2-OT
Bc
C

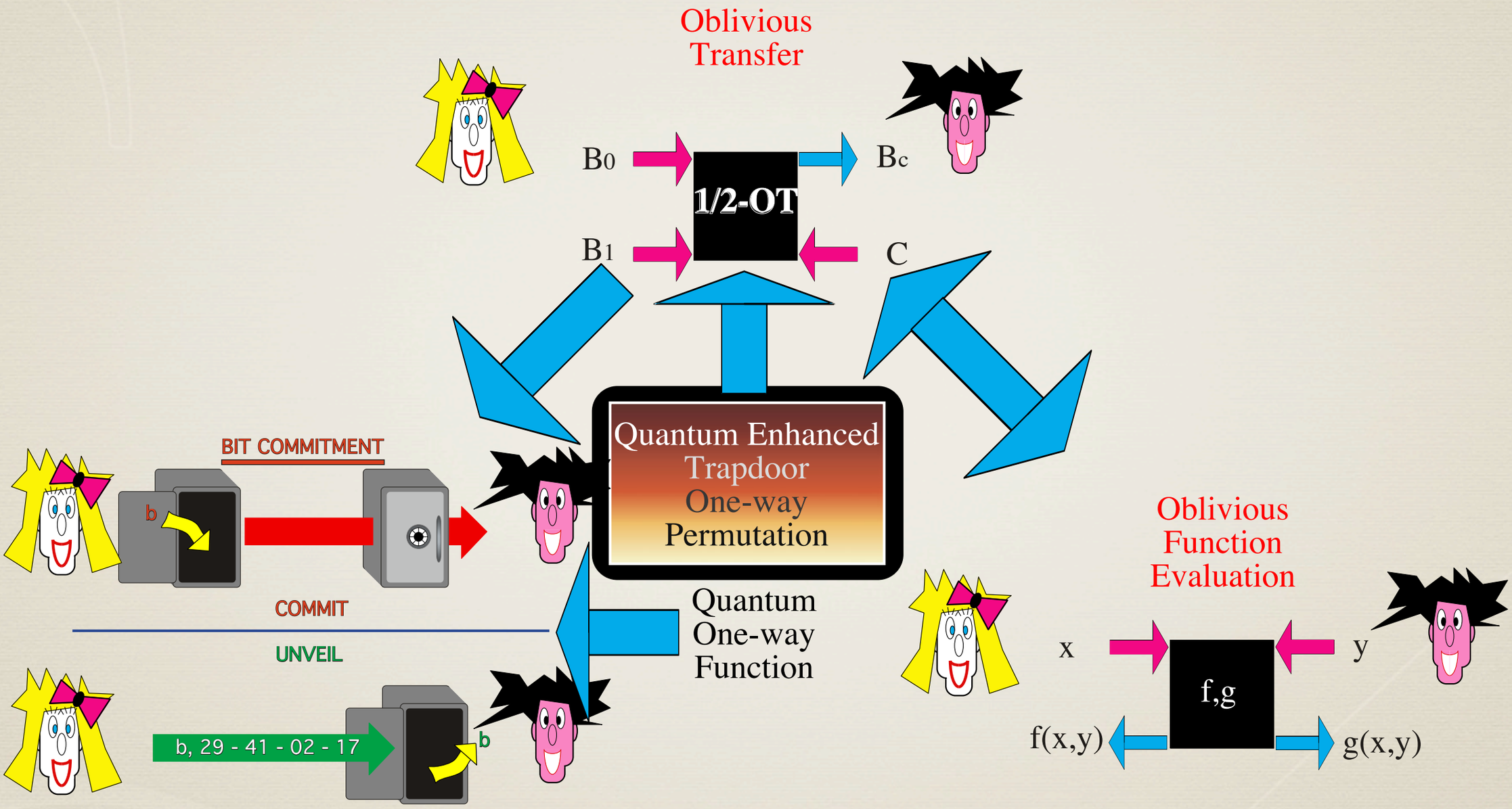

BIT COMMITMENT
b
COMMIT
UNVEIL
b, 29 - 41 - 02 - 17
b
Quantum
One-way
Function


Oblivious
Function
Evaluation
x
y
f,g
f(x,y)
g(x,y)

Quantumly


Oblivious
Transfer
B0
B1
1/2-OT
Bc
C
Quantum
Communication
BIT COMMITMENT
b
COMMIT
UNVEIL
b, 29 - 41 - 02 - 17
b
Quantum
One-way
Function

