Oblivious Transfer from Weakly Random-Self-Reducible Encryption

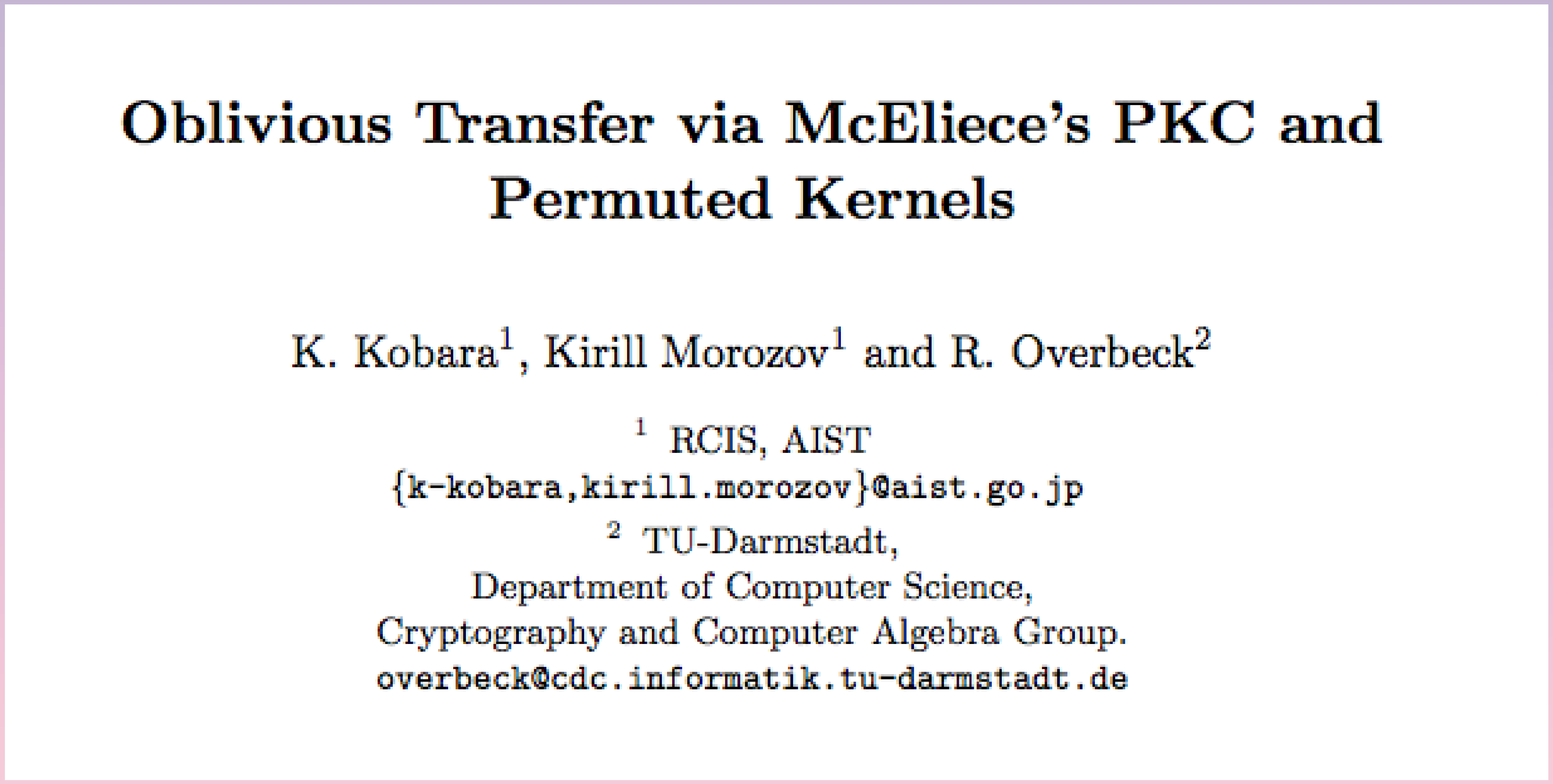
Claude Crépeau

School of Computer Science
McGill University

joint work with Raza Ali Kazmi

(0) Foreword

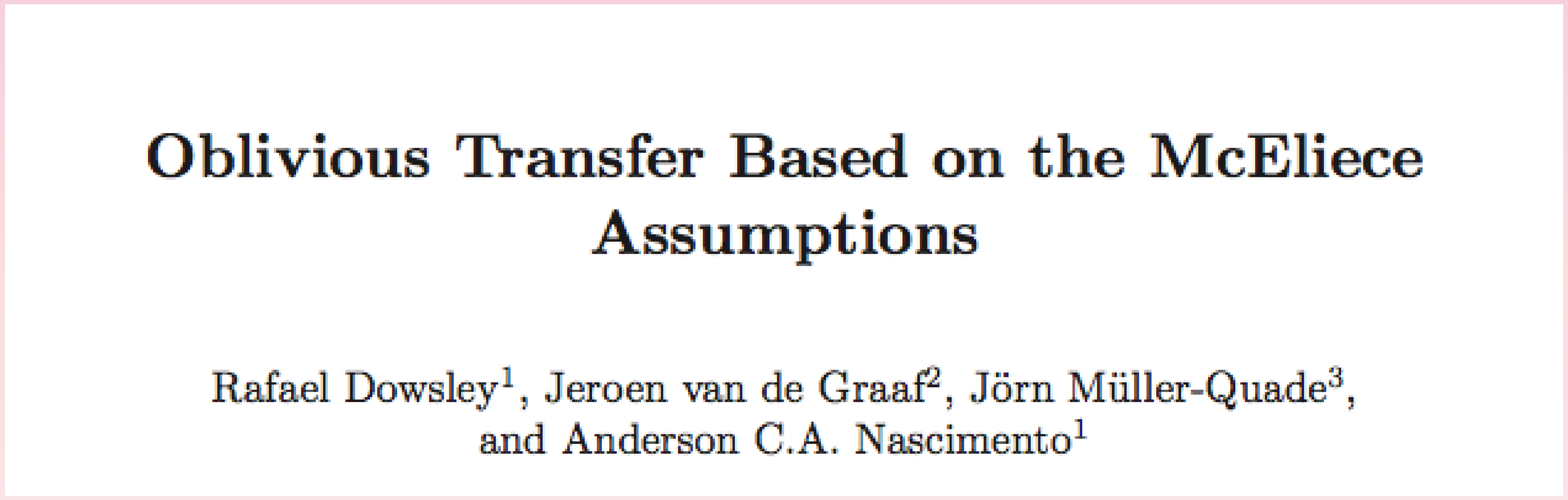
2008



Oblivious Transfer via McEliece's PKC and Permuted Kernels

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

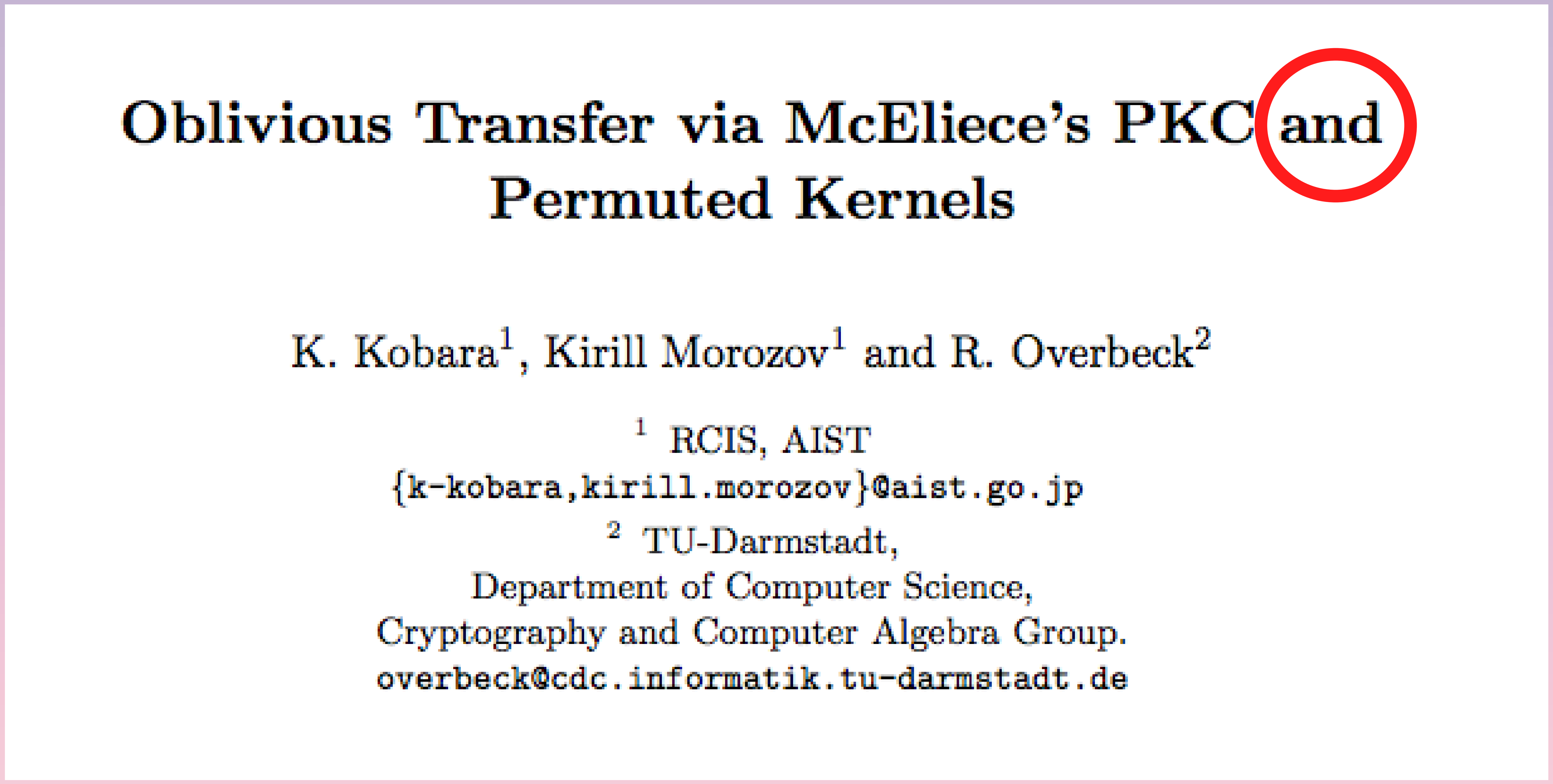
[1] RCIS, AIST
{k-kobara,kirill.morozov}@aist.go.jp
[2] TU-Darmstadt,
Department of Computer Science,
Cryptography and Computer Algebra Group.
overbeck@cdc.informatik.tu-darmstadt.de



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]

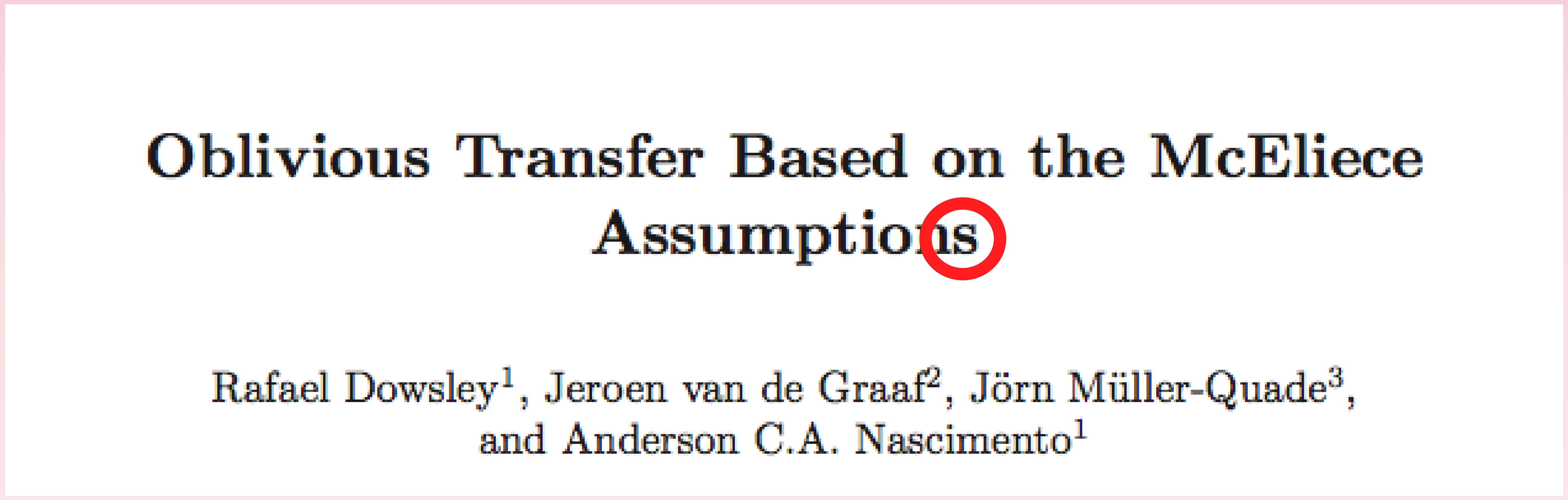
2008



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

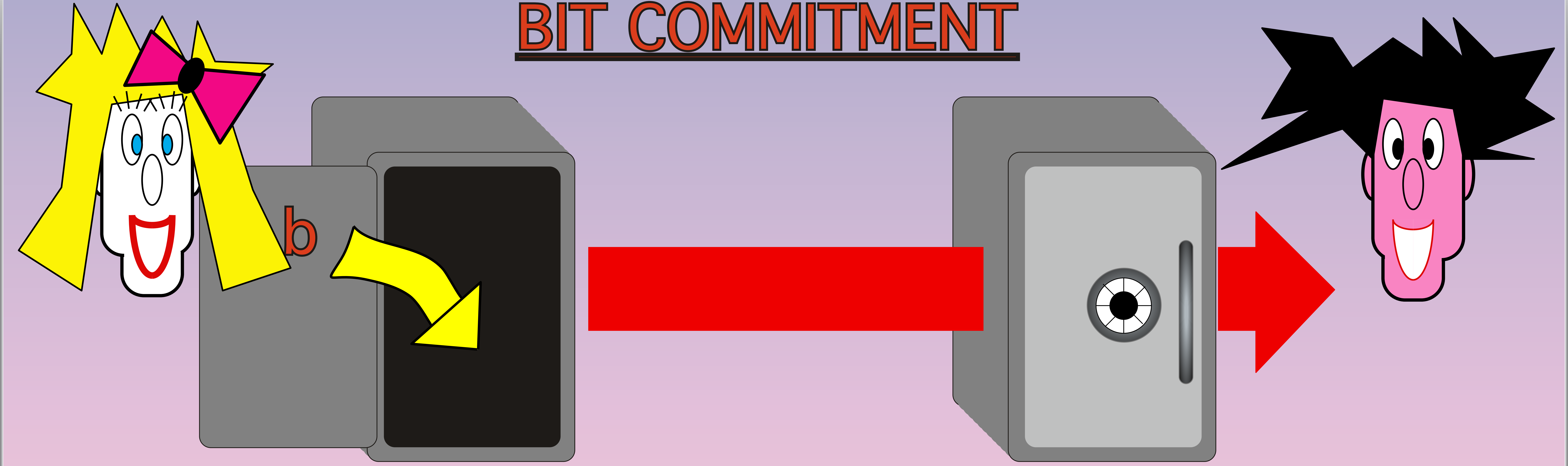


Oblivious Transfer Based on the McEliece Assumptions

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

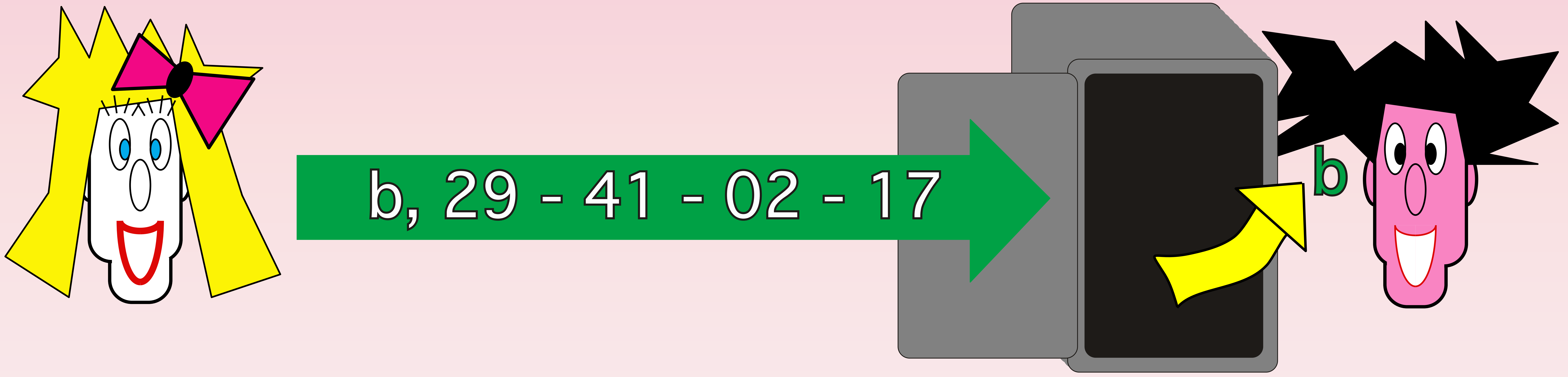
(1) two-party Cryptographic Protocols

BIT COMMITMENT



COMMIT

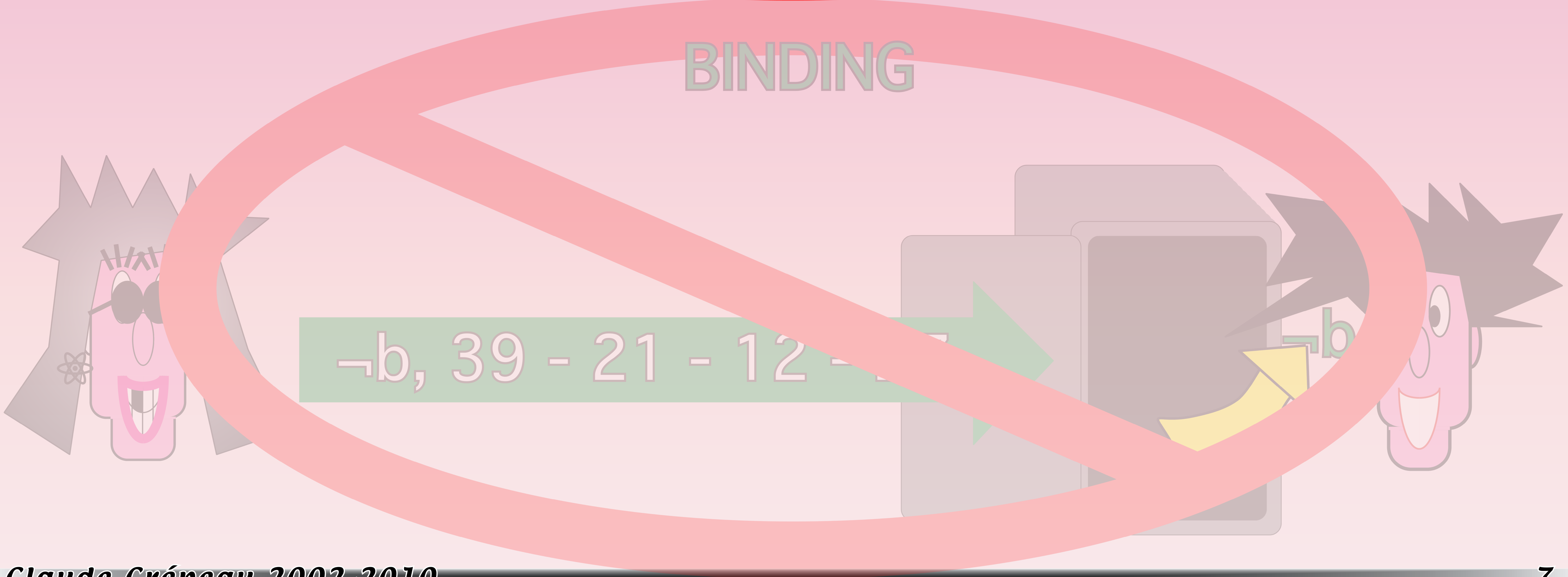
UNVEIL



BIT COMMITMENT

CONCEALING

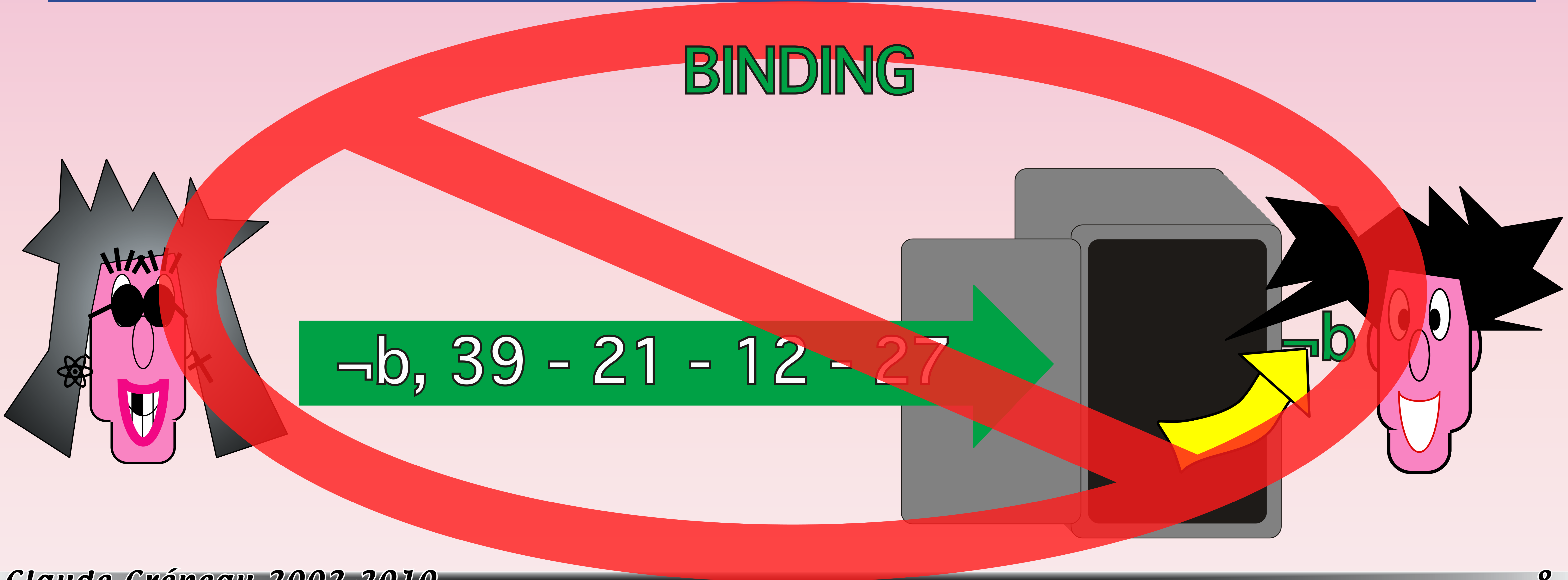
BINDING



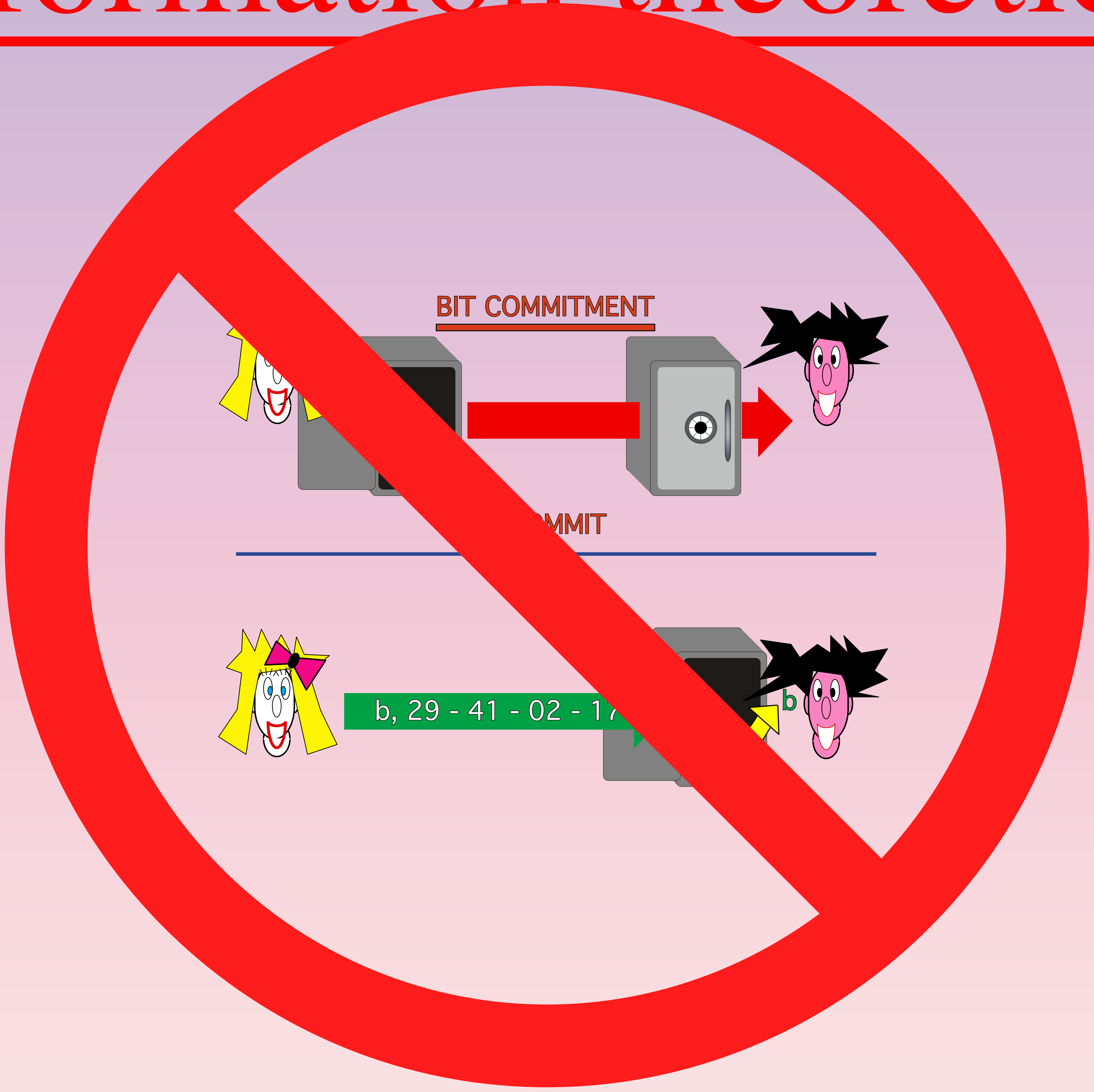
BIT COMMITMENT

b
b
CONCEALING

BINDING

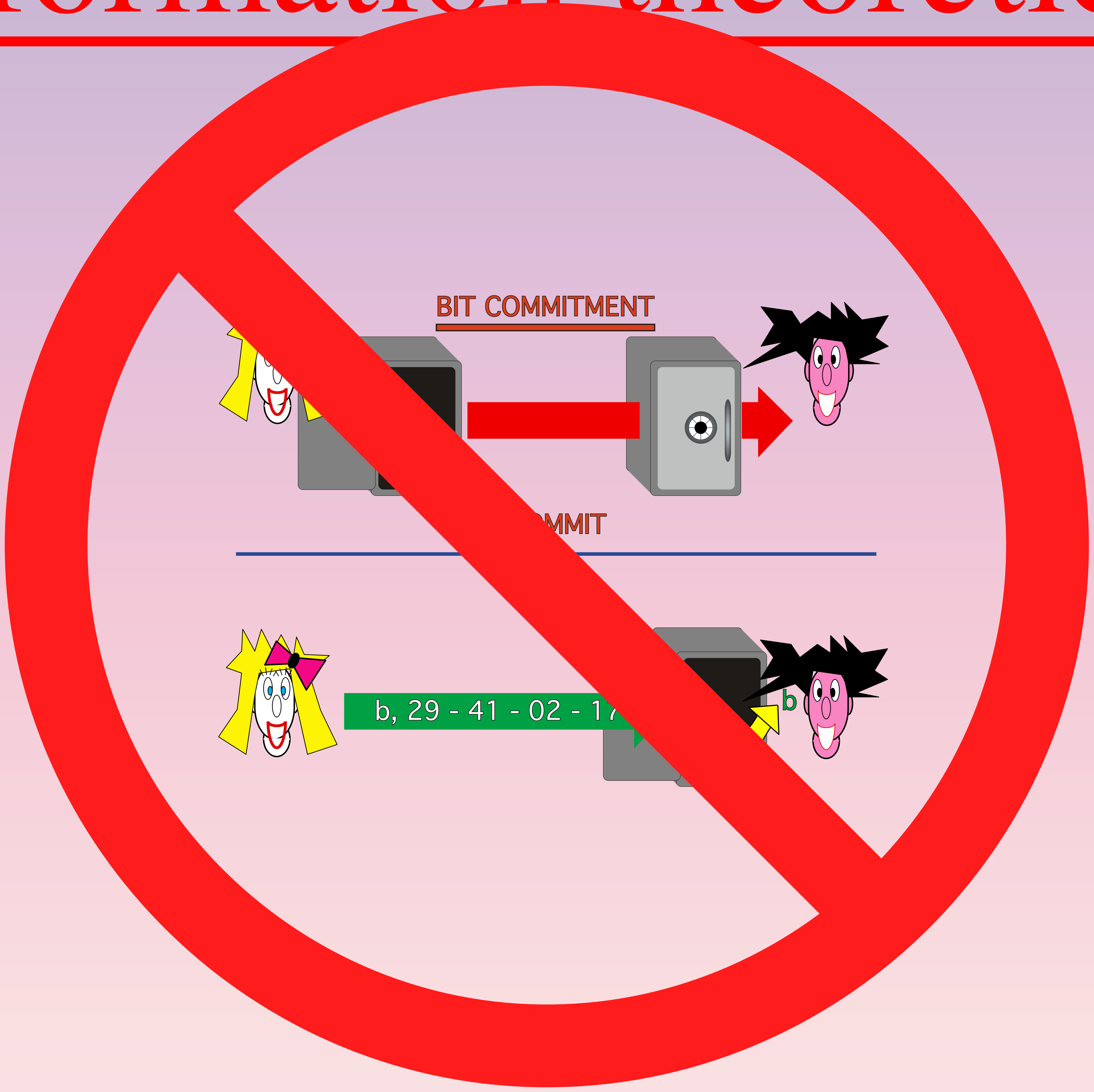


Classically (information theoretical)



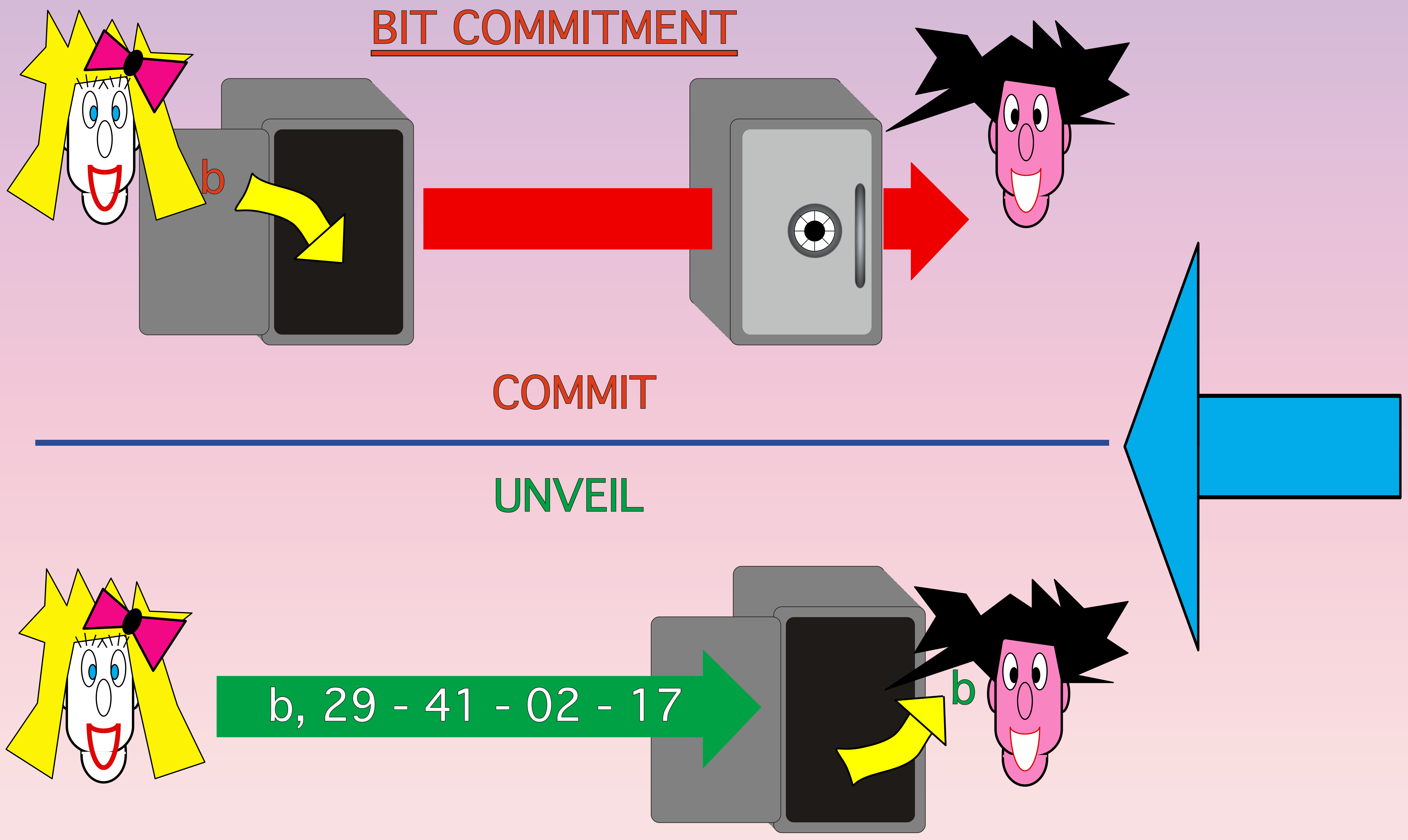
Folklore

Quantumly (information theoretical)