Quantumly Secure Classically Implemented



Quantum ETOWP

* Factoring

* RSA

* Elliptic Curves

* ElGamal

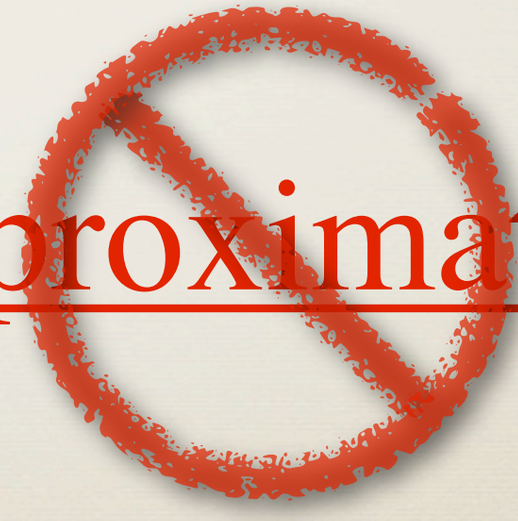
Quantum *ETOWP*

* McEliece

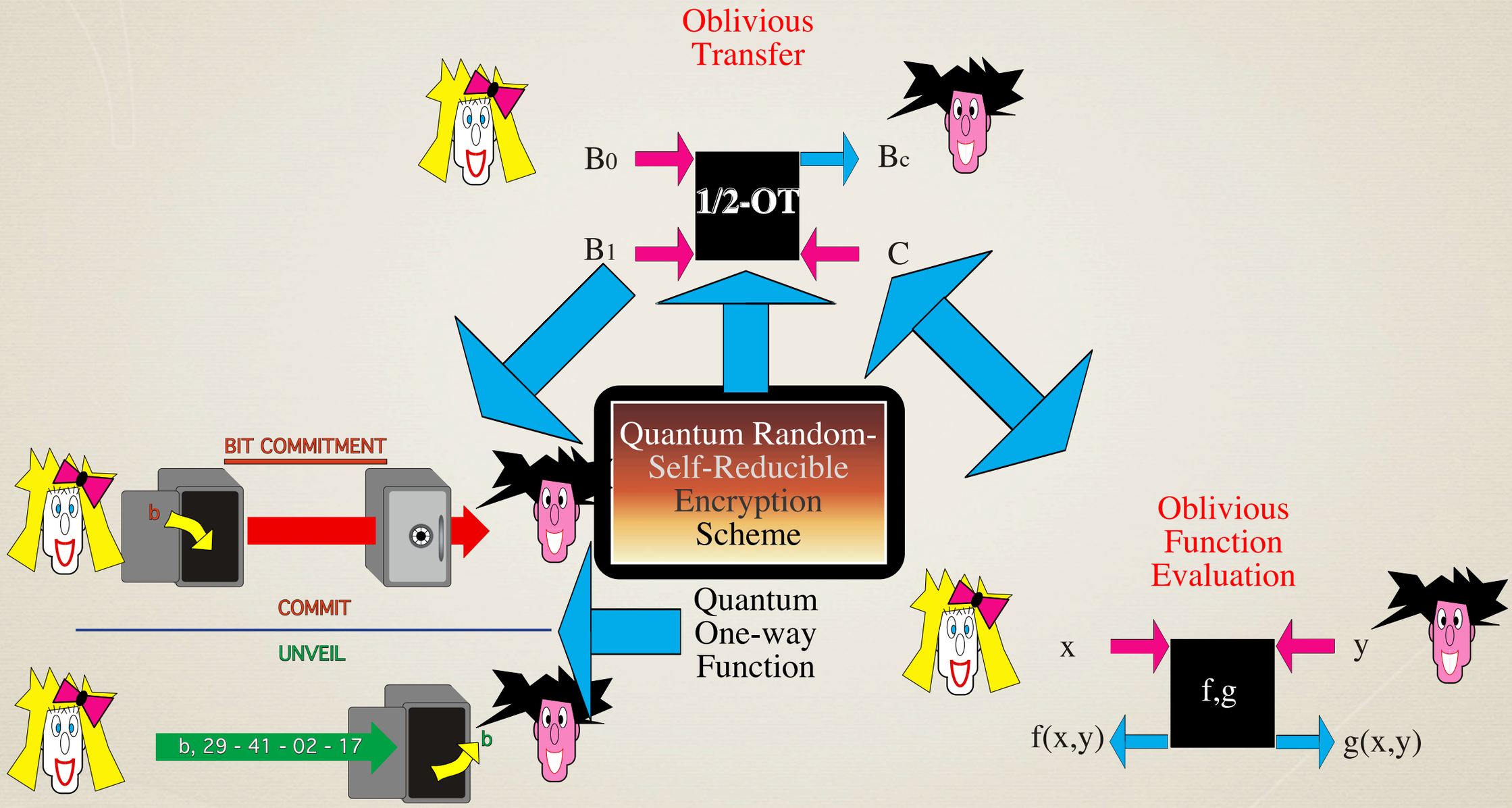
* Lattices

* LWE

* Approximate GCD



Quantumly Secure Classically Implemented



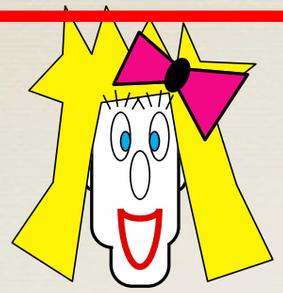
Quantum RSR Encryption

* RSA
* Rabin/Factoring
* Goldwasser-Micali
* Paillier
* ElGammal

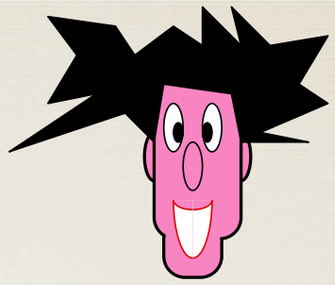
Quantum *RSR* Encryption

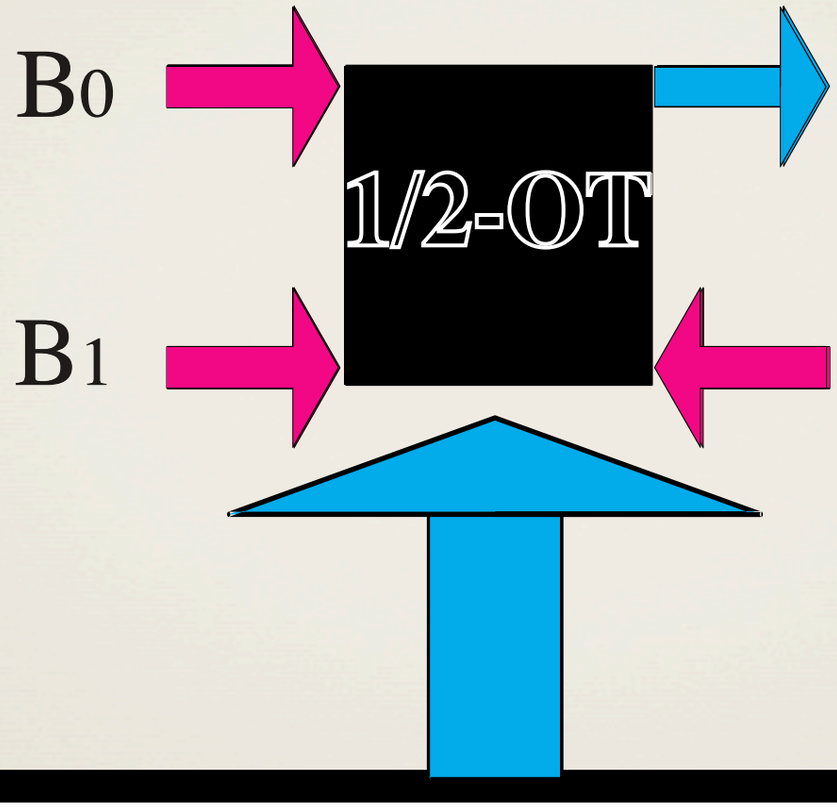
- Lattices
- McEliece
- Approximate GCD
- LWE

Quantumly Secure Classically Implemented



Oblivious Transfer





Quantum Weakly Random-Self-Reducible Encryption Scheme

This work : [CK13]

This work : [CK13]

* INGREDIENTS:
  Public-Key Cryptosystem (enc, dec)
  and hard-core predicate π

This work : [CK13]

* INGREDIENTS:
  Public-Key Cryptosystem (enc, dec)
  and hard-core predicate π

* Define $enc' = enc(\pi^{-1}(\bullet))$, $dec' = \pi(dec(\bullet))$
  we need (enc', dec') be semantically sec.

This work : [CK13]

- INGREDIENTS:
  Public-Key Cryptosystem (enc, dec)
  and hard-core predicate π
- Define $enc' = enc(\pi^{-1}(\bullet))$, $dec' = \pi(dec(\bullet))$
  we need (enc′,dec′) be semantically sec.
- We also need (enc,dec) to be wRSR.

This work : [CK13]

This work : [CK13]

- **Definition (wRSR cryptosystem)**
  A cryptosystem (enc,dec) is *weakly-Rand-Self-Reducible* if there exists a pair of PPT algorithms (RE,RD) such that for all ®,m (re-encrypt,re-decrypt)

$$RD(®,dec(RE(®,enc(m))))=m$$

and for all m,m′

$$RE(®,enc(m)) \sim RE(®,enc(m'))$$

for ® according to some PPT samplable distribution.

This work : [CK13]

This work : [CK13]

* **Definition (wRSR cryptosystem)**
  A cryptosystem (enc,dec) is *weakly-Rand-Self-Reducible* if there exists a pair of PPT algorithms (RE,RD) such that for all ®,m (re-encrypt,re-decrypt)

$$RD(®,dec(RE(®,enc(m))))=m$$

and for all m,m′

$$RE(®,enc(m)) \sim RE(®,enc(m'))$$

for ® according to some PPT samplable distribution.

This work : [CK13]

- **Definition (wRSR cryptosystem)**
  A cryptosystem (enc,dec) is *weakly-Rand-Self-Reducible* if there exists a pair of PPT algorithms (RE,RD) such that for all ®,m (re-encrypt,re-decrypt)

$$RD(®,dec(RE(®,enc(m))))=m$$

and for all m,m′

$$RE(®,enc(m)) \sim RE(®,enc(m'))$$

for ® according to some PPT samplable distribution.

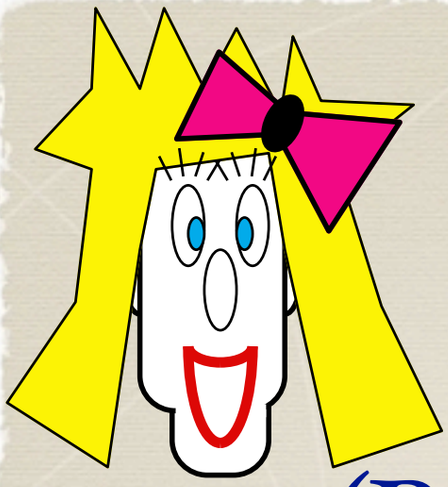
- Notice however that RE(®,enc(m) might not be a valid ciphertext, but dec(RE(®,enc(m)) still make sense.

[BCR86]

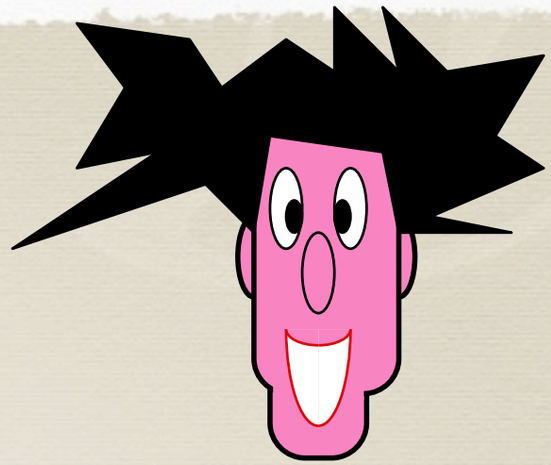
$$Z_O = Z = Z_I$$

[CK13]

$$Z_O \sim Z_I$$

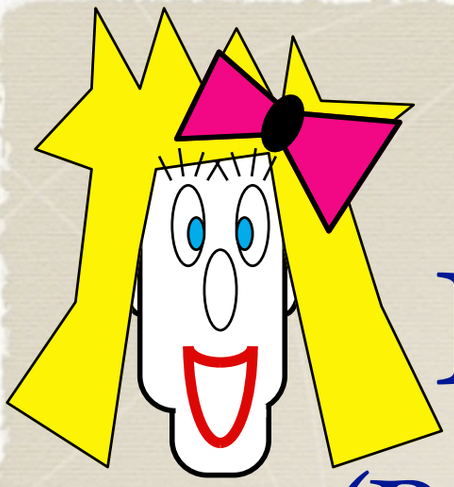


[BCR86]
[CK13]



$m_O = \pi^{-I}(B_O)$

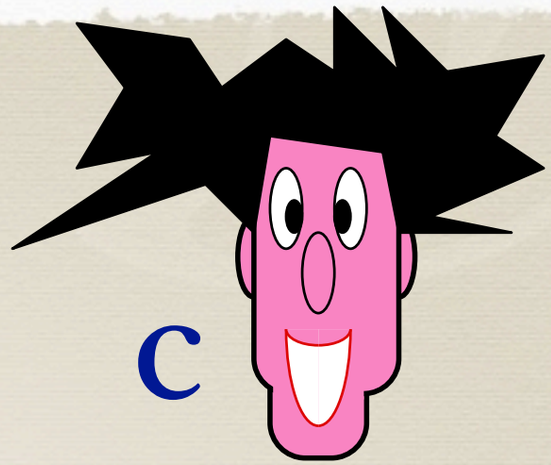
$m_I = \pi^{-I}(B_I)$



B_O, B_I

[BCR86]