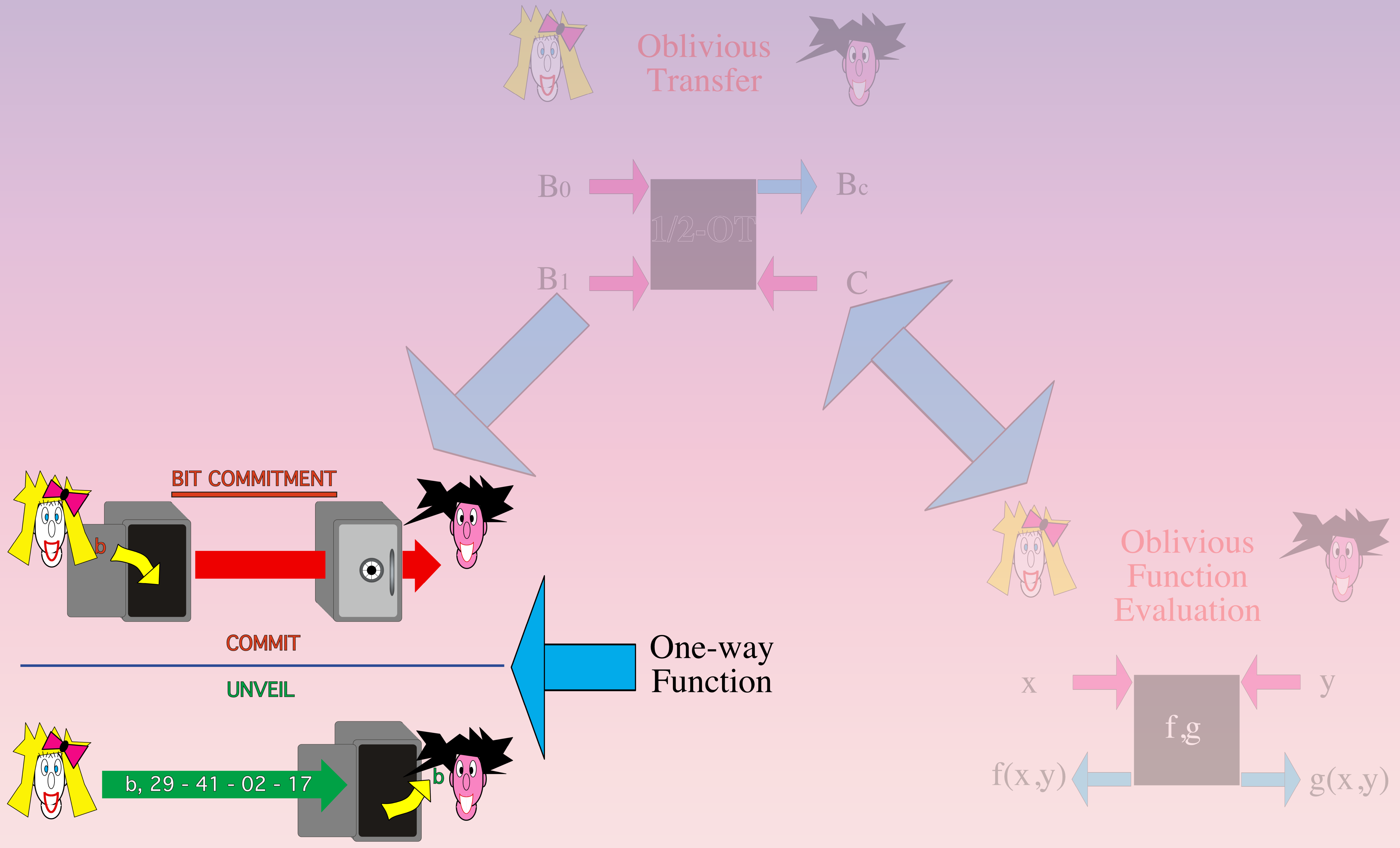


Mayers, Lo-Chau

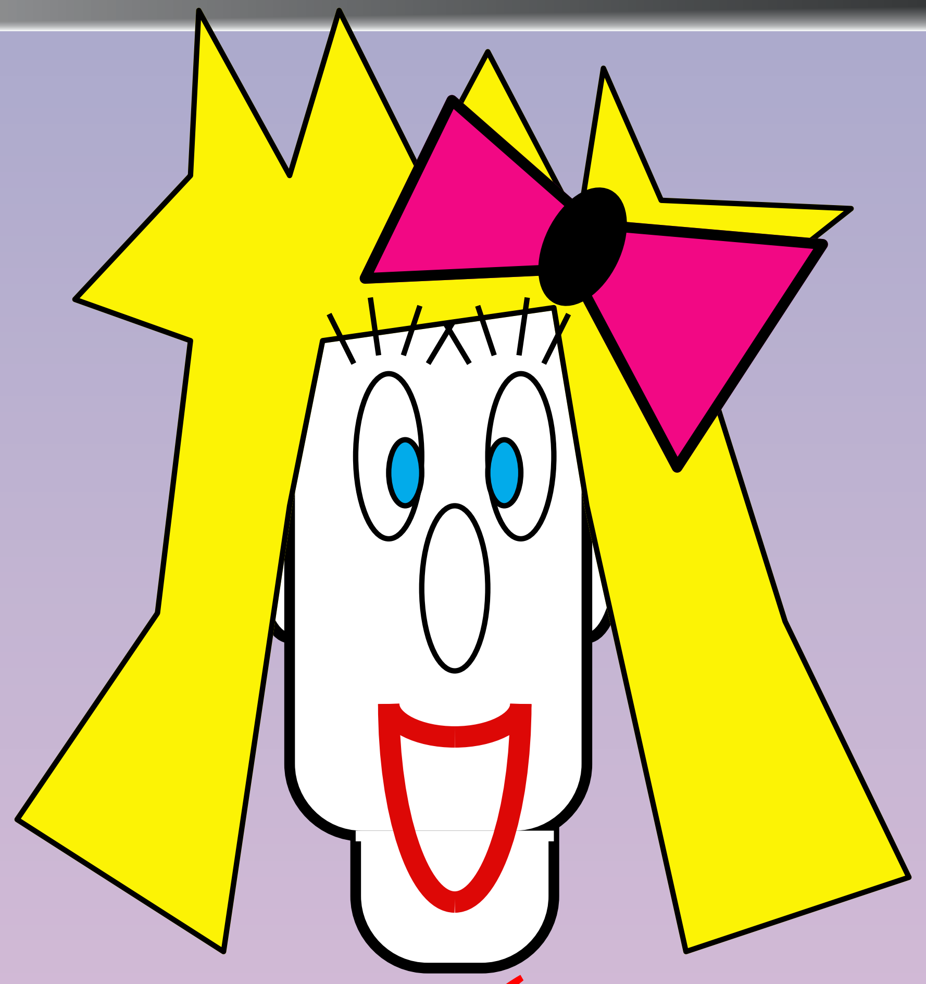
Classically

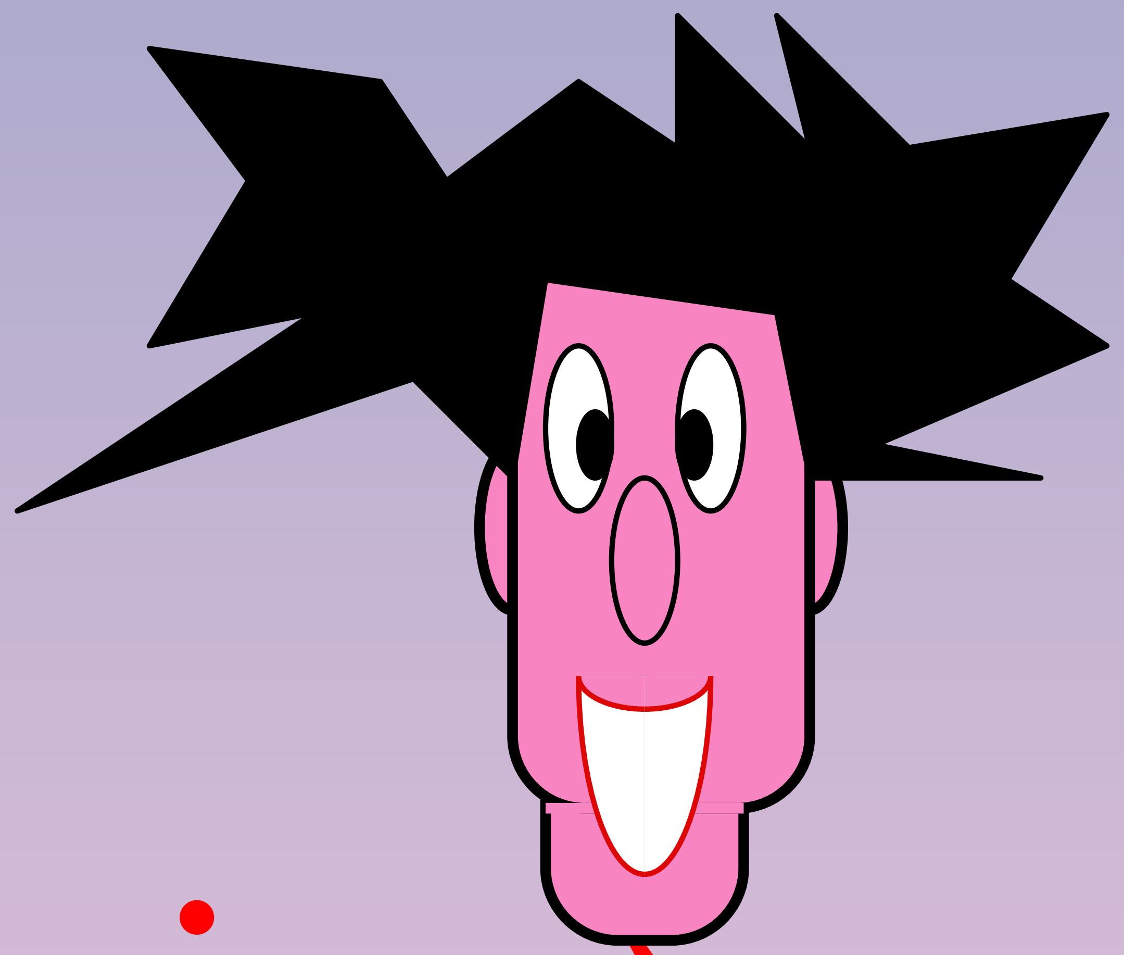


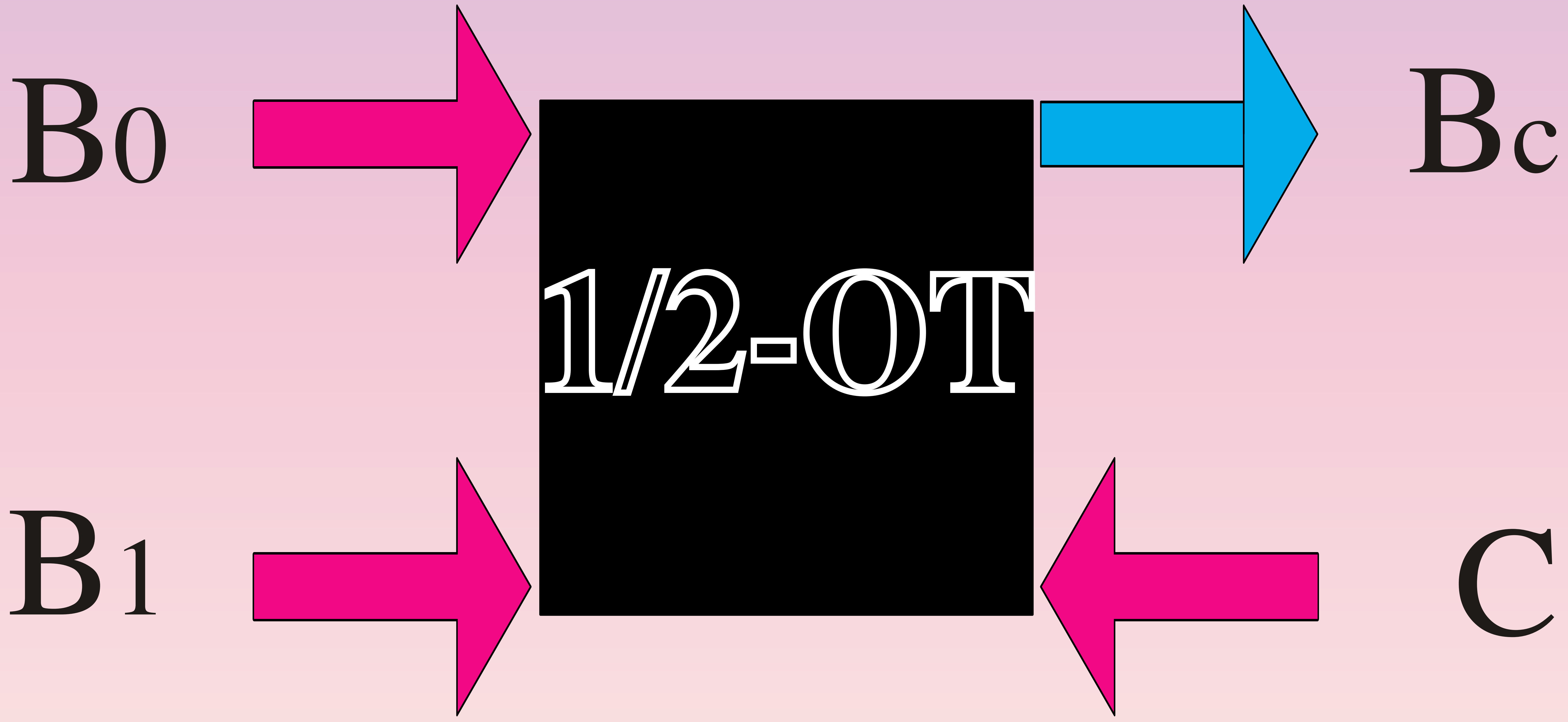
Classically



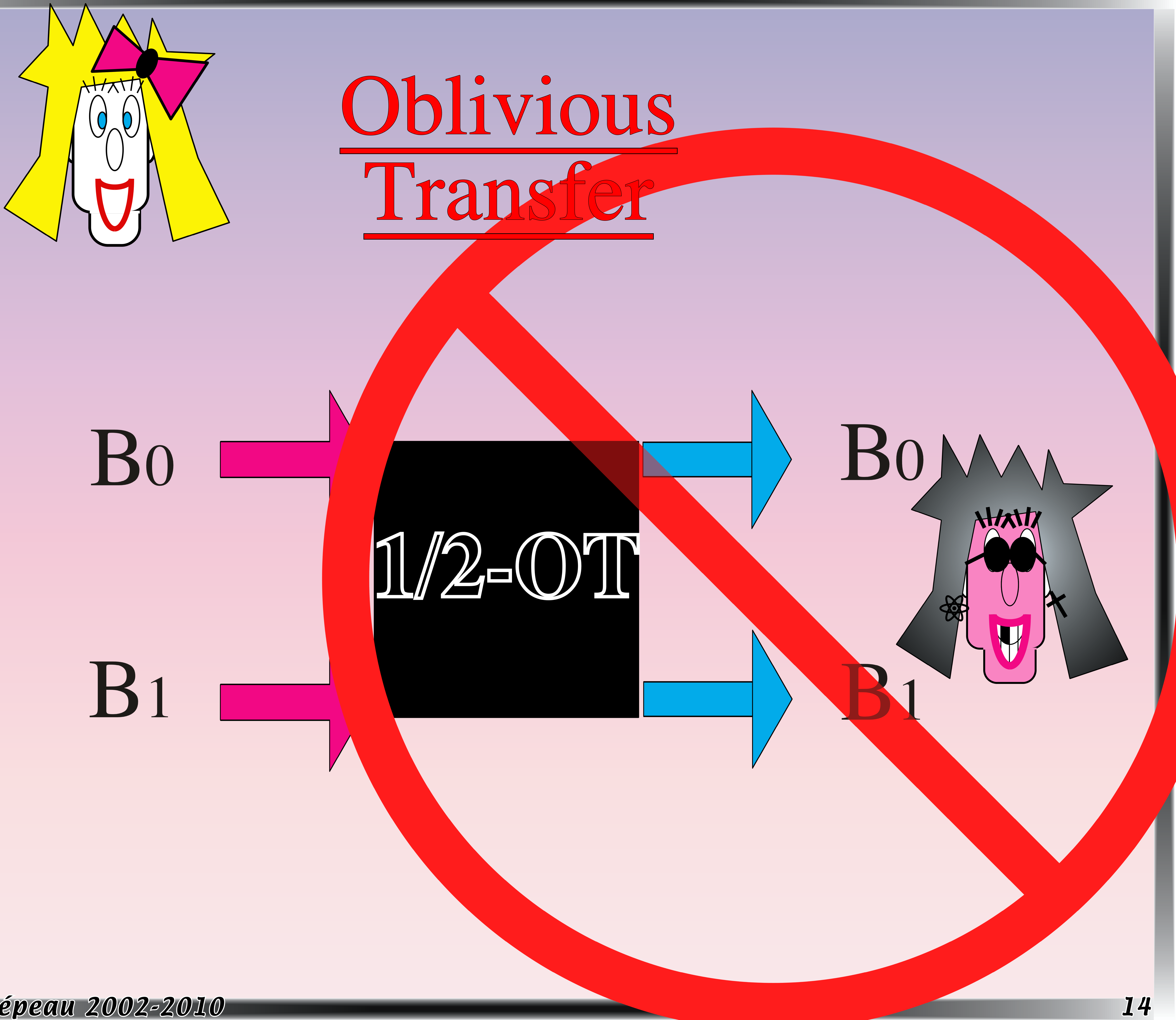
Oblivious Transfer (message multiplexing)