[CK13]



c

$m_O = \pi^{-1}(B_O)$

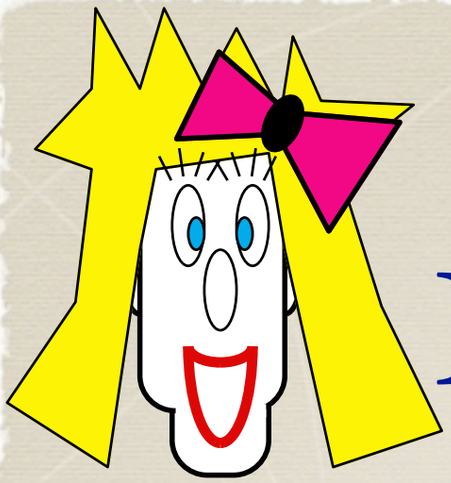
$enc_A(m_O)$

U_O

$m_I = \pi^{-1}(B_I)$

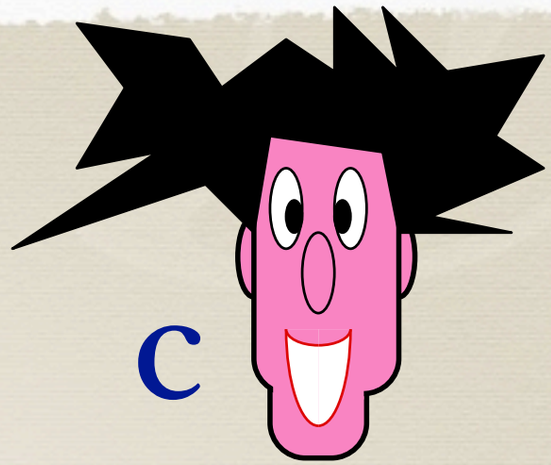
$enc_A(m_I)$

U_I



B_O, B_I

[CK13]

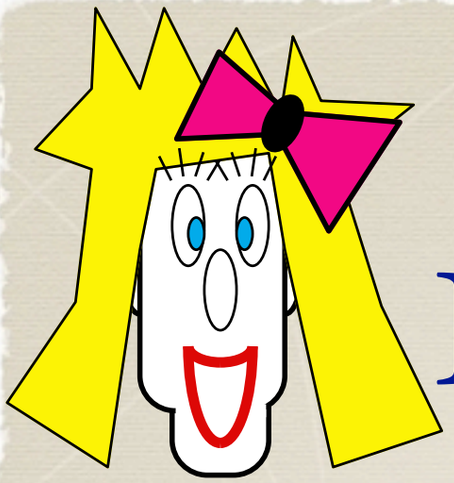


C

$enc_A(m_O)$ → U_O

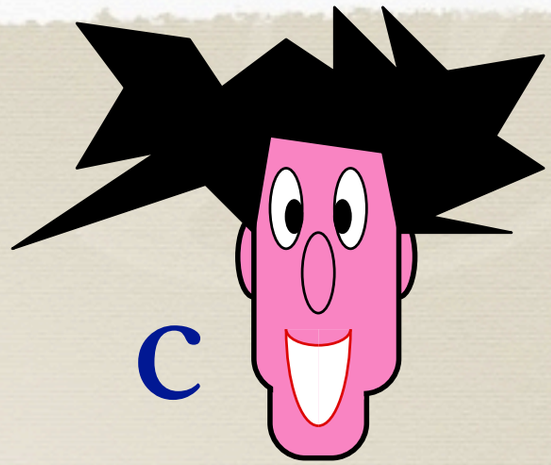
$enc_A(m_I)$ → U_I

Z_c ← $RE_A(®, U_C)$



B_O, B_I

[CK13]



c

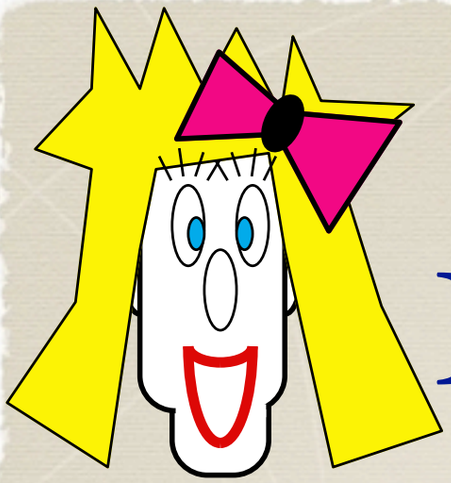
$enc_A(m_O)$

U_O

U_I

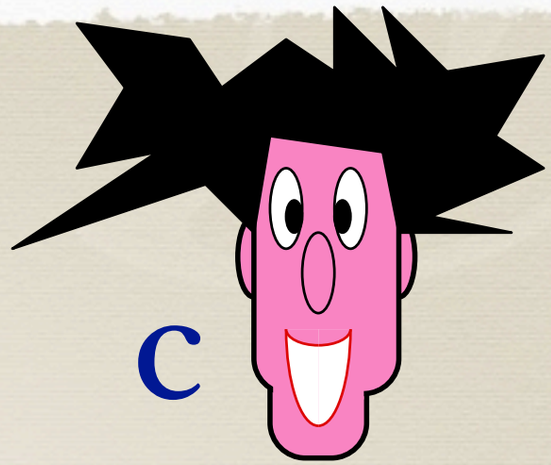
$RE_A(®, U_C)$

Z_C



B_O, B_I

[CK13]



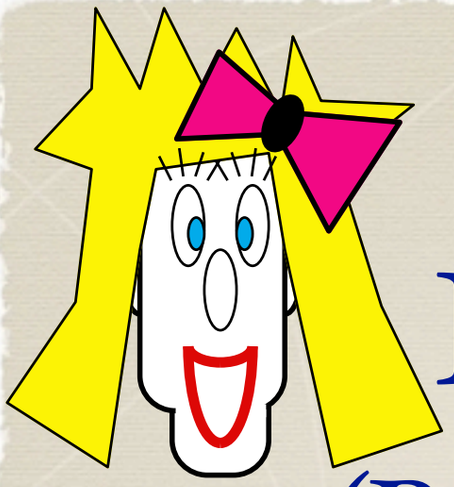
C

$enc_A(m_O)$ → U_O

$enc_A(m_I)$ → U_I

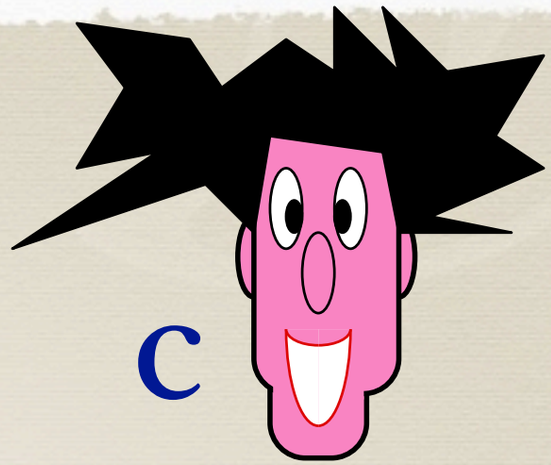
Z_c ← $RE_A(®, U_C)$

$dec_A(Z_c)$ → y_c



B_O, B_I

[CK13]