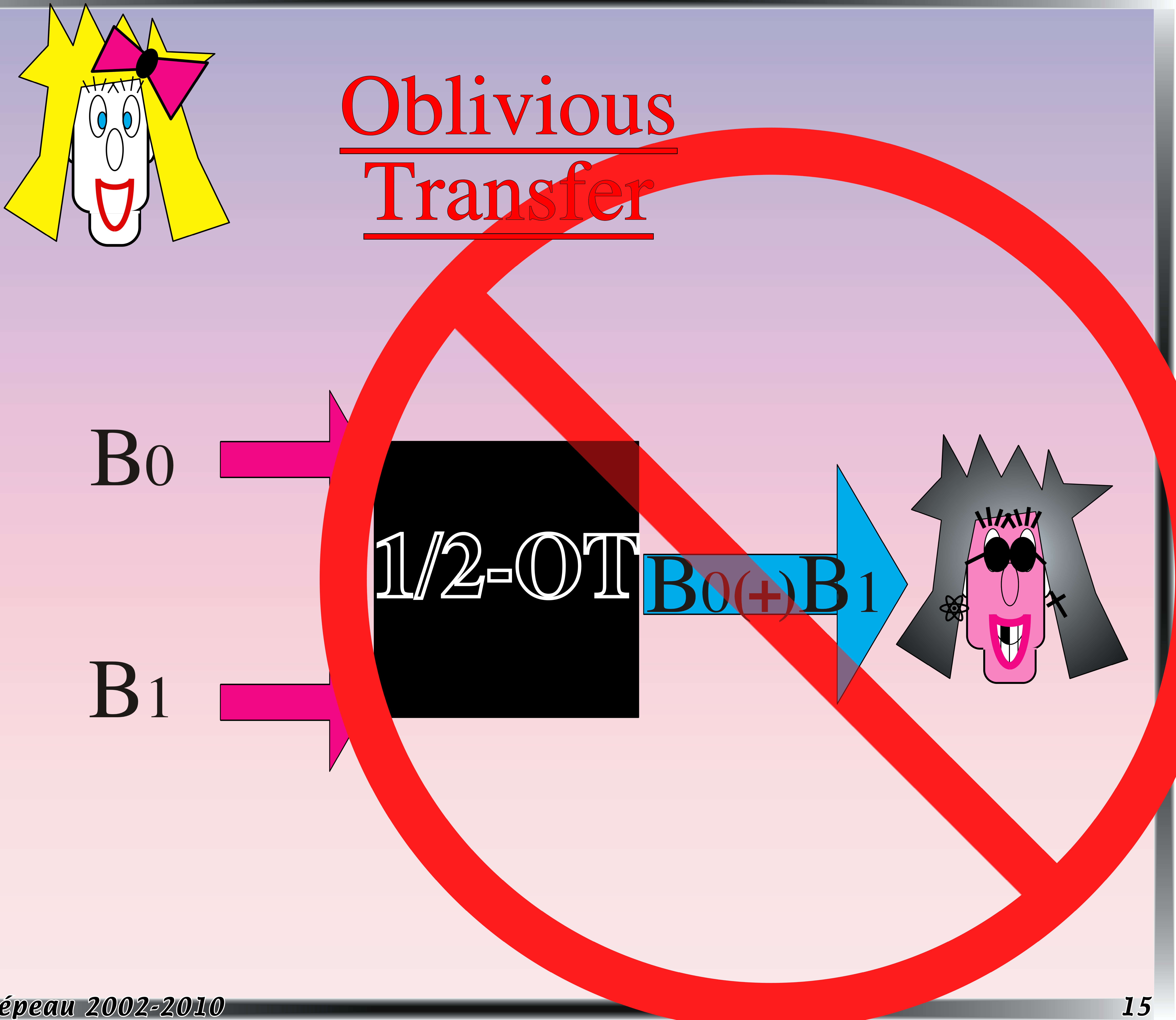




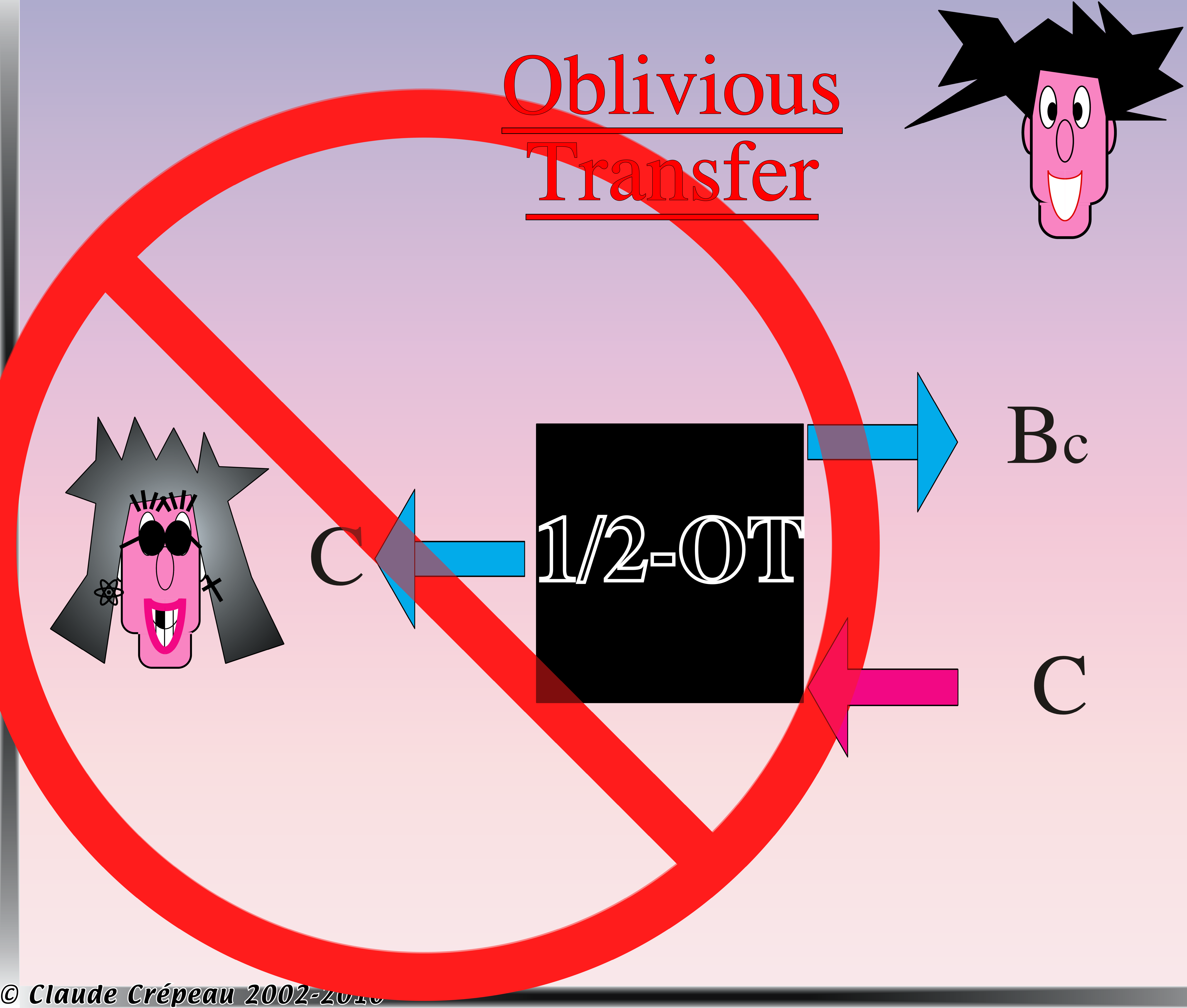
Oblivious Transfer

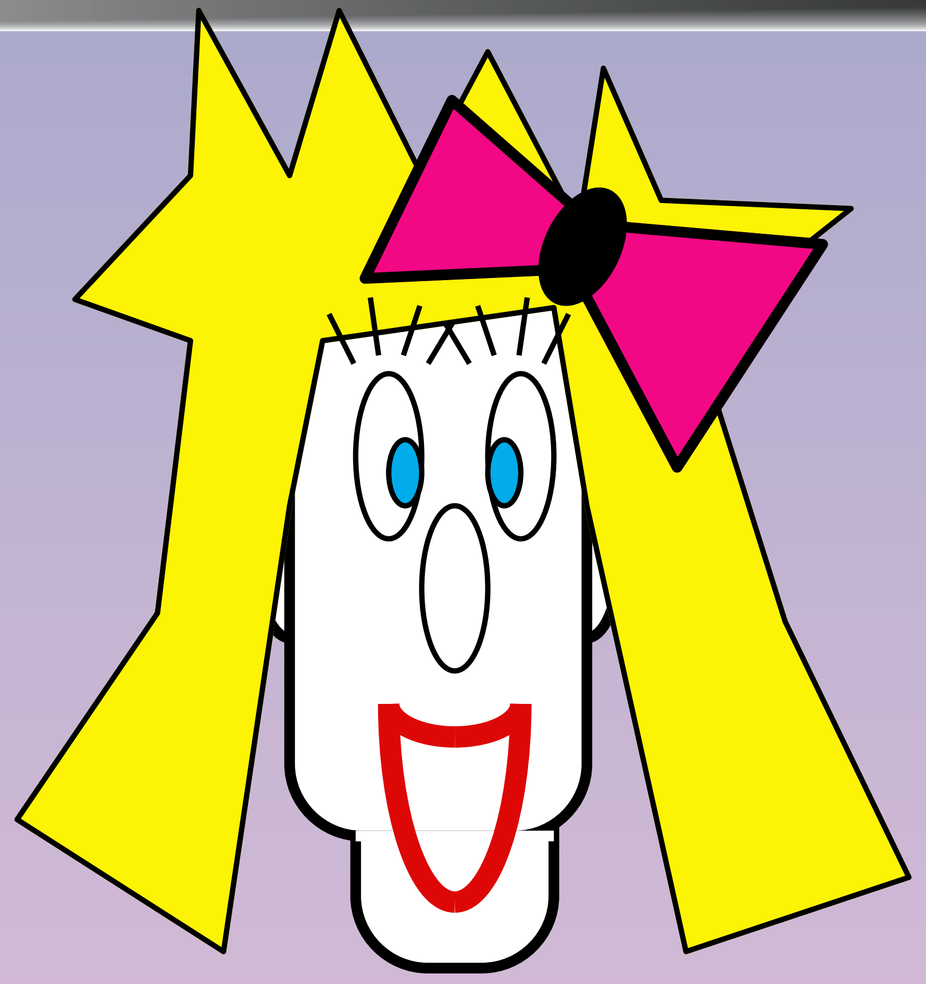


Oblivious Transfer

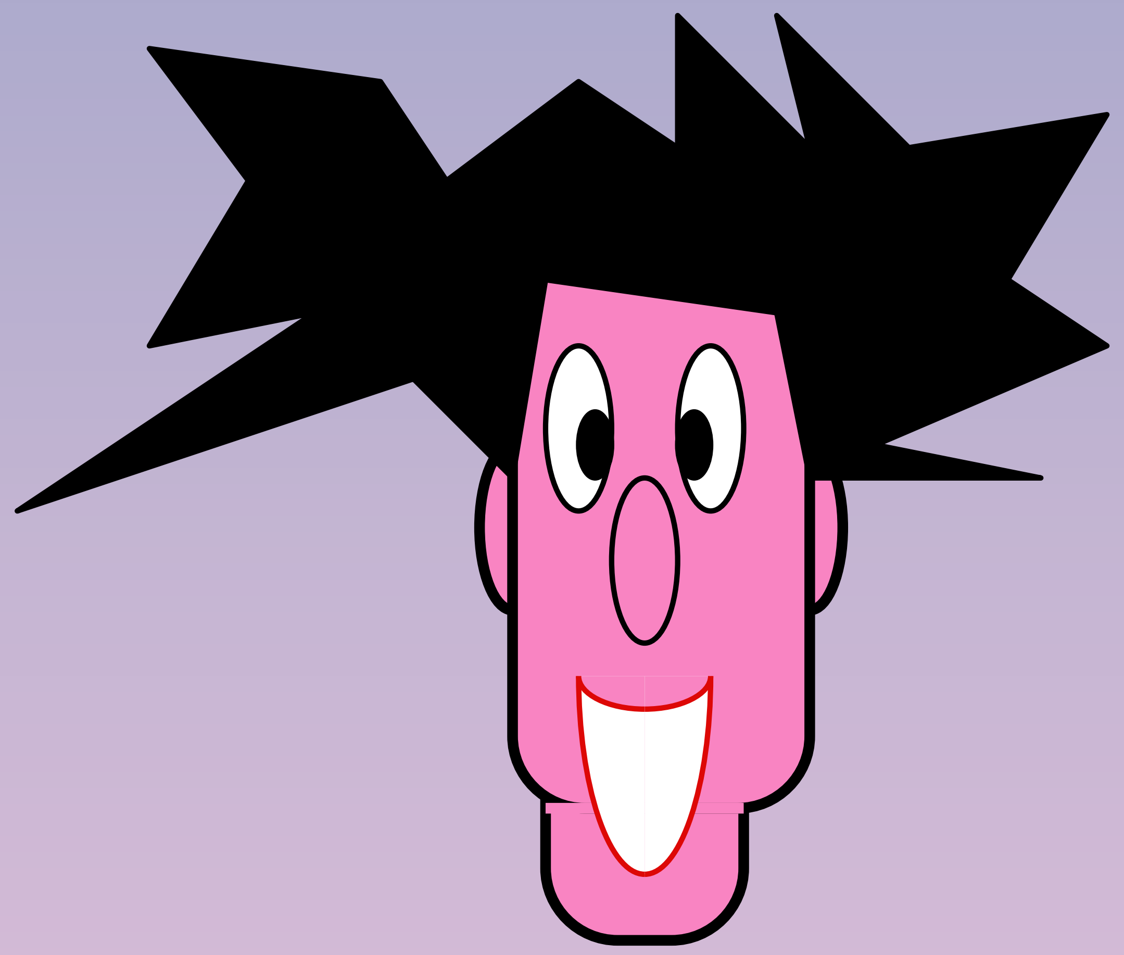


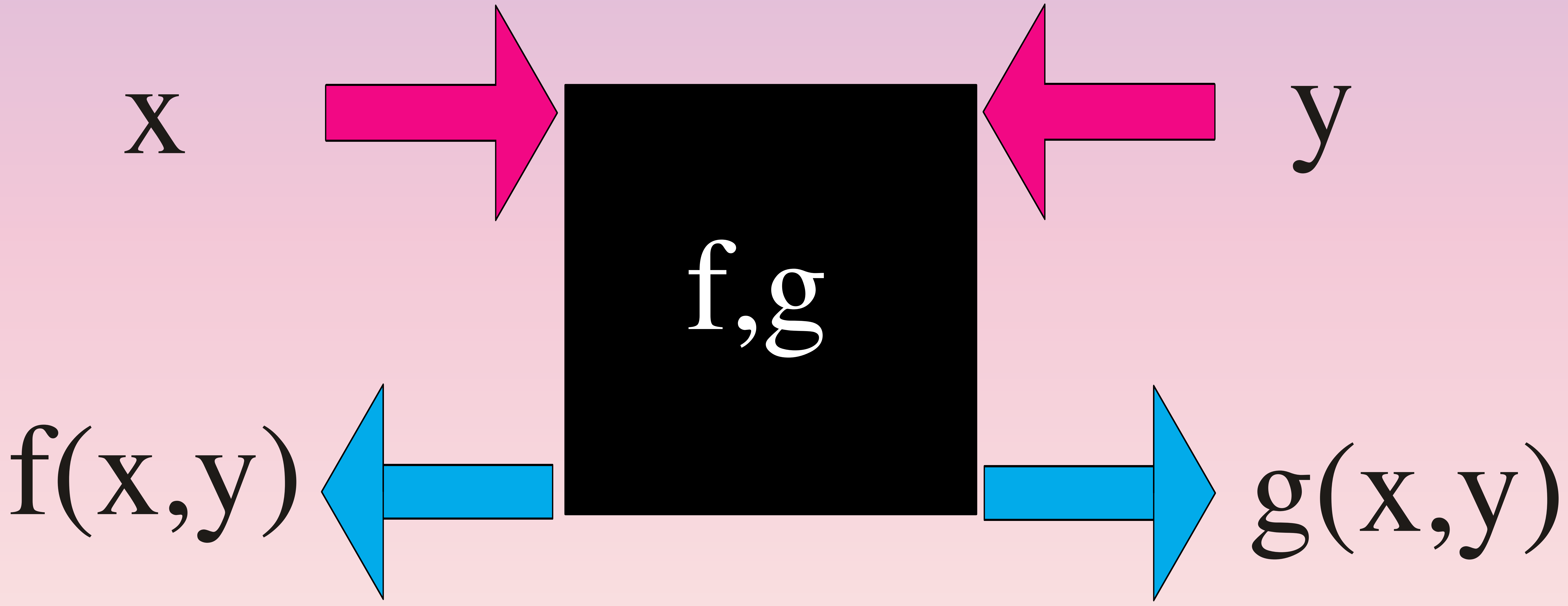
Oblivious Transfer



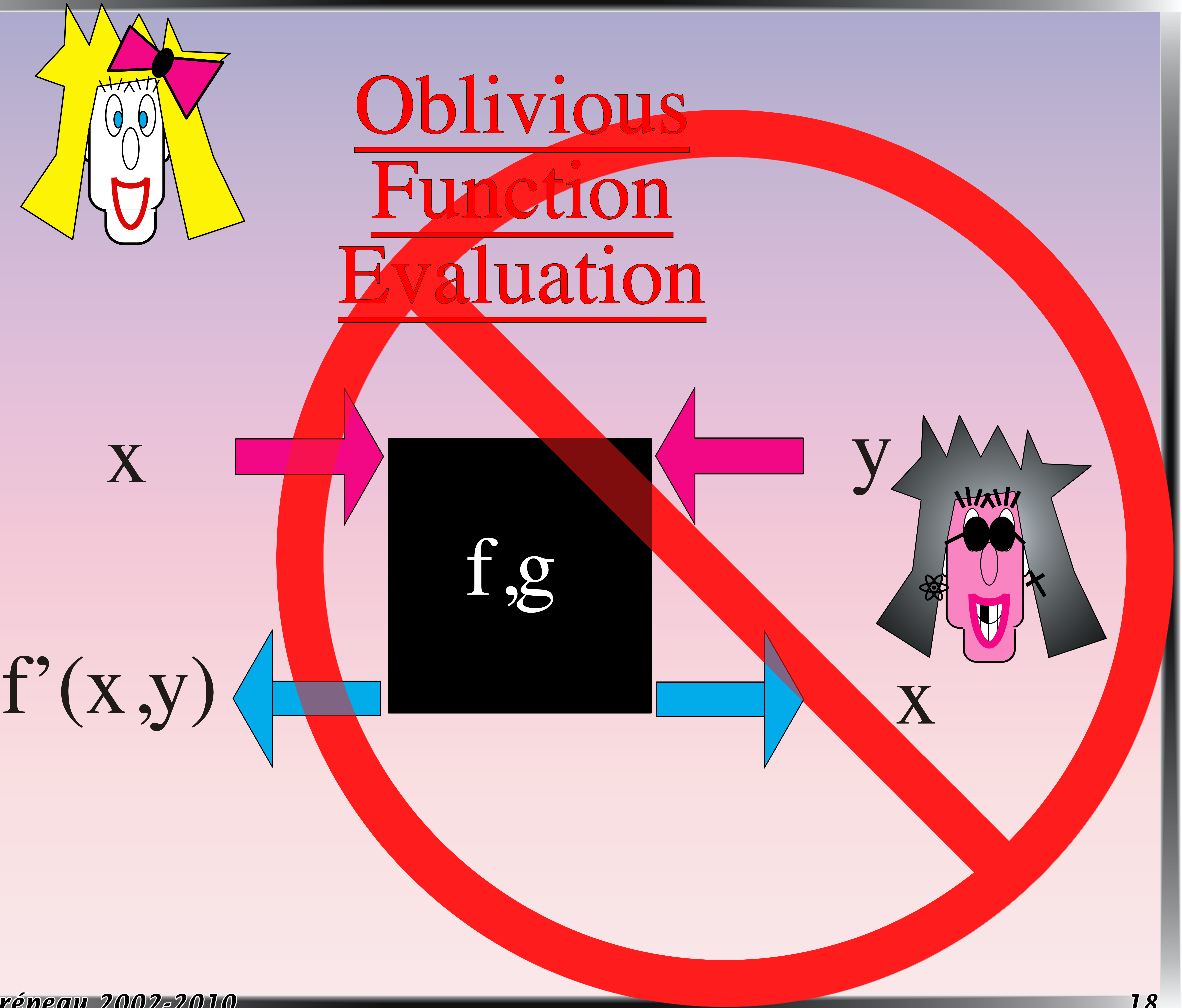


Oblivious Function Evaluation

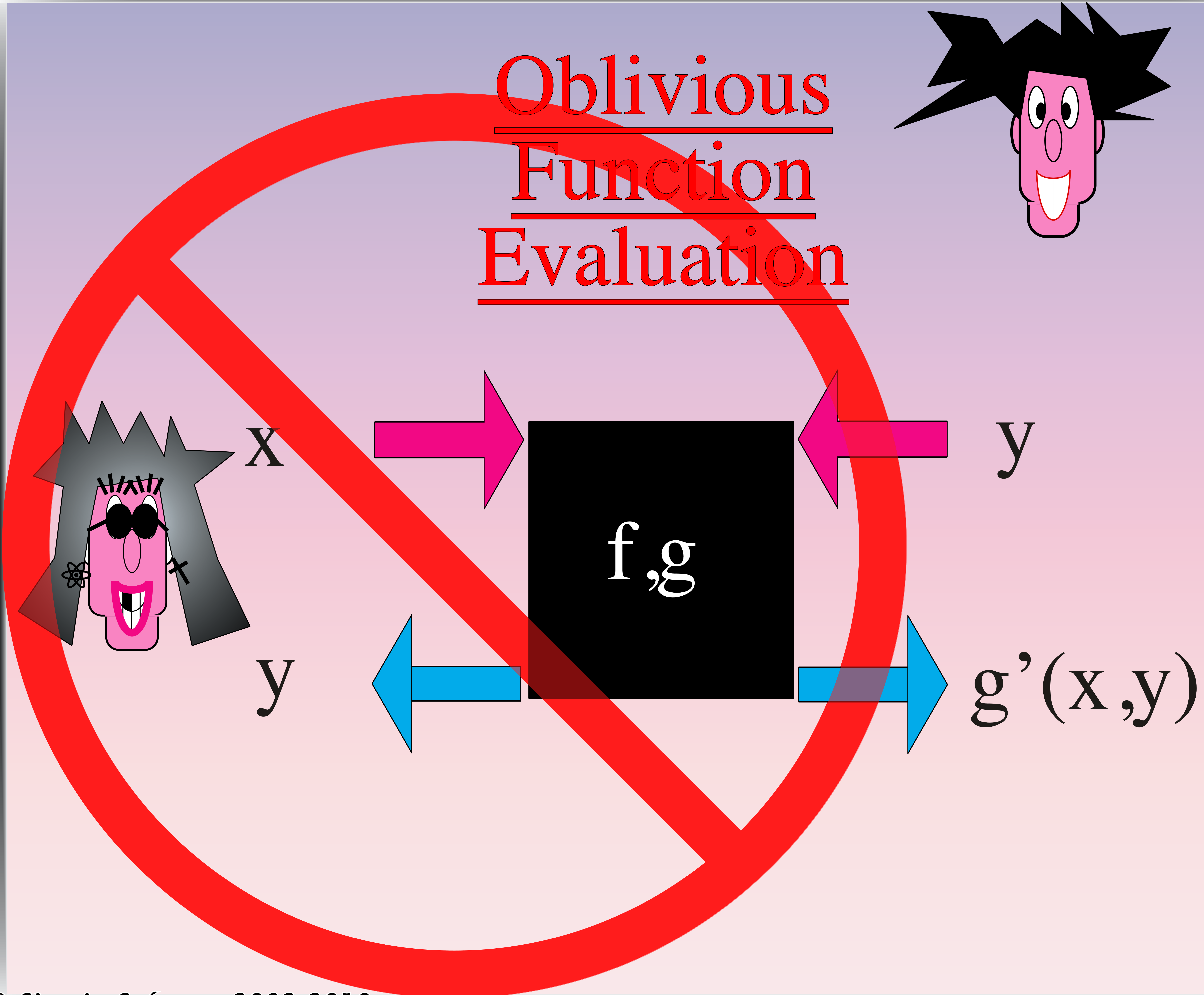


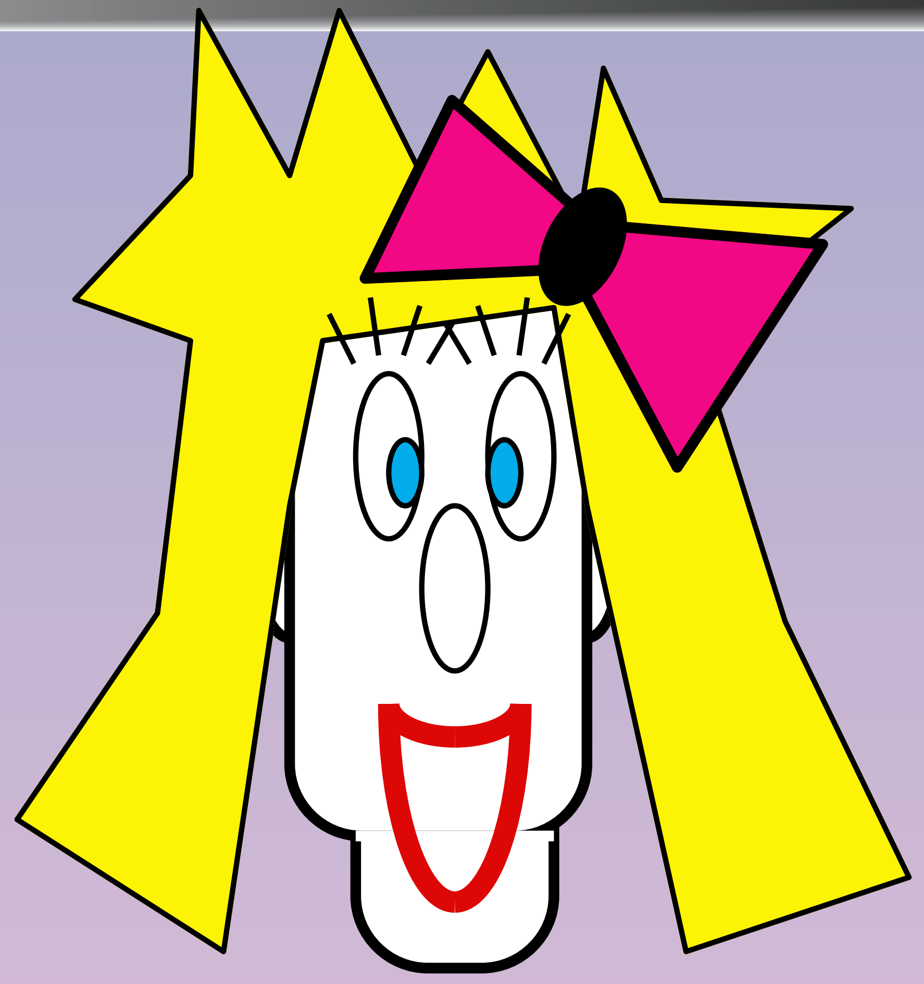


Oblivious Function Evaluation

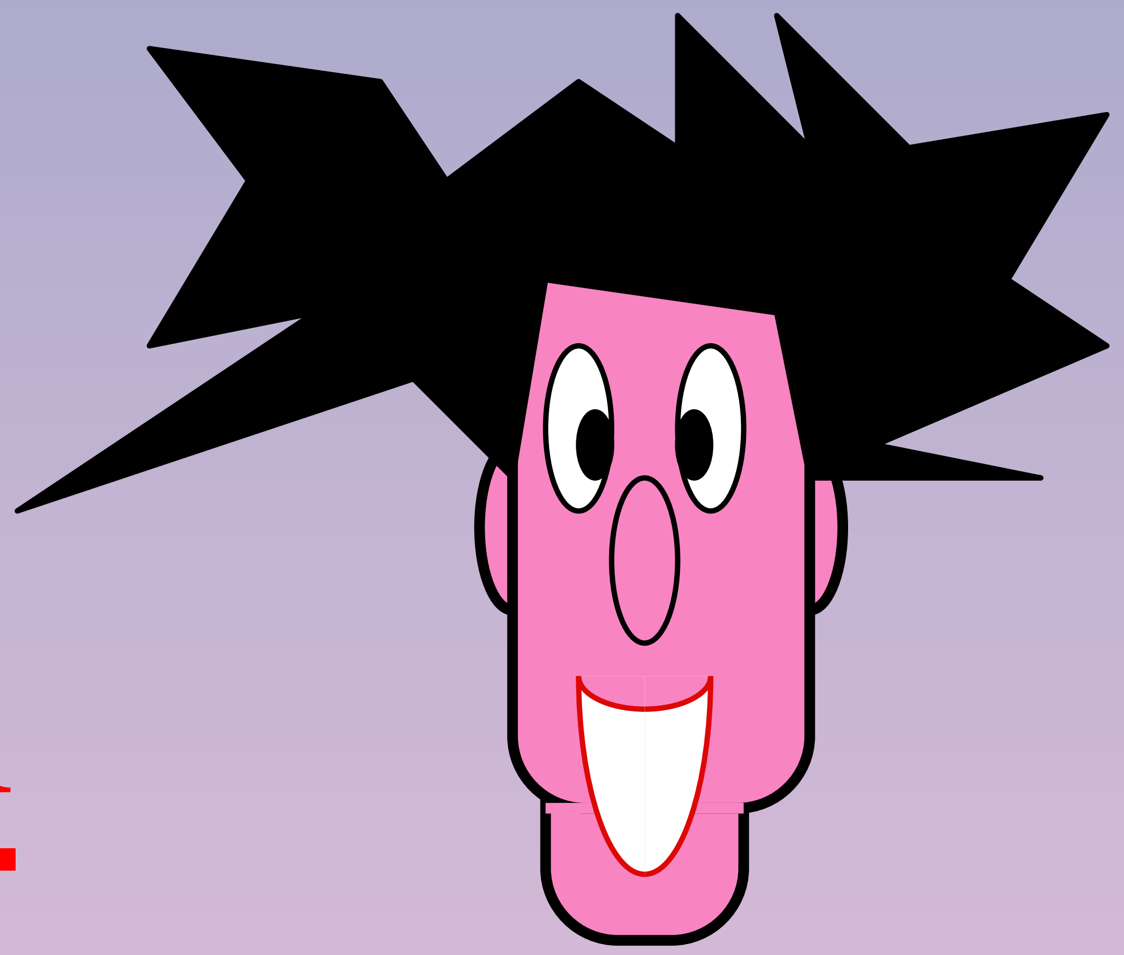