c

$m_O = \pi^{-1}(B_O)$

$enc_A(m_O)$

U_O

$m_I = \pi^{-1}(B_I)$

$enc_A(m_I)$

U_I

Z_c

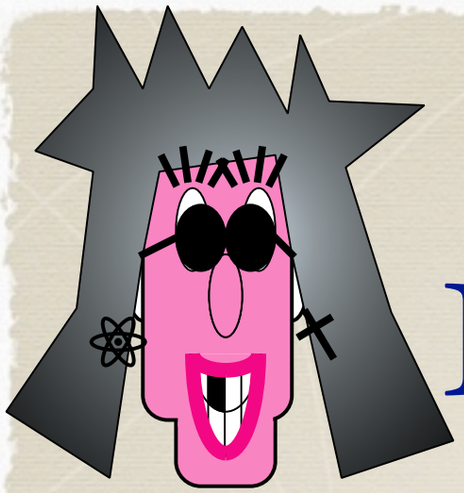
$RE_A(®, U_c)$

$dec_A(Z_c)$

y_c

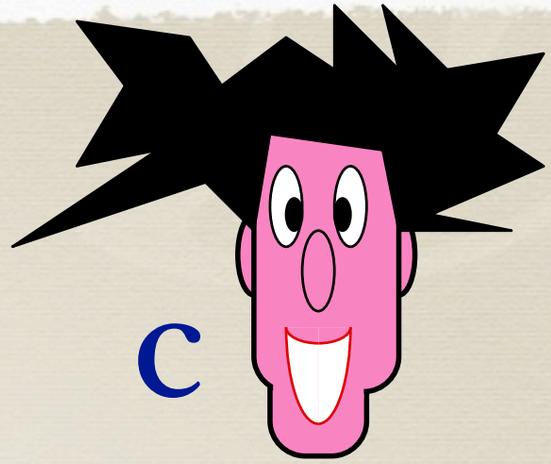
$m_c = RD_A(®, y_c)$

$B_c = \pi(m_c)$



B_O, B_I

[CK13]



C

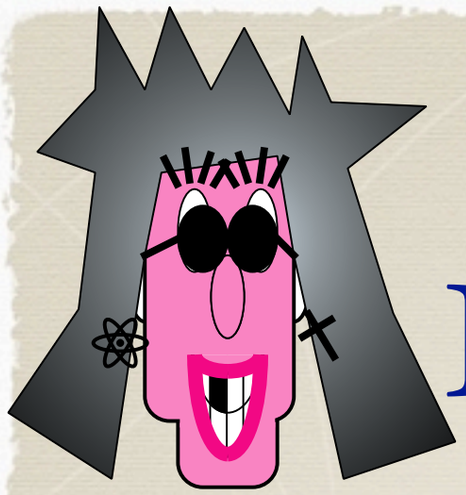
$che_A(m_O)$ → U_O

$che_A(m_I)$ → U_I

Z_C ← $RE_A(®, U_C)$

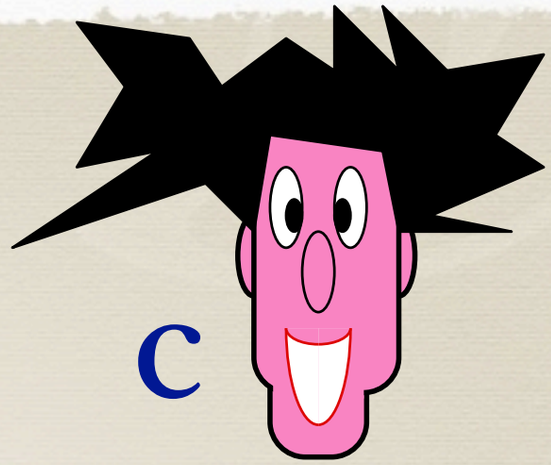
$dec_A(Z_C)$ → y_c

$B_c = \pi(RD_A(®, y_c))$



B_o, B_I

[CK13]



c

$che_A(m_o)$

cl

Z_C

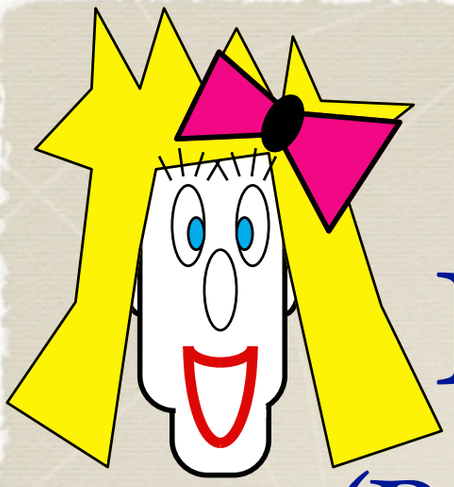
U_o

U_I

$RE_A(®, U_C)$

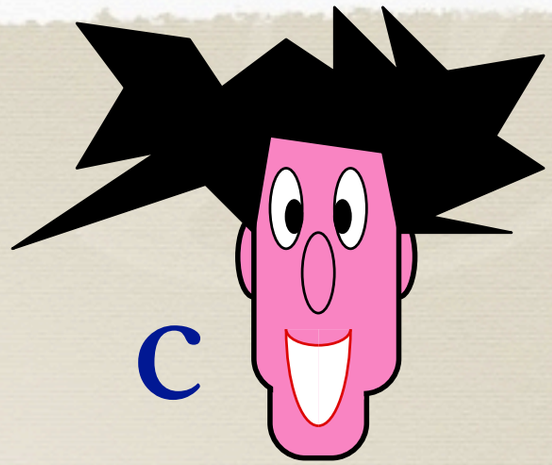
y_c

$B_c = \pi(RD_A(®, y_c))$



B_O, B_I

[CK13]



C

$m_O = \pi^{-1}(B_O)$

$enc_A(m_O)$

U_O

$m_I = \pi^{-1}(B_I)$

$enc_A(m_I)$

U_I

Z_c

$RE_A(®, U_C)$

$dec_A(Z_c)$

y_c

Use ZK proofs to make sure both parties follow protocol.

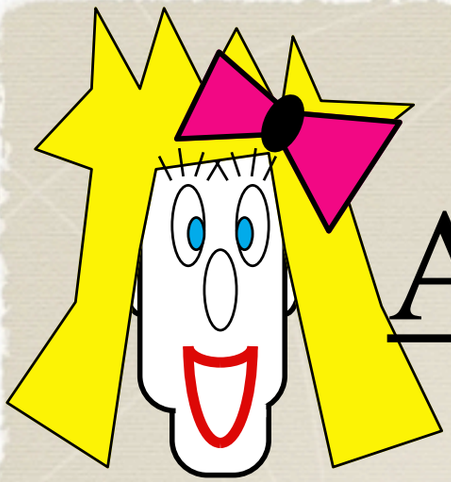
Quantum wRSR Encryption

Lattices

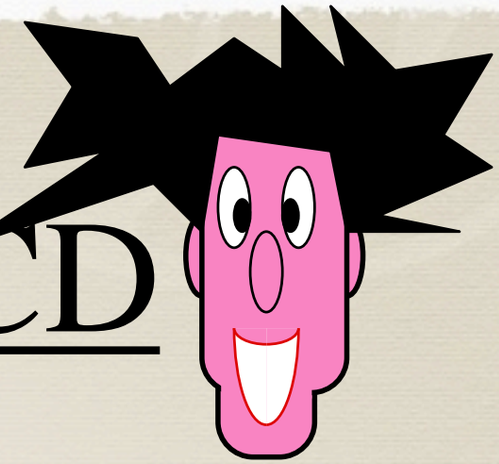
Approximate GCD

LWE

(4)
Post-Quantum Secure OT
Example :
Approximate Integer GCD

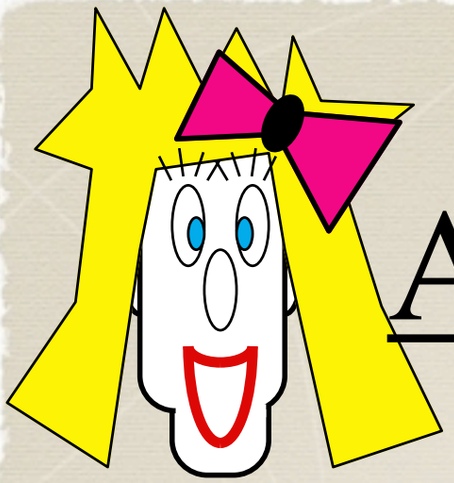


Approximate Integer GCD

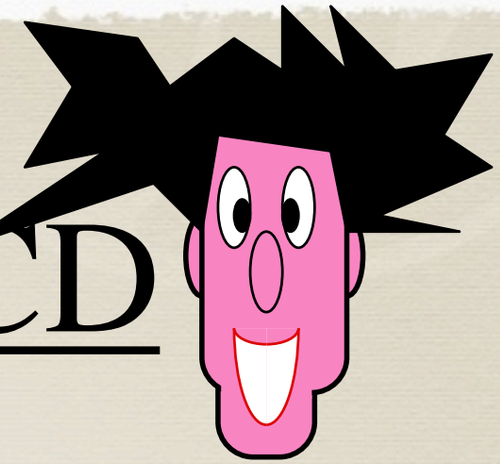


$\mathrm{enc}_A(s_o, e_o, b_o)$ ⇒ U_o

$\mathrm{enc}_A(s_I, e_I, b_I)$ ⇒ U_I



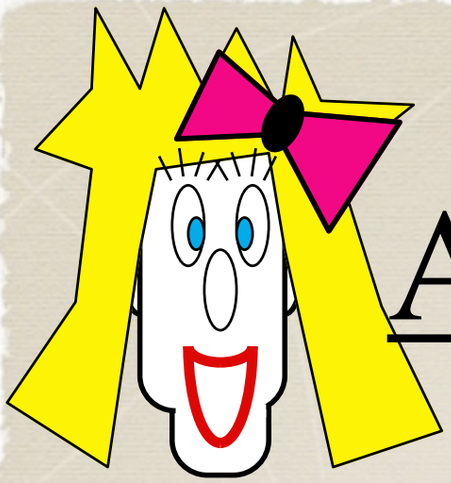
Approximate Integer GCD



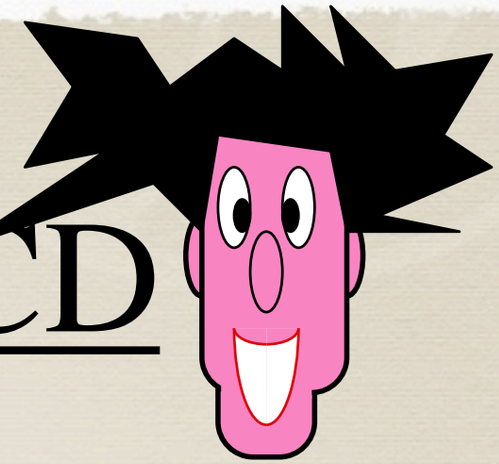
$enc_A(s_0,e_0,b_0)$ → U_0

$enc_A(s_1,e_1,b_1)$ → U_1

$U_c+enc_A(s,E,b) \bmod x_0$ → Z_c



Approximate Integer GCD

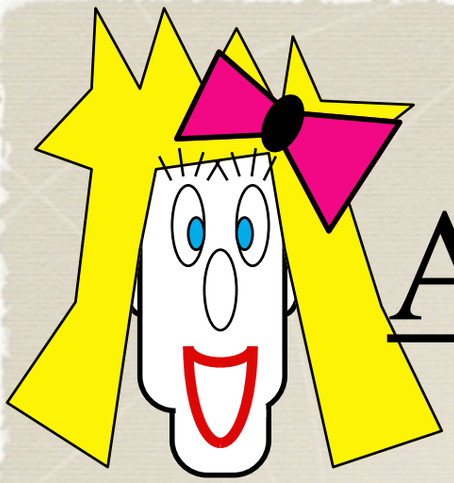


$\mathrm{enc}_A(s_0,e_0,b_0)$ ⇒ U_0

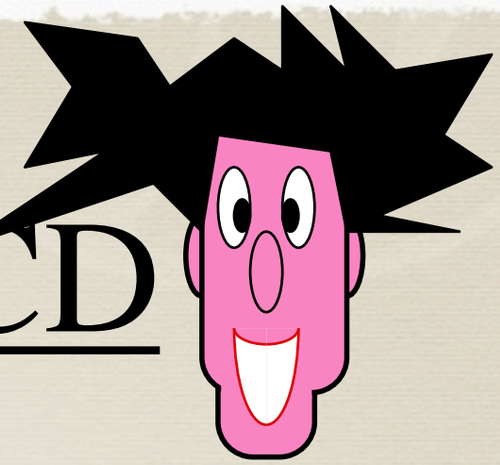
$\mathrm{enc}_A(s_1,e_1,b_1)$ ⇒ U_1

Z_c ⇐ $U_c+\mathrm{enc}_A(s,E,b) \bmod x_0$

$= \mathrm{enc}_A(s',e_c+E,b_c \oplus b)$



Approximate Integer GCD



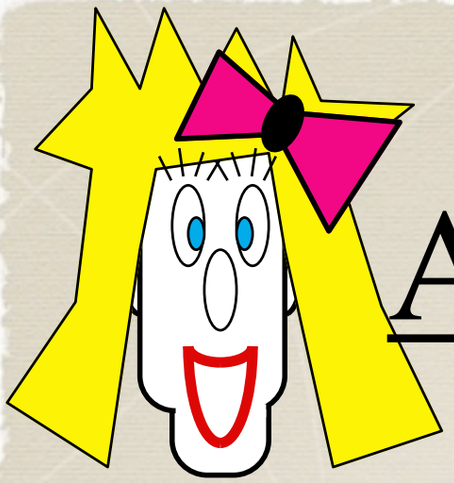