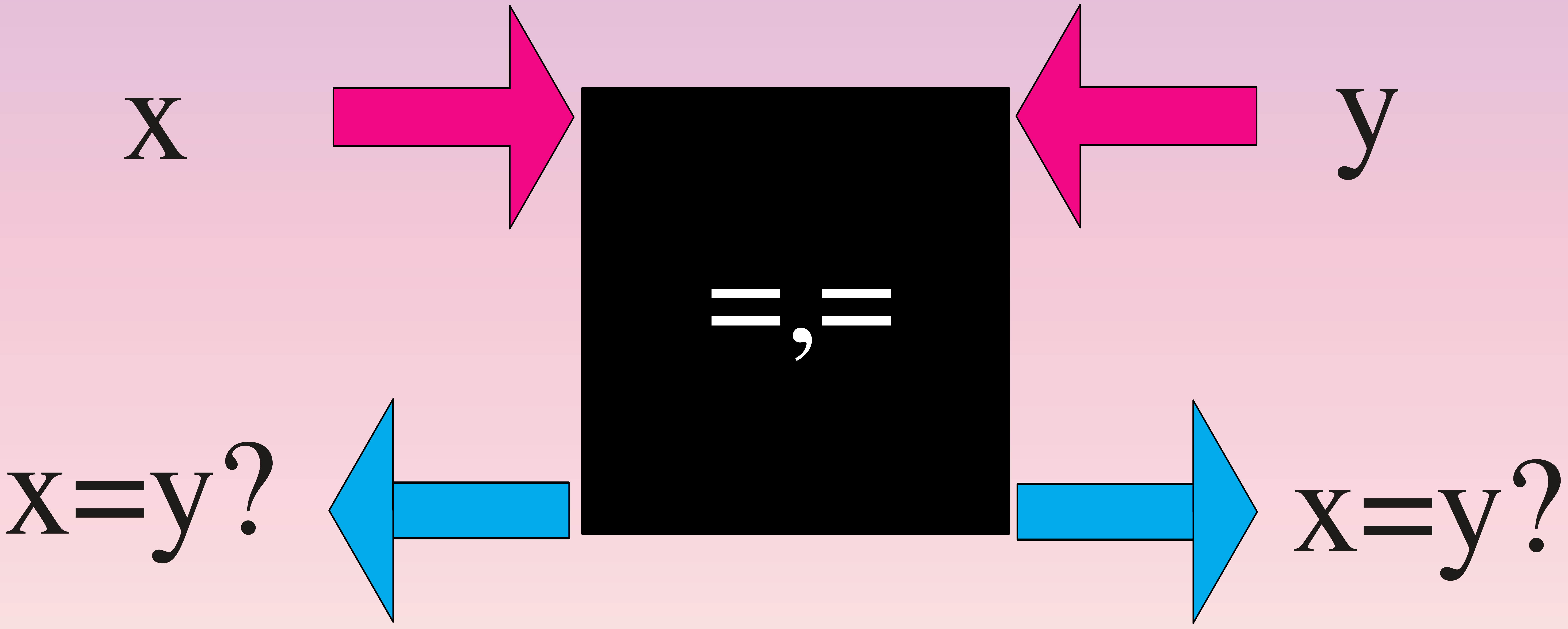
Oblivious Function Evaluation

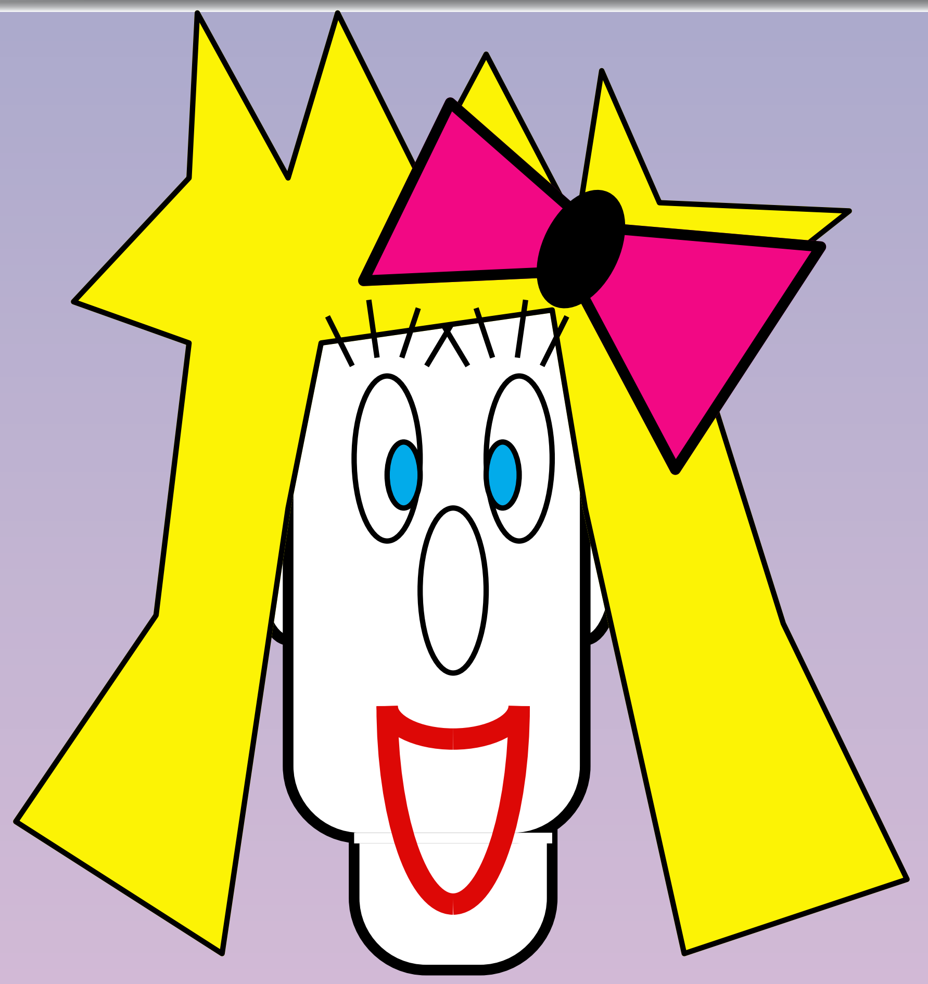




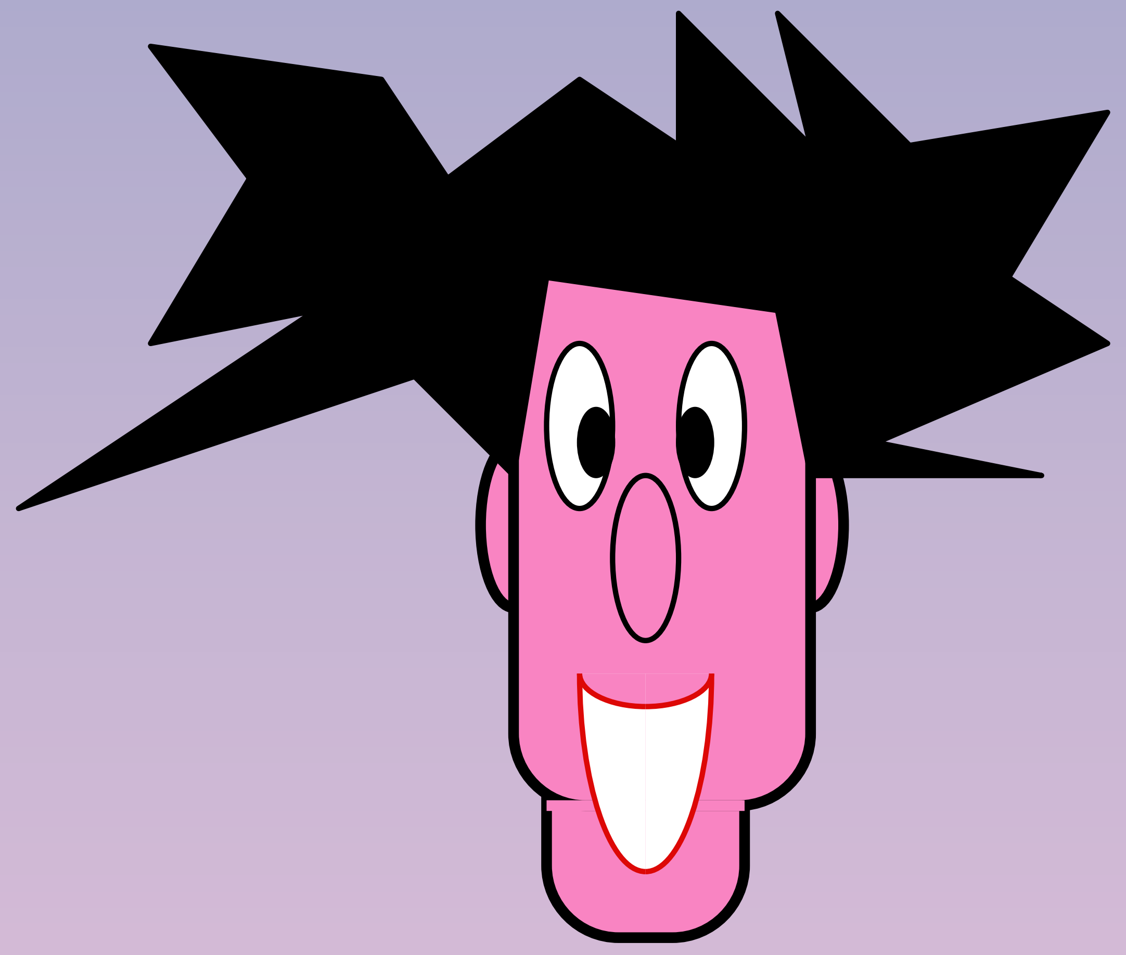
Mutual Identification

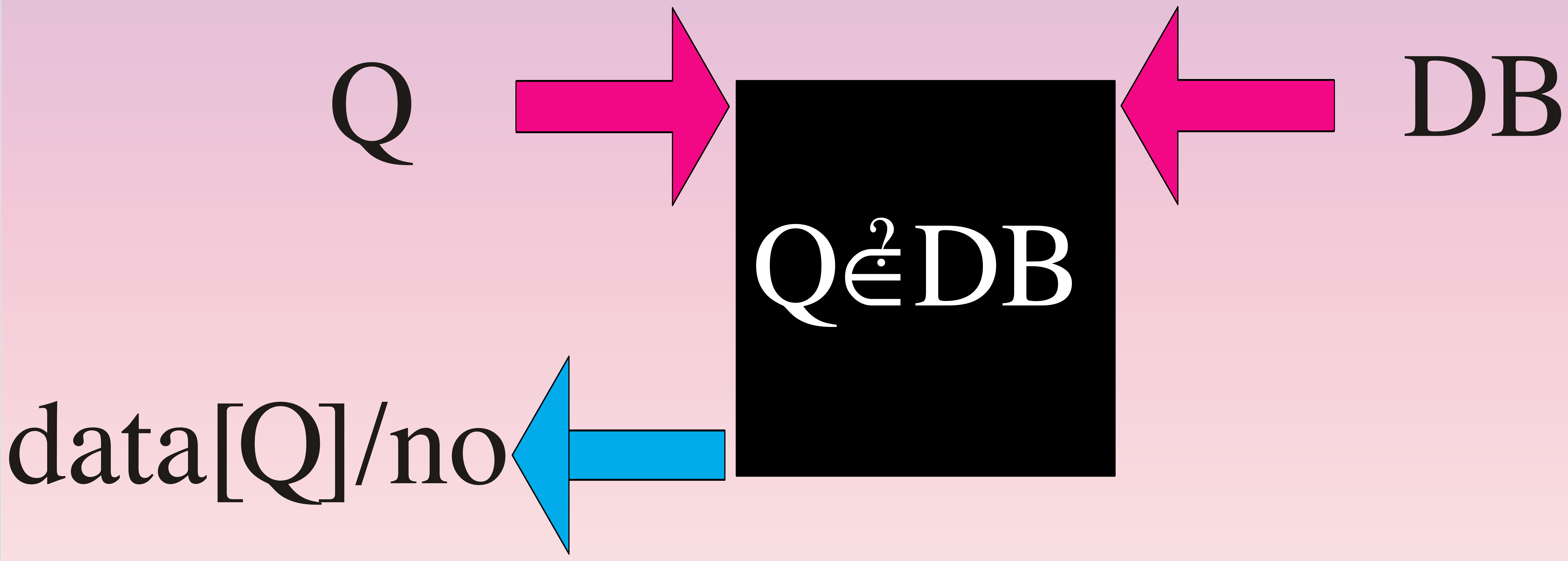




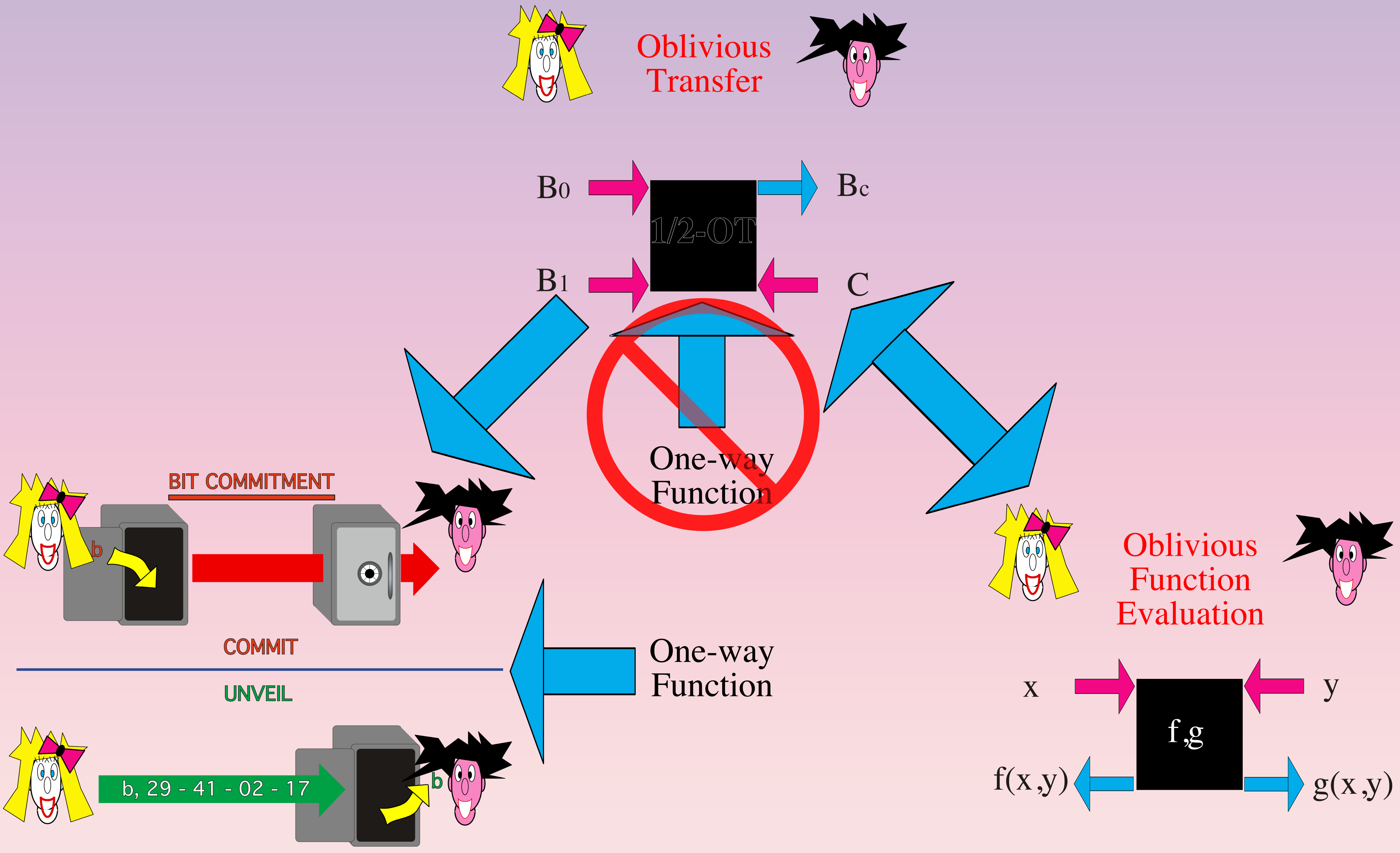


Oblivious DB query



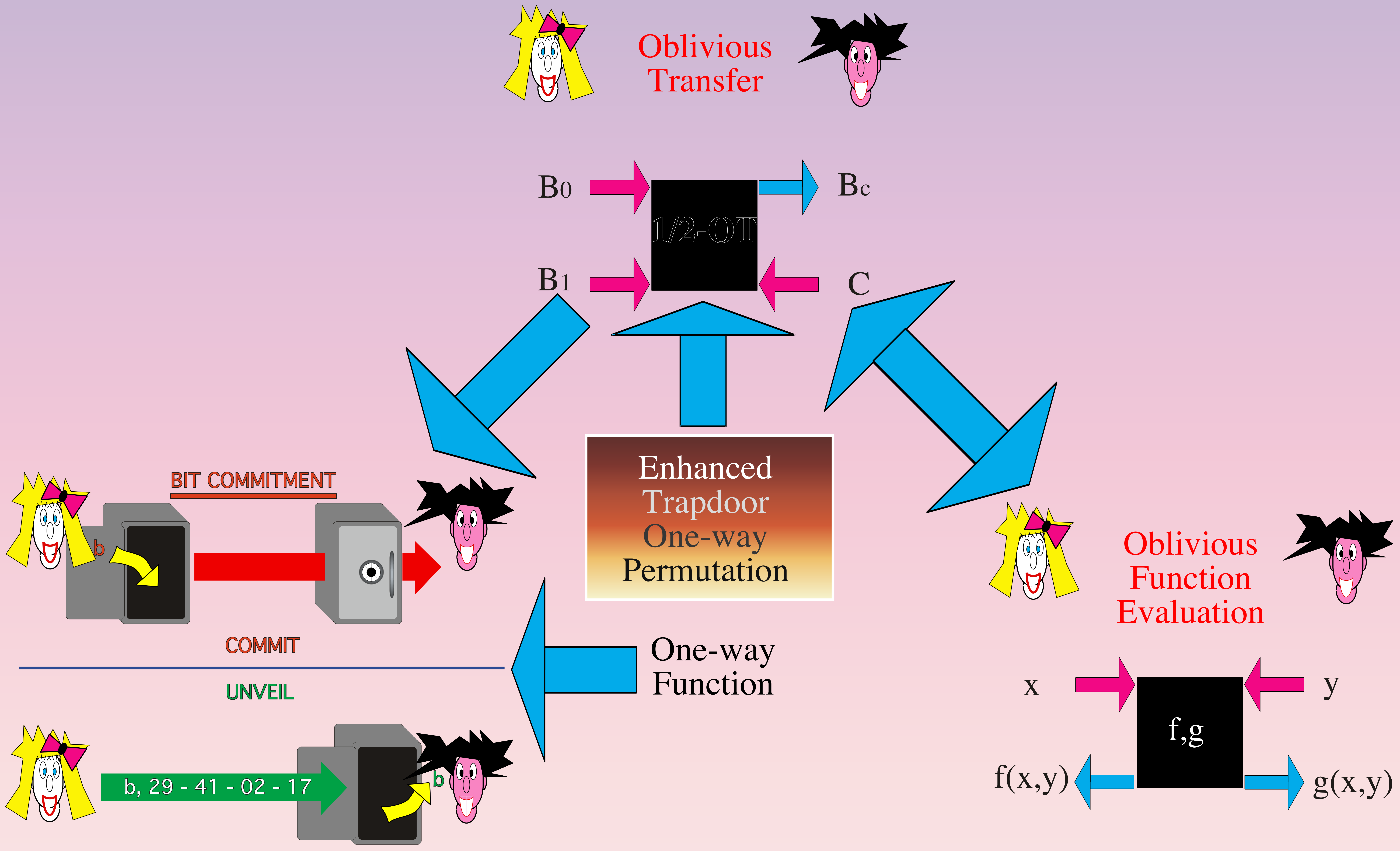


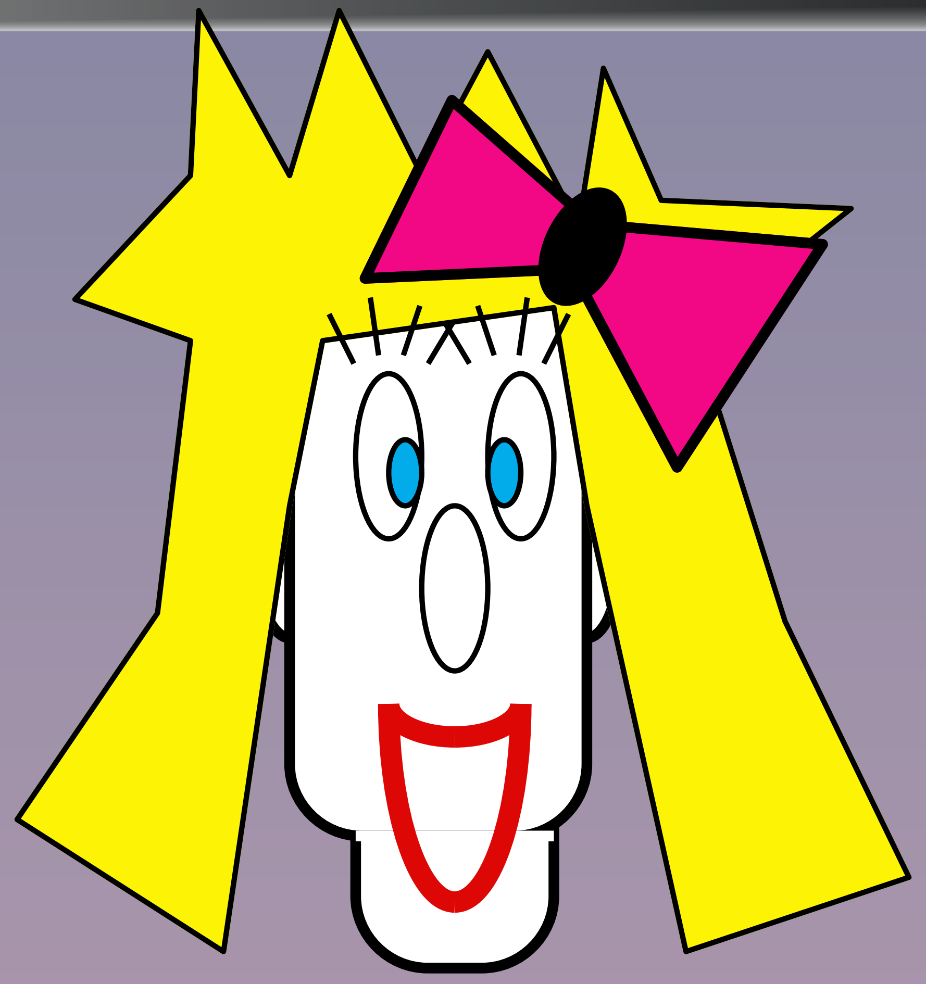
Classically



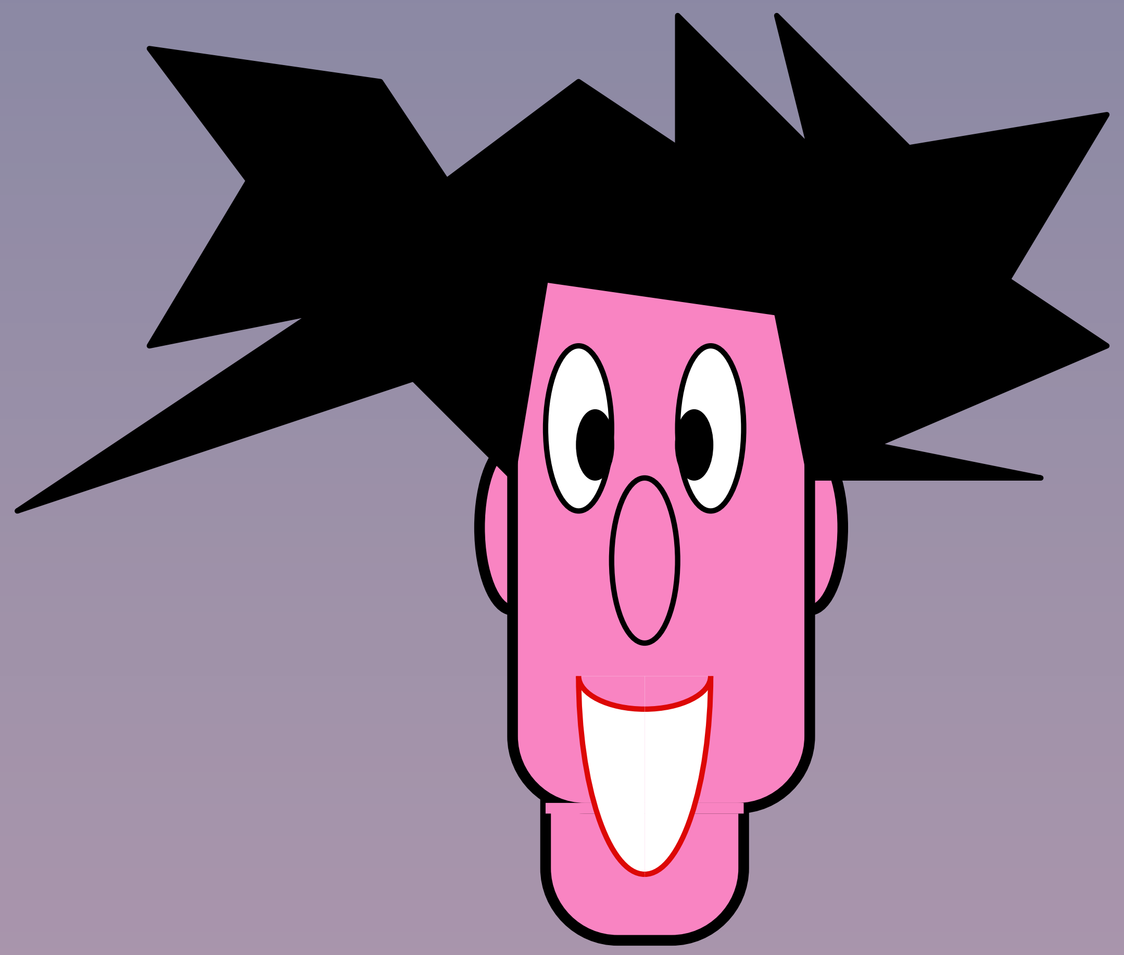
(2) Secure OT Implementations

Classically





[EGL85] [GMW87]

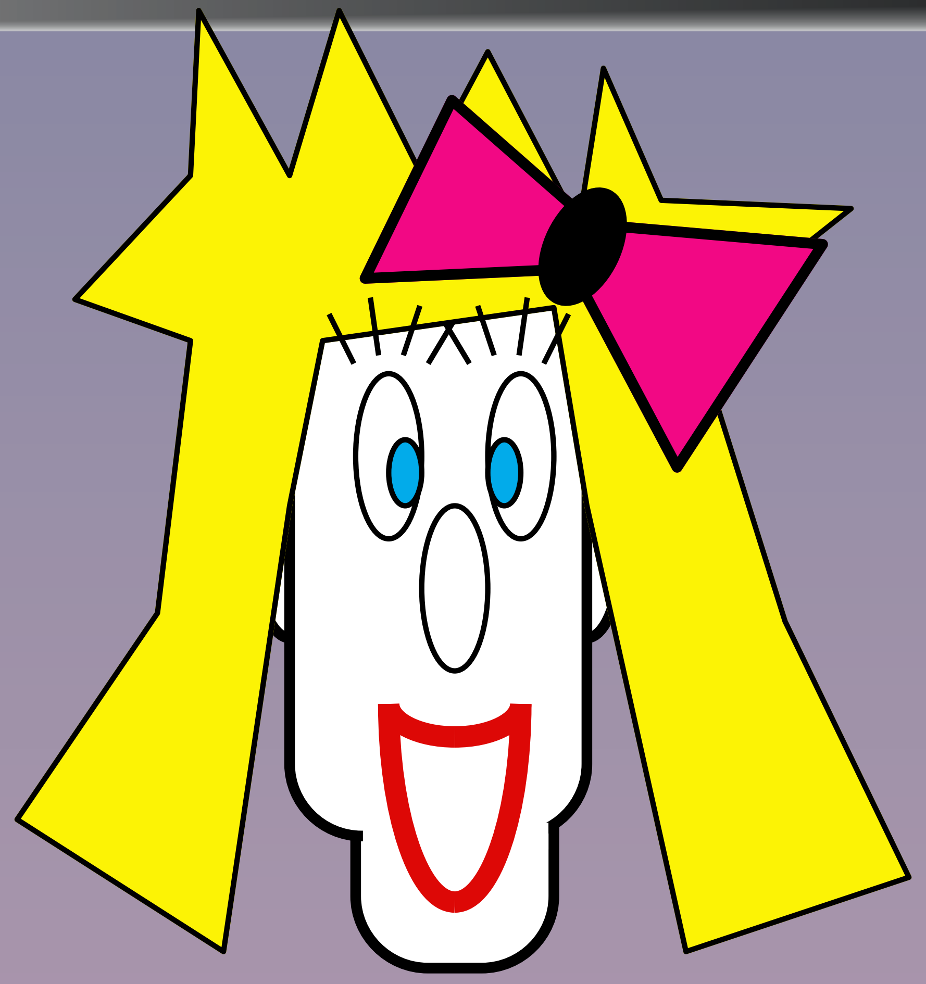


INGREDIENTS

a public-key block cipher:

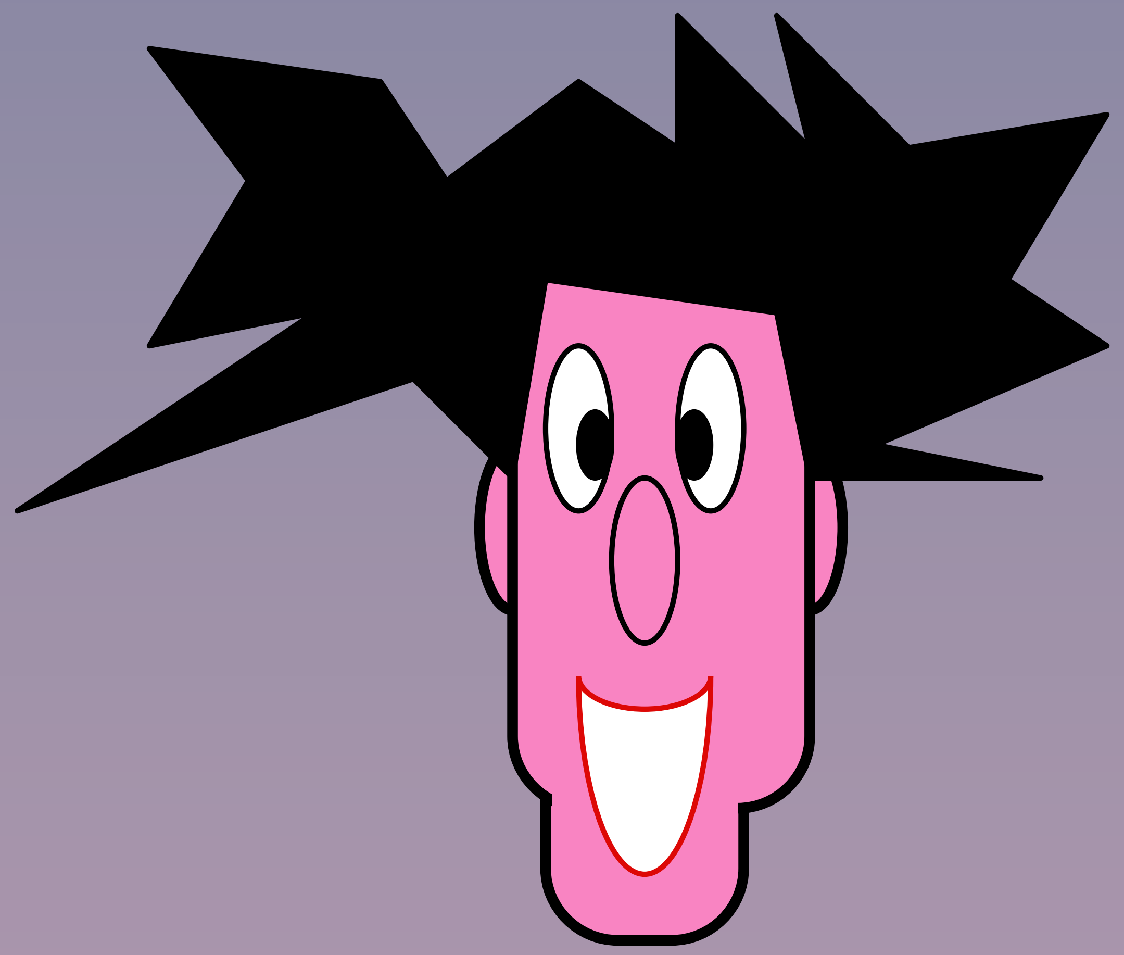
$(\mathrm{enc}_B, \mathrm{dec}_B)$

a public predicate: π



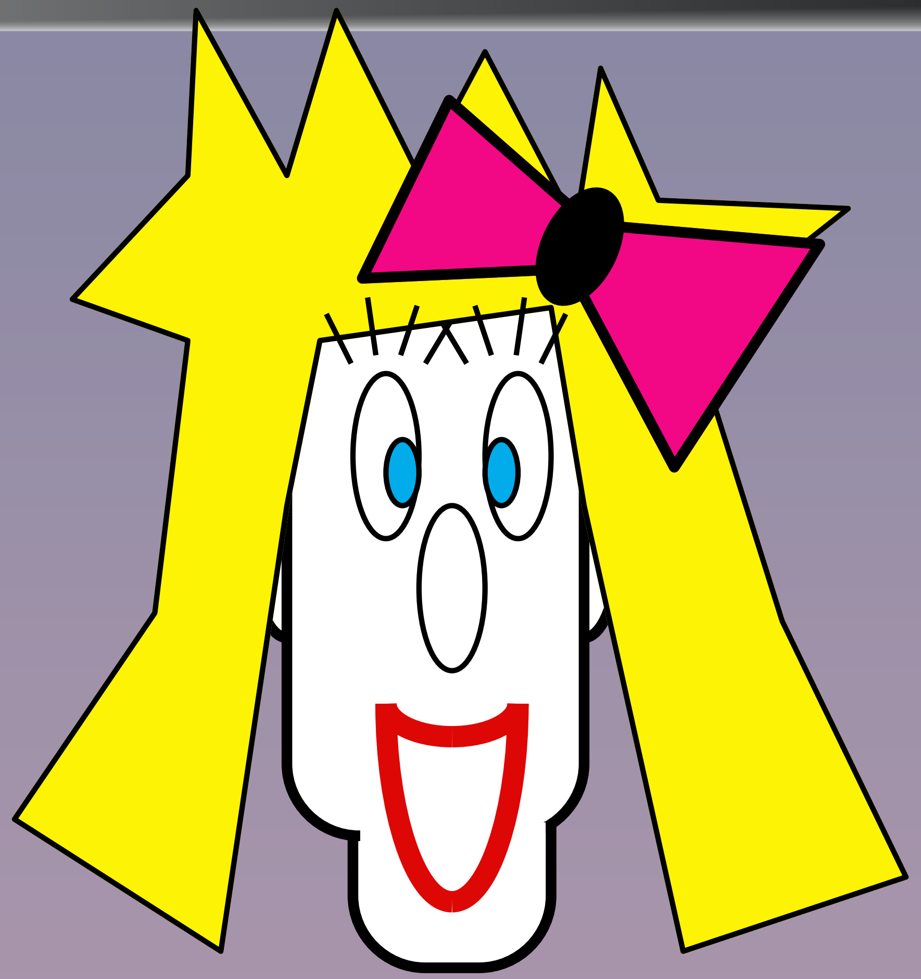
1/2-OT

[EGL85] [GMW87]

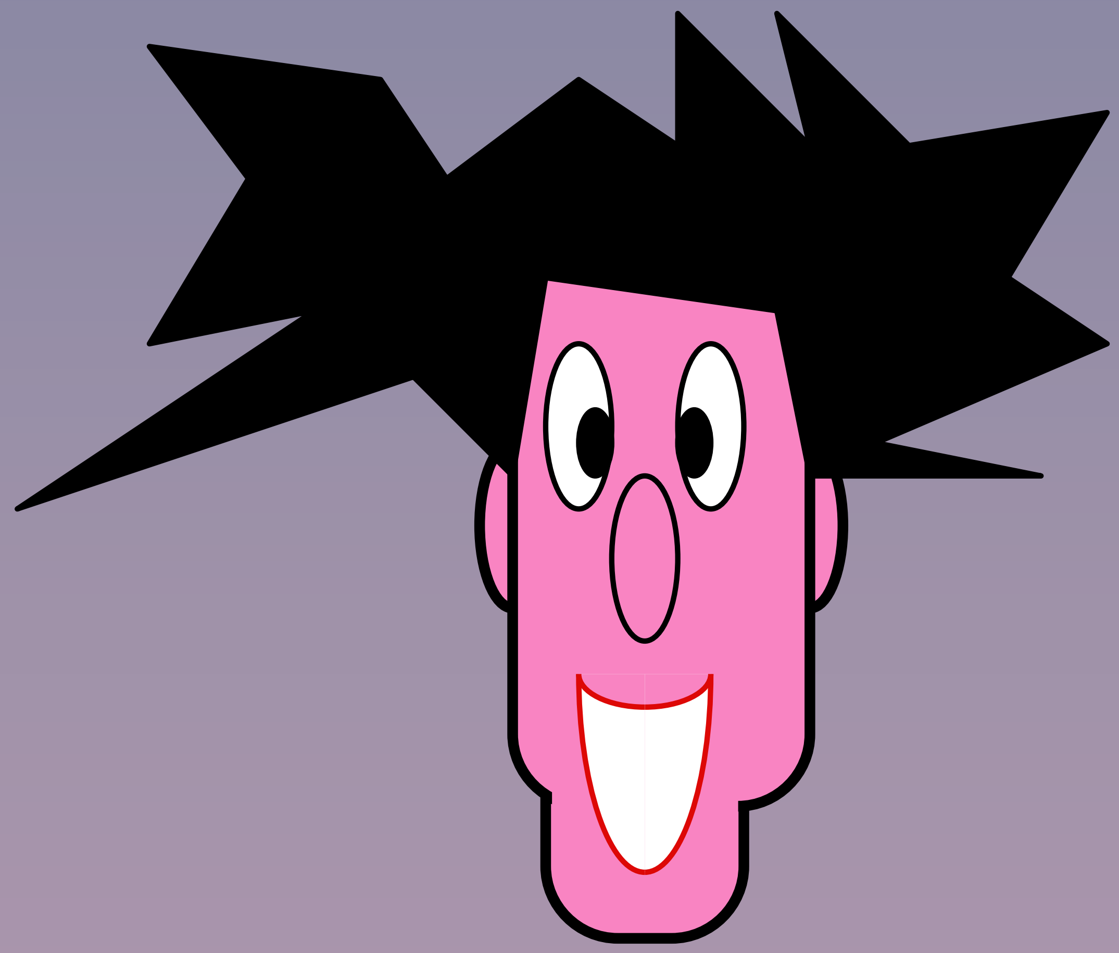


$$\mathrm{enc}_B(®_0) \longrightarrow U_0$$

$$®_1 \longrightarrow U_1$$



[EGL85] [GMW87]

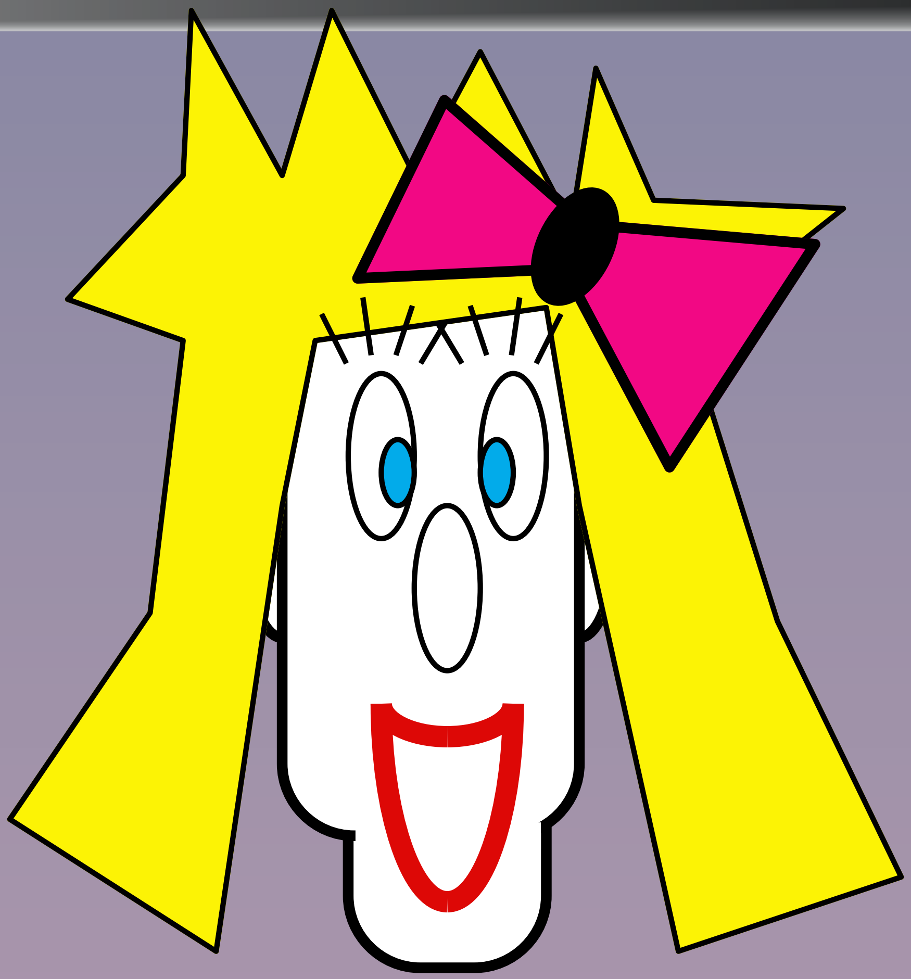


$\text{enc}_B(\text{®}0) \longrightarrow U0$

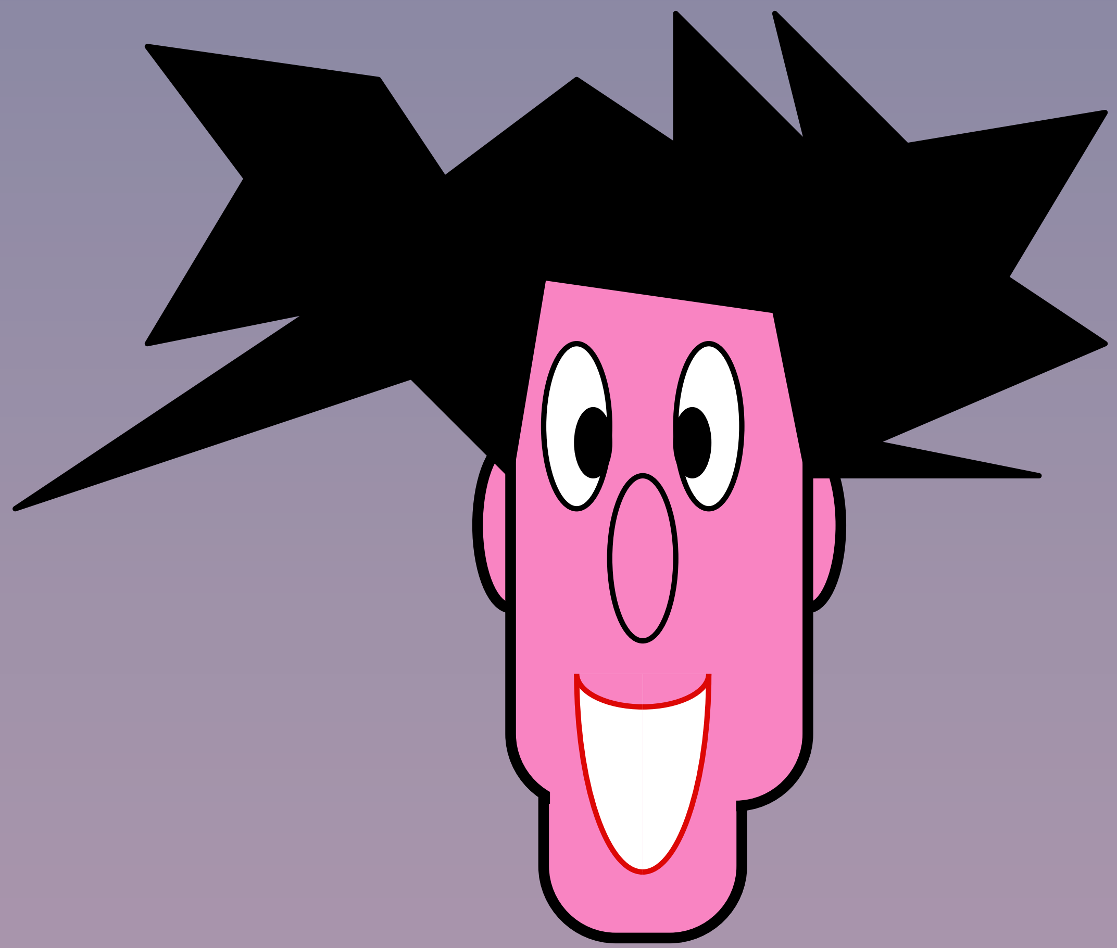
$\text{®}1 \longrightarrow U1$

$Z_0 \longleftarrow \pi(\text{dec}_B(U0))(+)B0$

$Z_1 \longleftarrow \pi(\text{dec}_B(U1))(+)B1$



[EGL85] [GMW87]