$enc_A(s_O,e_O,b_O)$ ⇒ U_O

$enc_A(s_I,e_I,b_I)$ ⇒ U_I

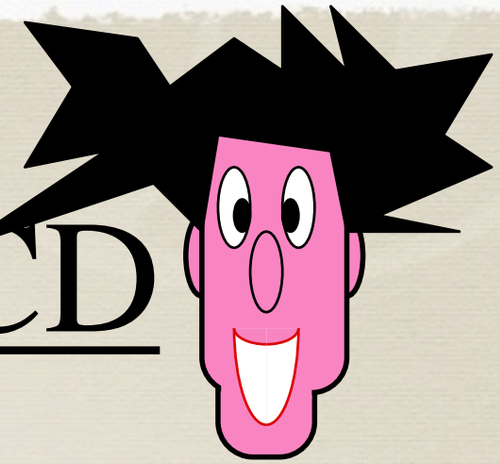
Z_c ⇐ $U_c+enc_A(s,E,b) \bmod x_O$

secure if $|e_O| \ll |E|$, $|e_I| \ll |E|$

$U_O+enc_A(s,E,b) \bmod x_O \sim U_I+enc_A(s,E,b) \bmod x_O$



Approximate Integer GCD



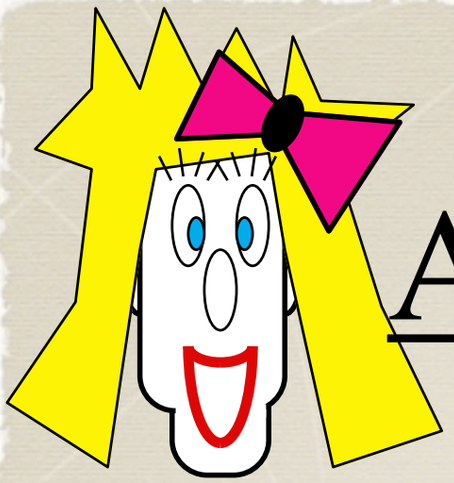
$enc_A(s_O,e_O,b_O)$ → U_O

$enc_A(s_I,e_I,b_I)$ → U_I

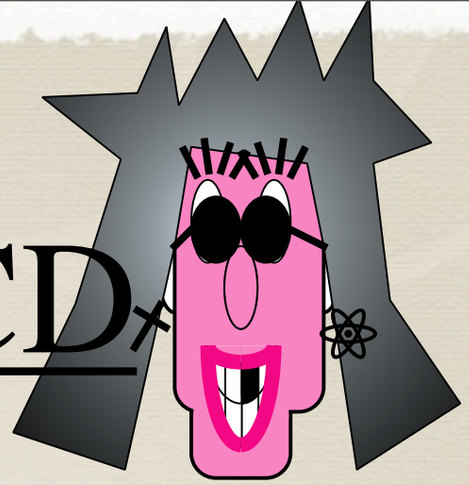
Z_c ← $U_c+enc_A(s,E,b) \bmod x_O$

$dec_A(Z_c)$ → y_c

$b_c=y_c \oplus b$



Approximate Integer GCD



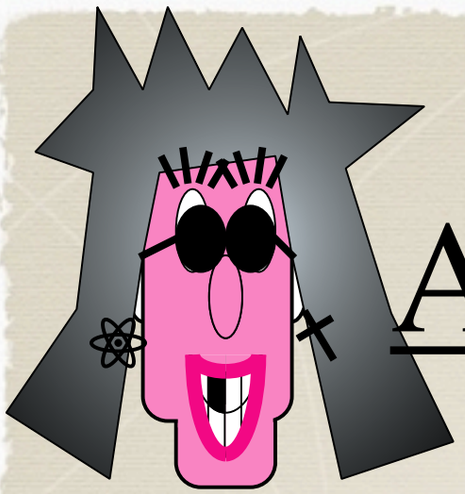
$enc_A(s_0,e_0,b_0)$ → U_0

$enc_A(s_1,e_1,b_1)$ → U_1

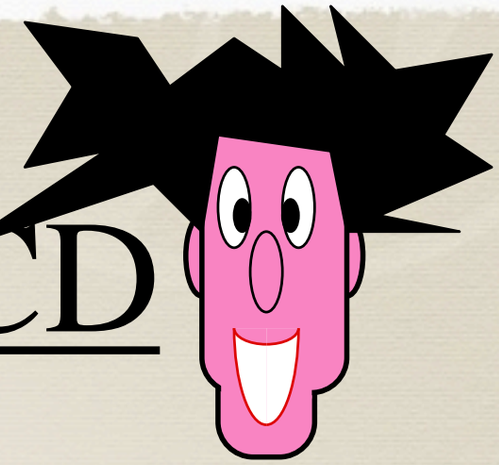
Z ← $U_0+U_1+enc_A(s,E,b) \bmod x_0$

$dec_A(Z)$ → y

$b_0 \oplus b_1 = y \oplus b$



Approximate Integer GCD



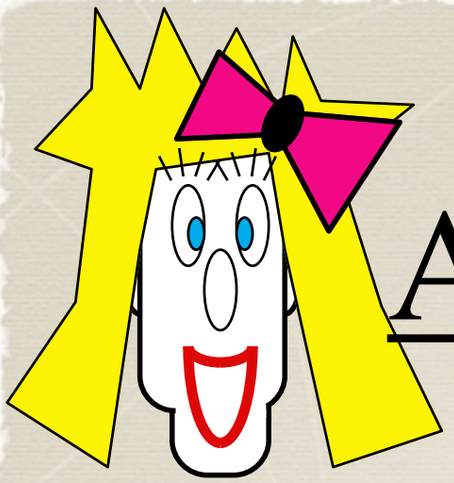
$enc_A(s_O,E_O,b_O)$ ⇒ U_O

$enc_A(s_I,E_I,b_I)$ ⇒ U_I

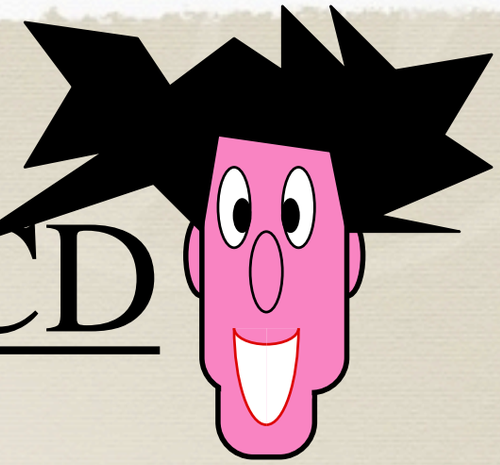
Z_C ⇐ $U_C+enc_A(s,E,b) \bmod x_O$