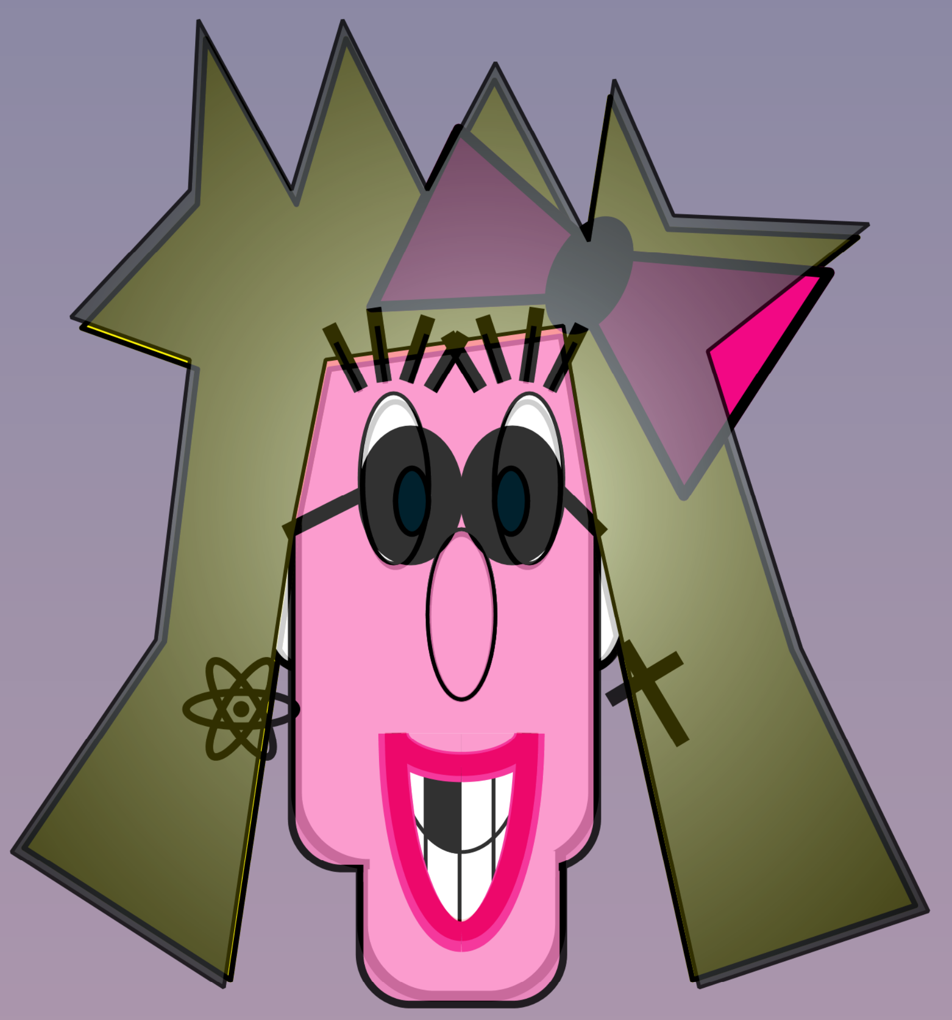
$$\mathrm{enc}_B(\circledR 0) \longrightarrow U0$$

$$\circledR 1 \longrightarrow U1$$

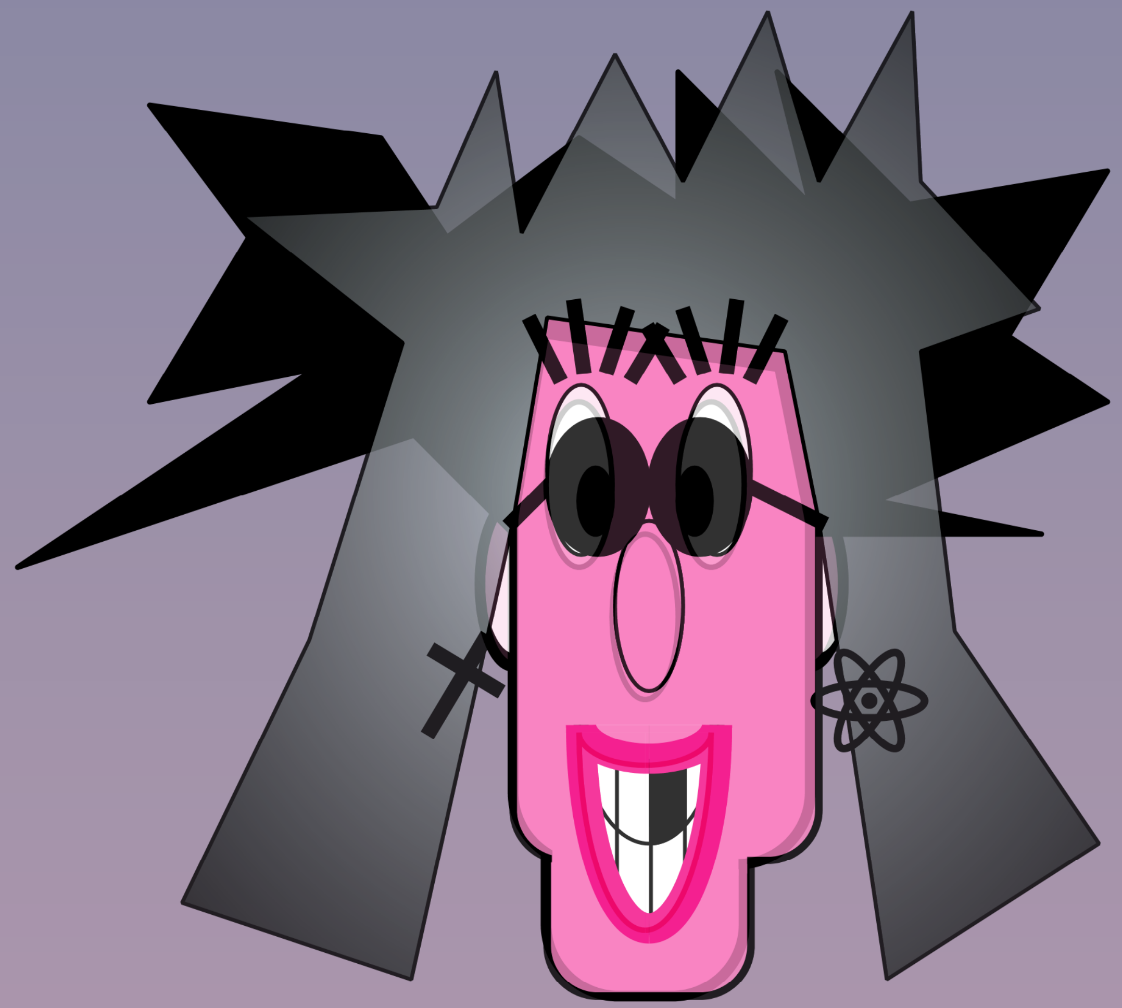
$$Z_0 \longleftarrow \pi(\mathrm{dec}_B(U0))(+)B0$$

$$Z_1 \longleftarrow \pi(\mathrm{dec}_B(U1))(+)B1$$

$$B0=\pi(\circledR 0)(+)Z0$$



[GMW87]



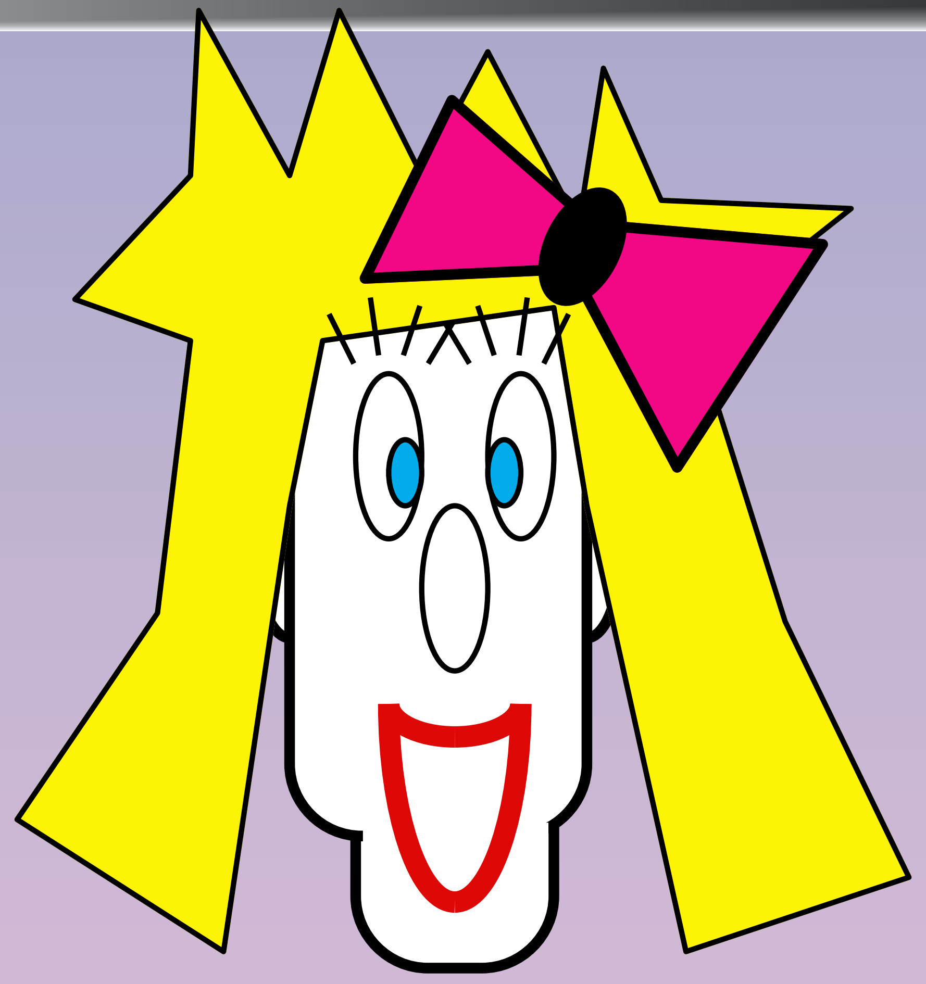
$\text{enc}_B(®0) \longrightarrow U0$

$®1 \longrightarrow U1$

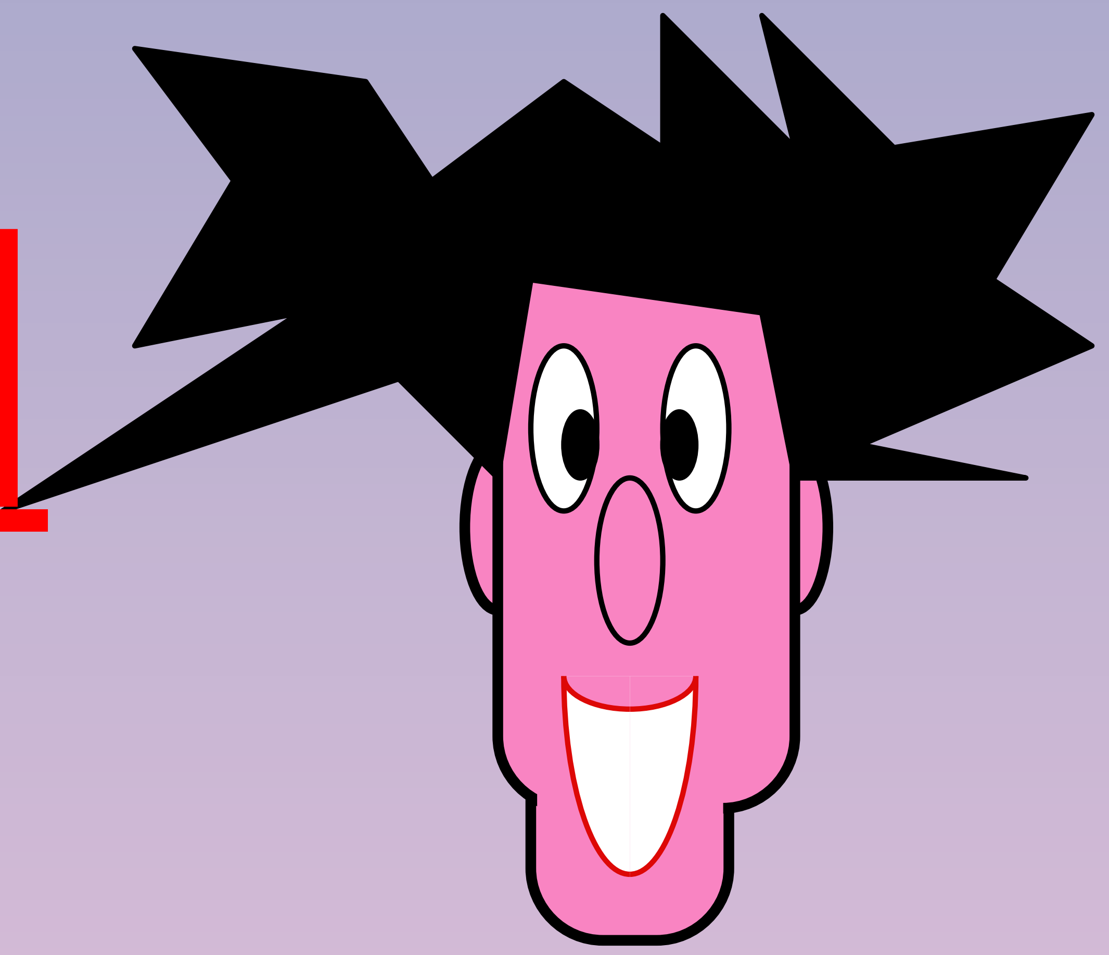
$Z_0 \longleftarrow \pi(\text{dec}_B(U0))(+)B0$

$Z_1 \longleftarrow \pi(\text{dec}_B(U1))(+)B1$

Use ZK proofs to make sure both party follow the protocol.



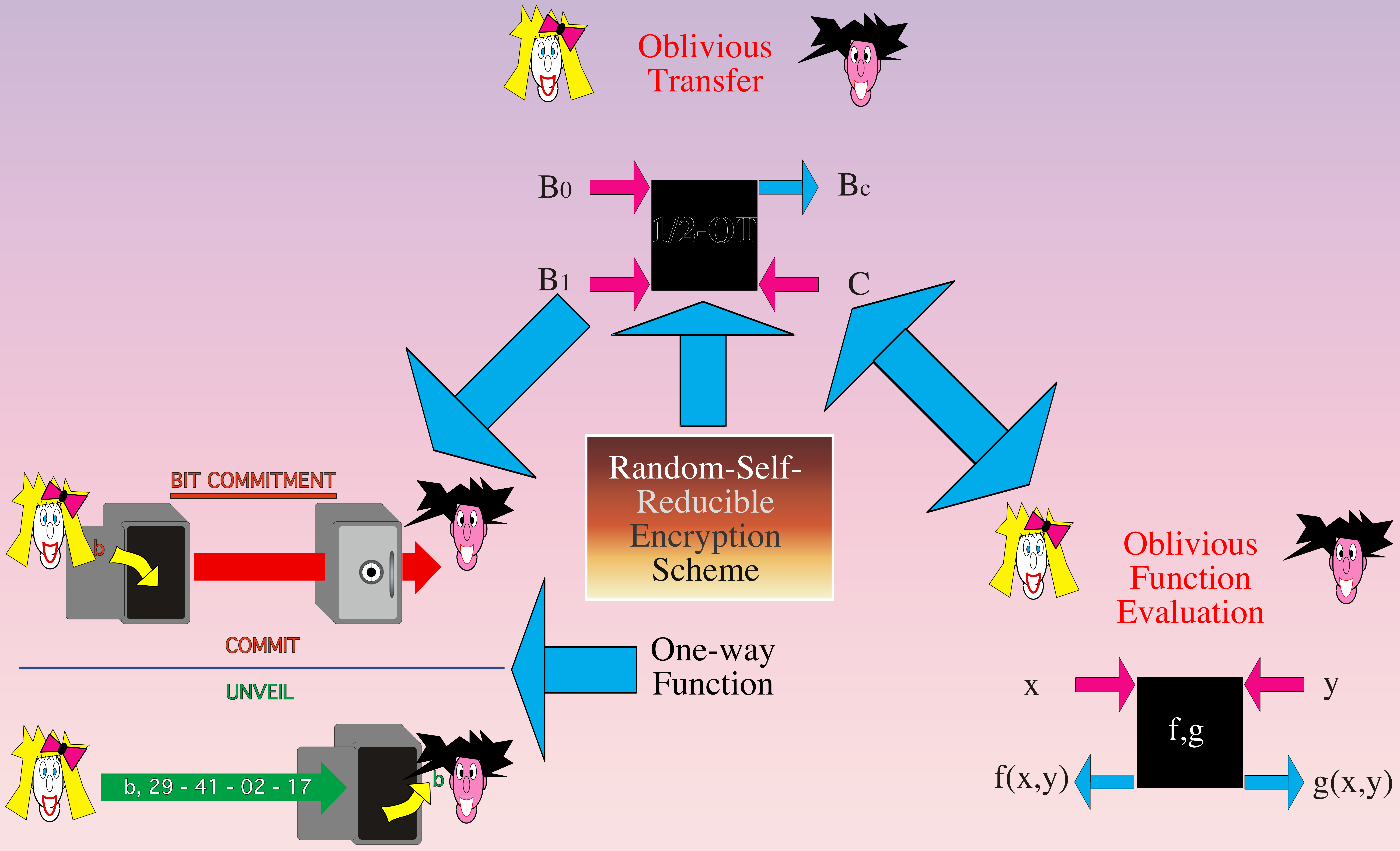
[Goldreich02]

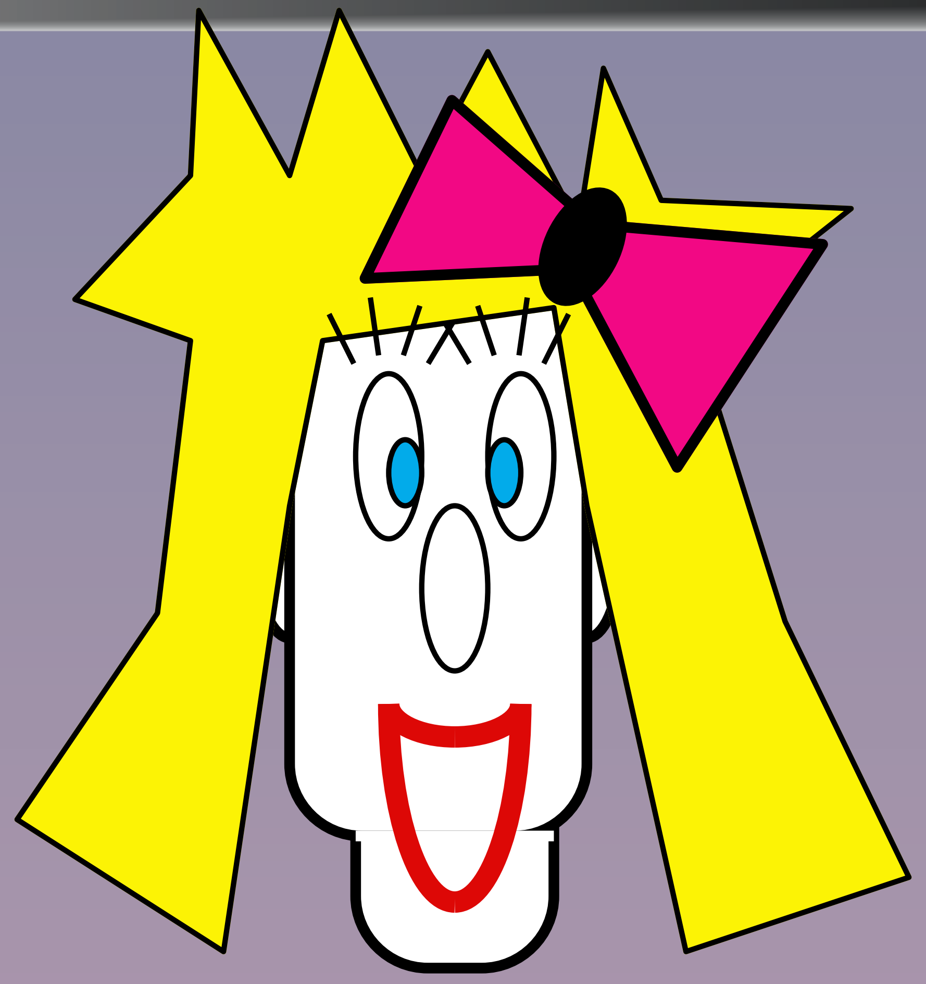


Definition (*Enhanced* TOWP)

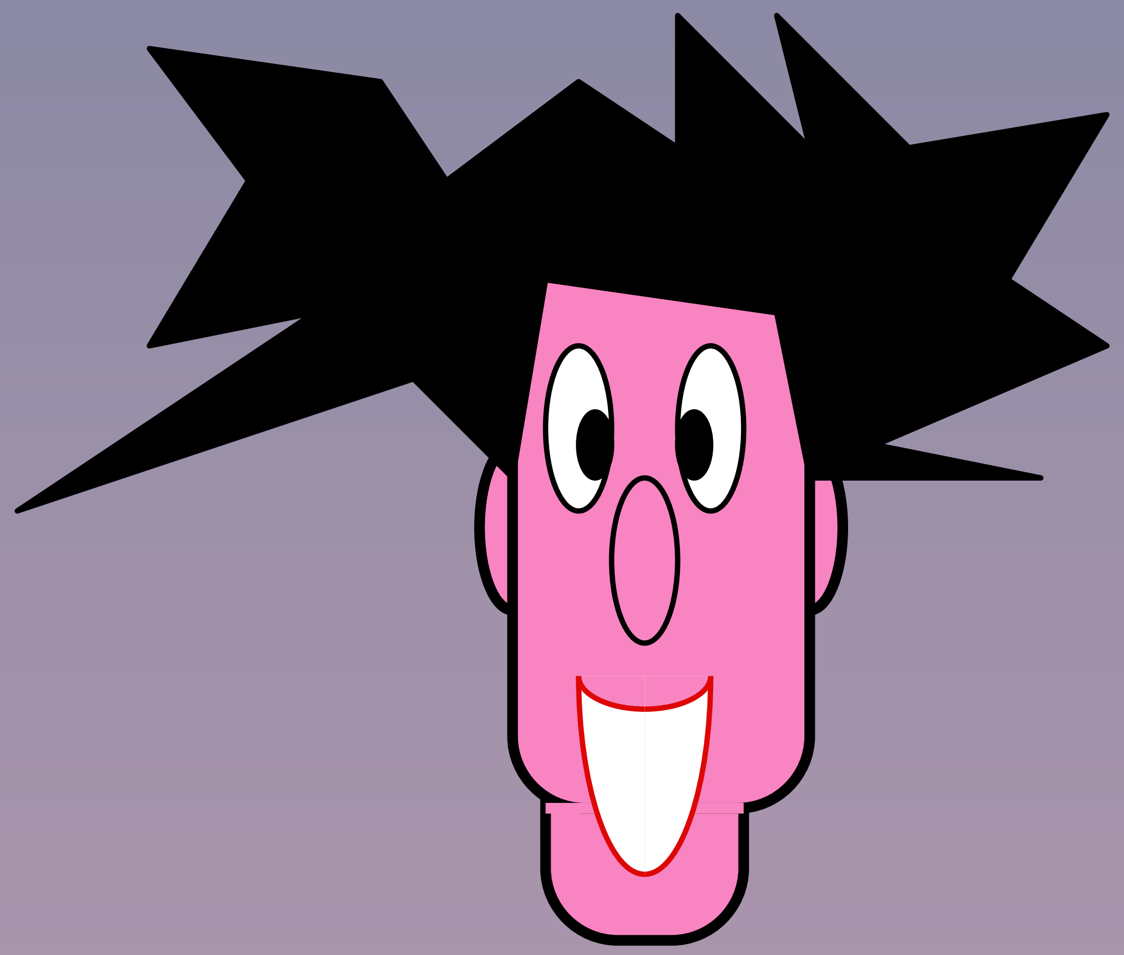
A TOWP **enc** is *enhanced* if there exists a PPT algorithm to select random elements from the image of **enc** without knowledge of the corresponding pre-image.