$dec_A(Z_C)$ ⇒ y_C

$b_C=y_C \oplus b$



Approximate Integer GCD



$enc_A(s_O,e_O,b_O)$ → U_O

$enc_A(s_I,e_I,b_I)$ → U_I

Z_c ← $U_c+enc_A(s,E,b) \bmod x_O$

$dec_A(Z_c)$ → y_c

Use ZK proofs to make sure both parties follow protocol.

[CK13]

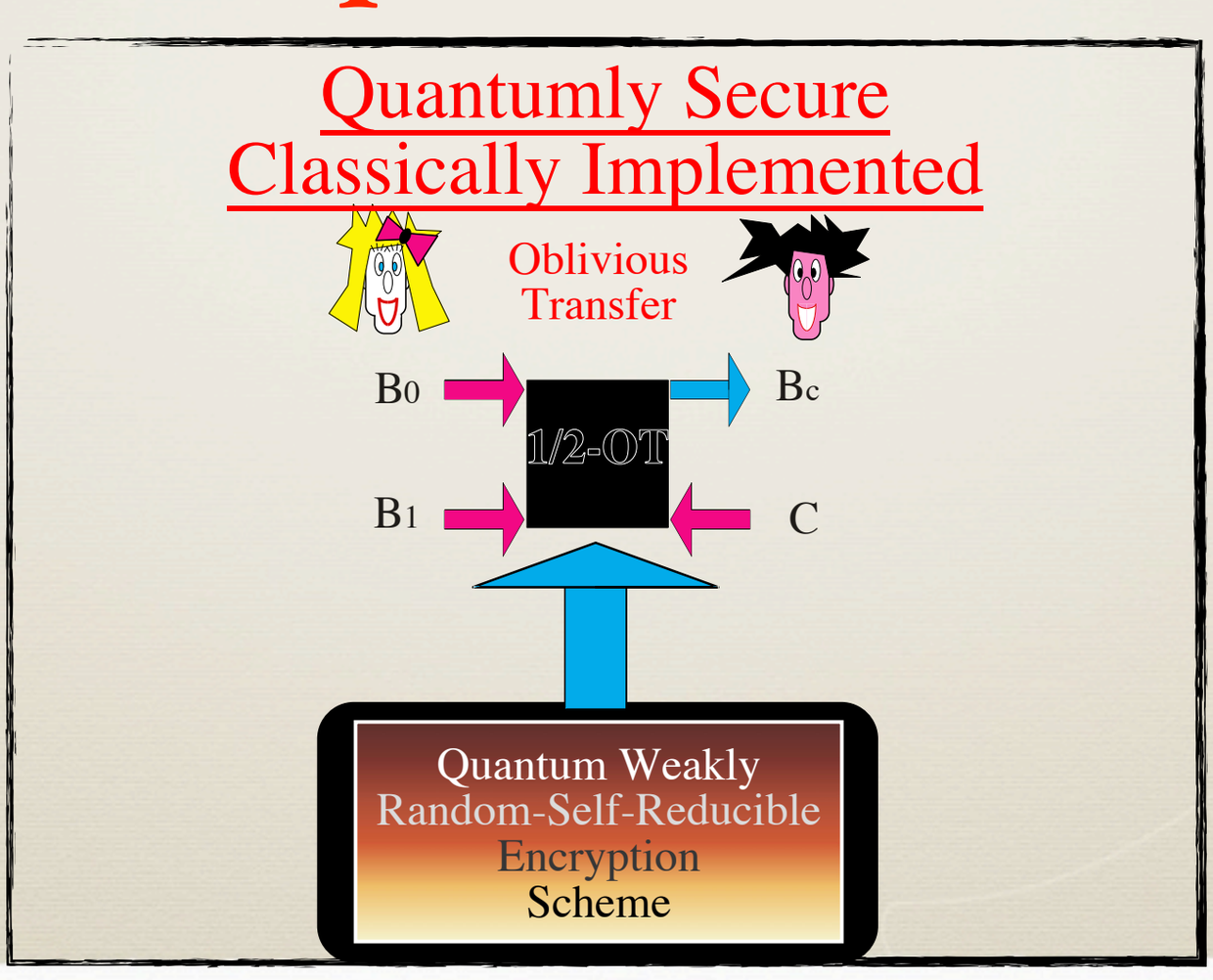
- Use ZK proofs to make sure both parties follow protocol.
- Use $\mathrm{enc}_A(s,e,b)$ to implement both Bit Commitments : protocol is not Quantum Secure.

(5)

Conslusions & Open Problems

Quantum wRSR Encryption

- A new general methodology
- Several Implementations



Quantum wRSR Encryption

???

- McEliece
- Lattices
- Approximate GCD
- LWE

Quantum wRSR Encryption

- Prove Quantum Security when using $enc_A(s,E,b)$ to implement both Bit Commitments.

Oblivious Transfer from Weakly Random-Self-Reducible Encryption

by Claude Crépeau
School of Computer Science
McGill University
joint work with Raza Ali Kazmi

Approximate Integer GCD based crypto

§