Classically





[GM84]
[BCR86]

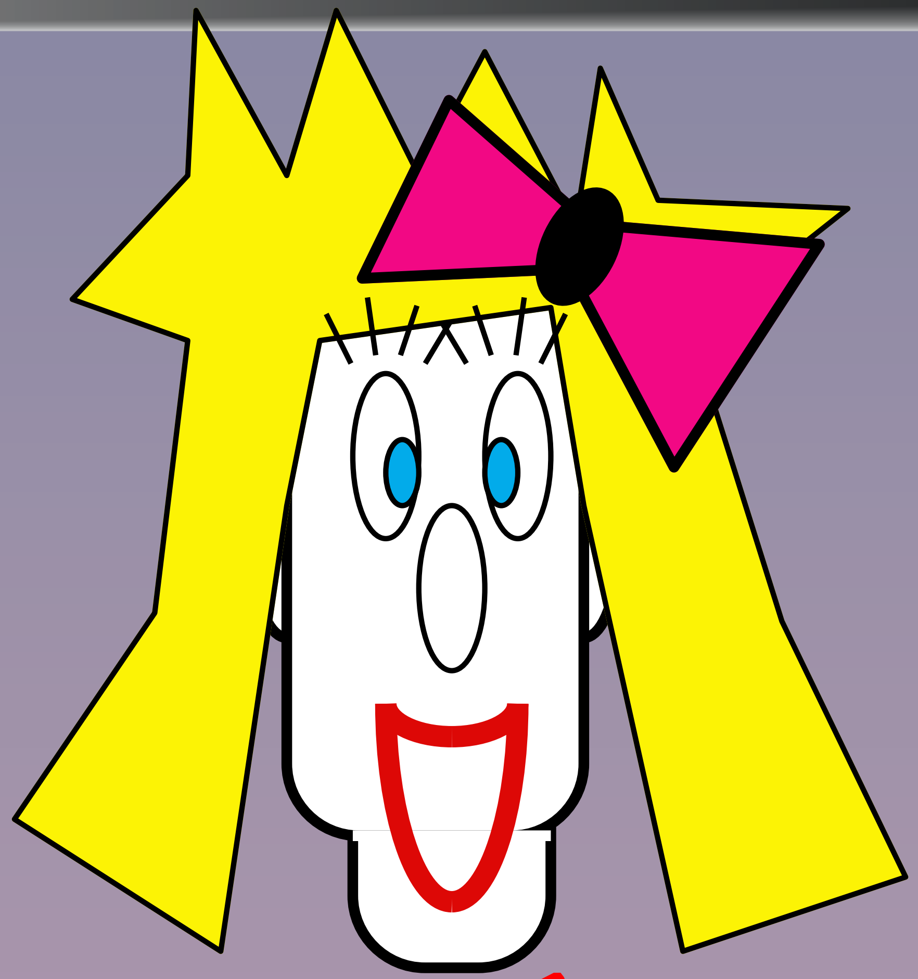


INGREDIENTS

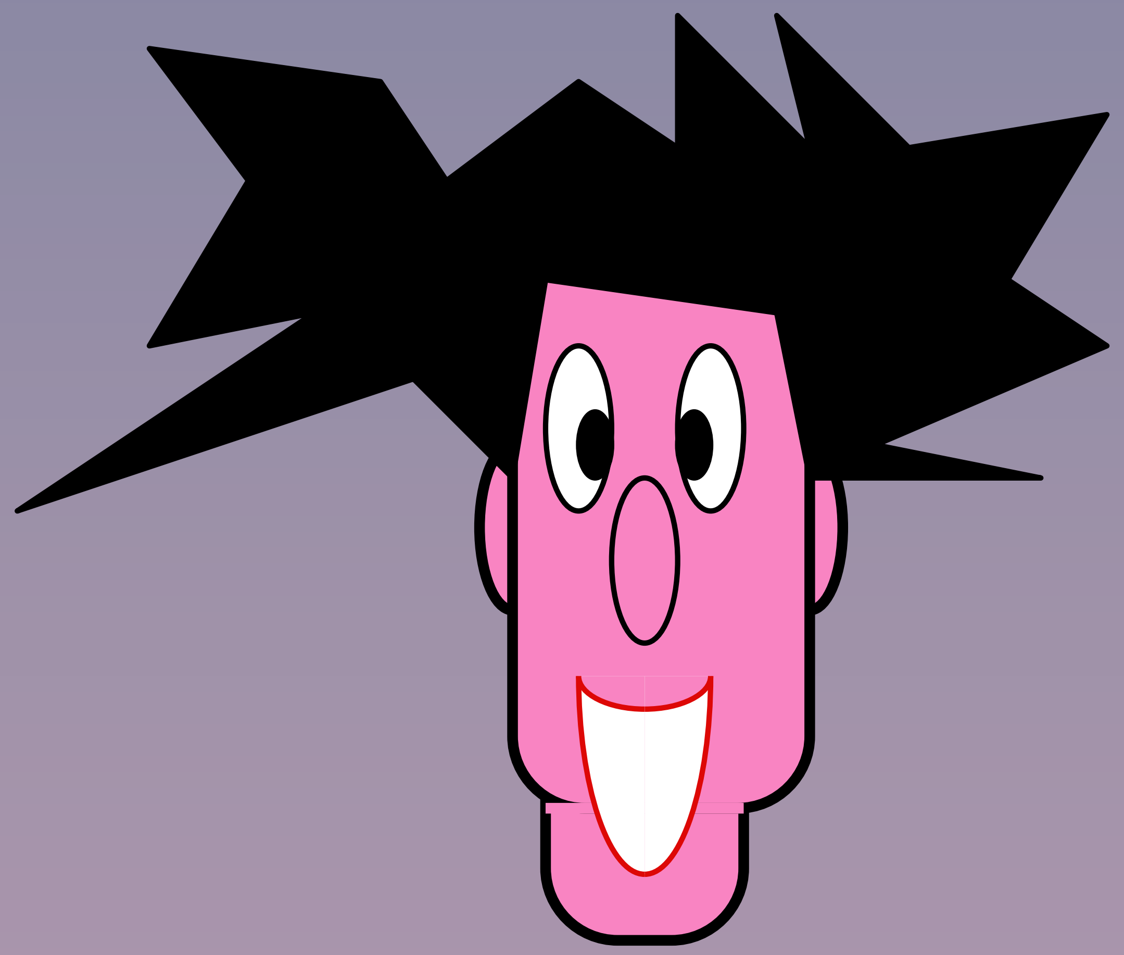
a public-key block cipher:

(enc_B, dec_B)

a public predicate: π



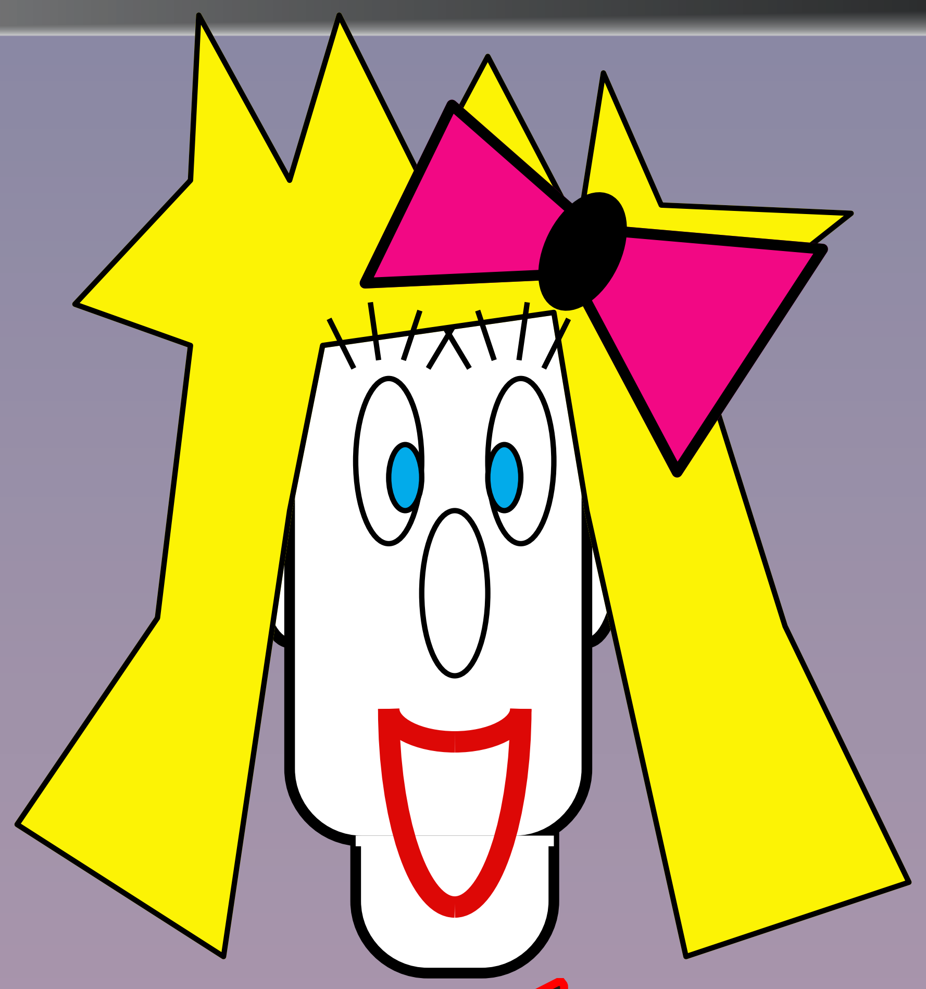
[GM84]
[BCR86]



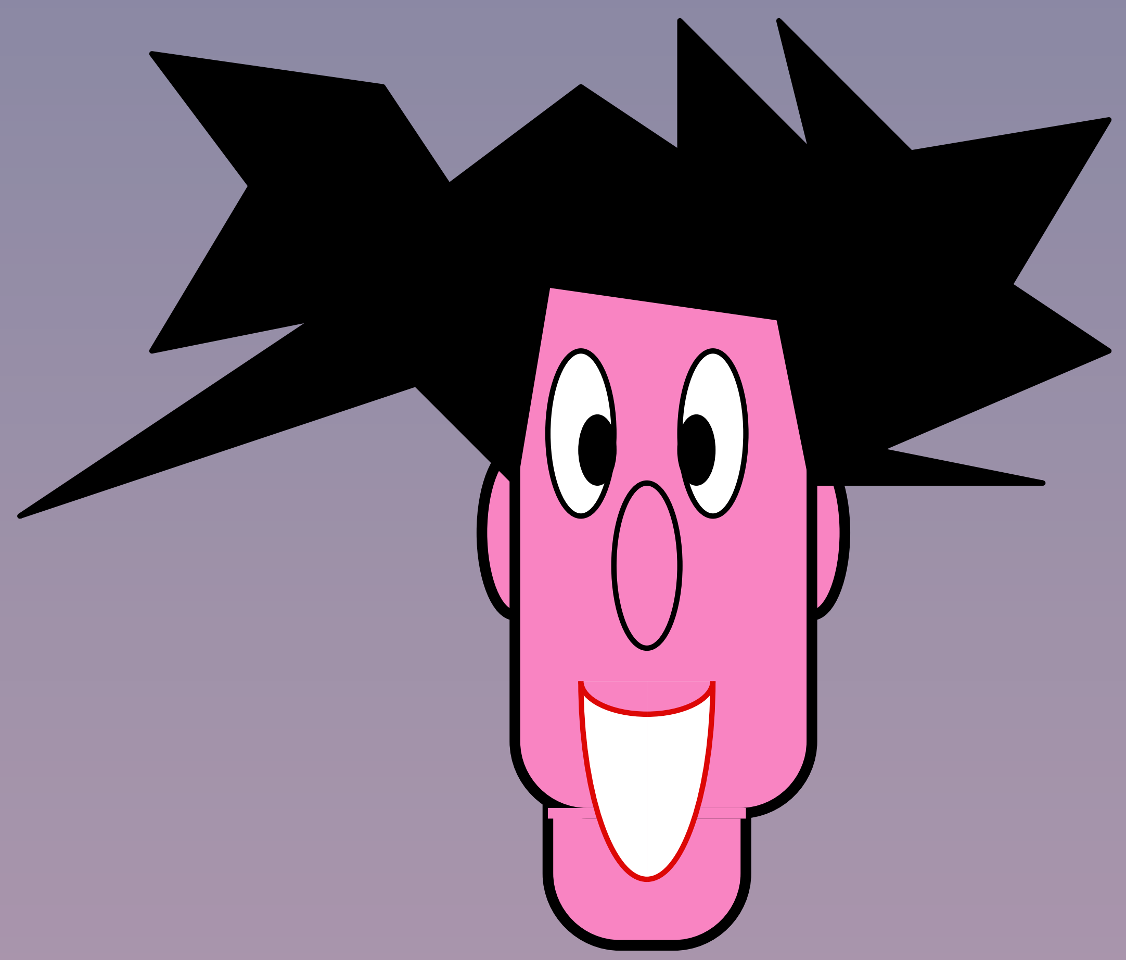
$m_0=\pi^{-1}(B_0)$

1/2-OT

$m_1=\pi^{-1}(B_1)$



[GM84]
[BCR86]

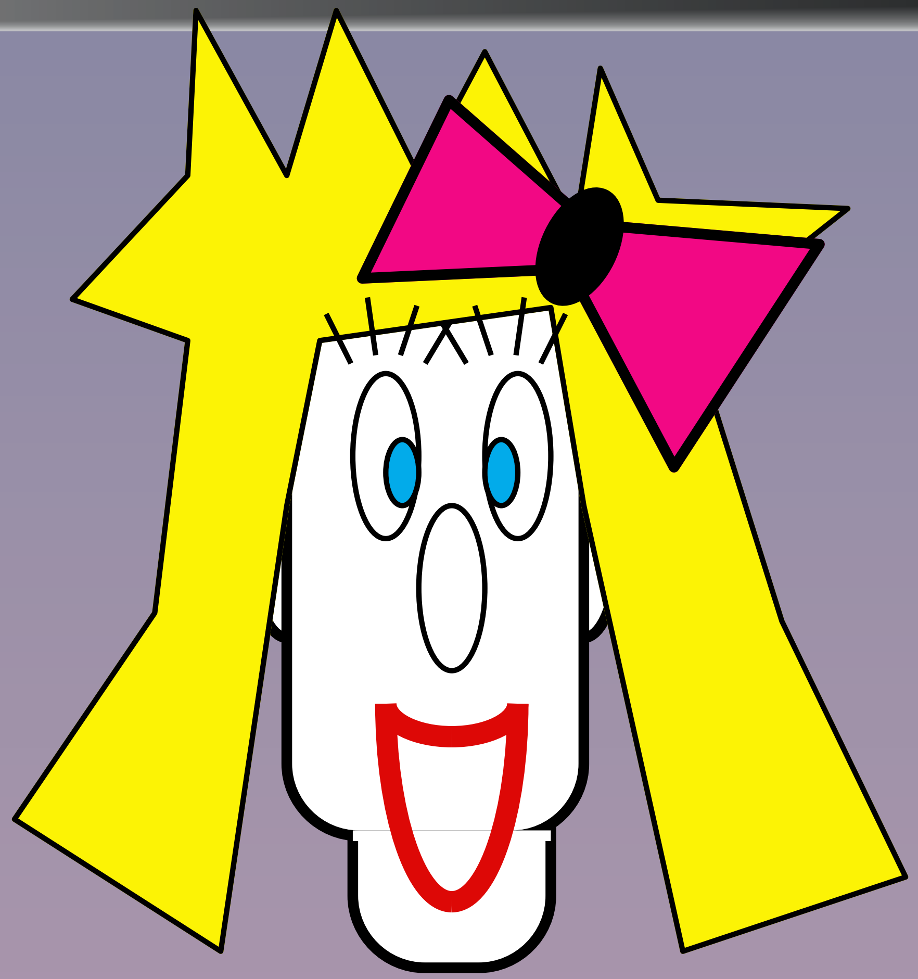


$m_0 = \pi^{-1}(B_0)$

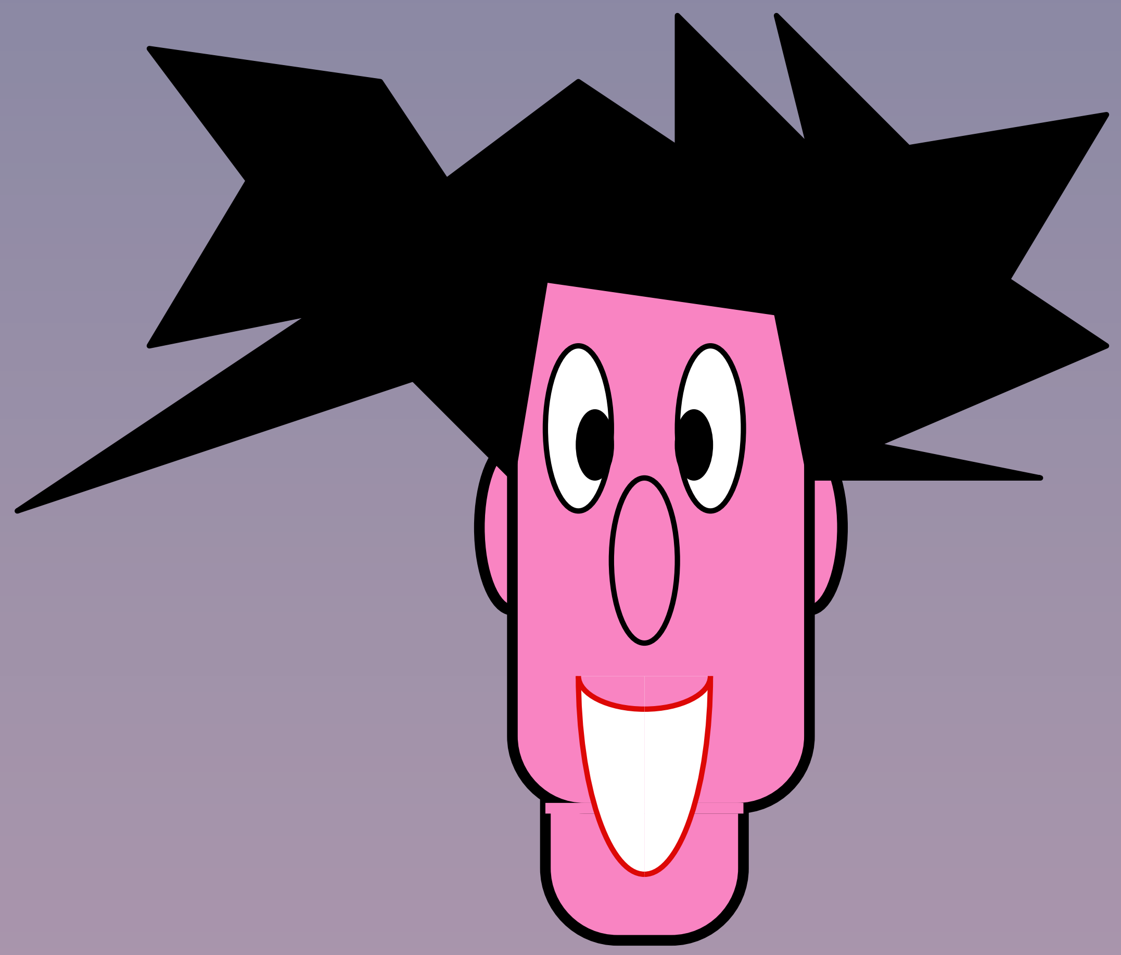
$\mathrm{enc}_A(m_0) \rightarrow U_0$

$m_1 = \pi^{-1}(B_1)$

$\mathrm{enc}_A(m_1) \rightarrow U_1$



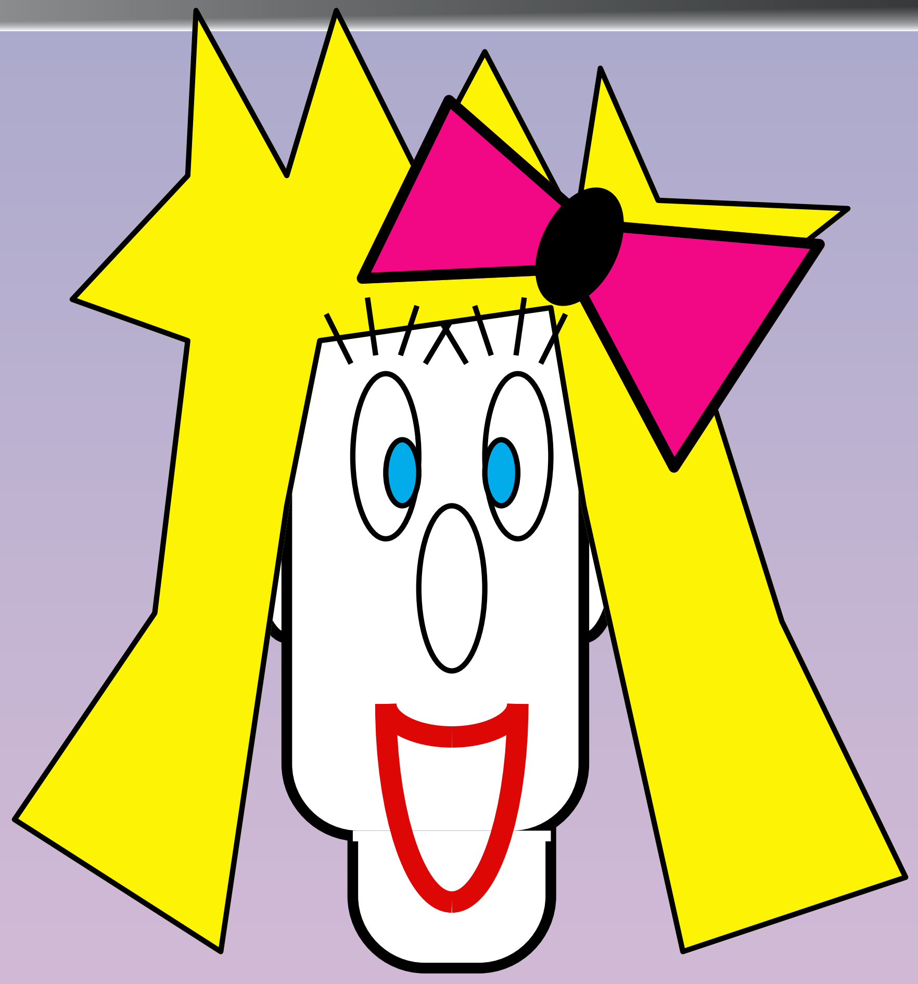
[GM84] [BCR86]



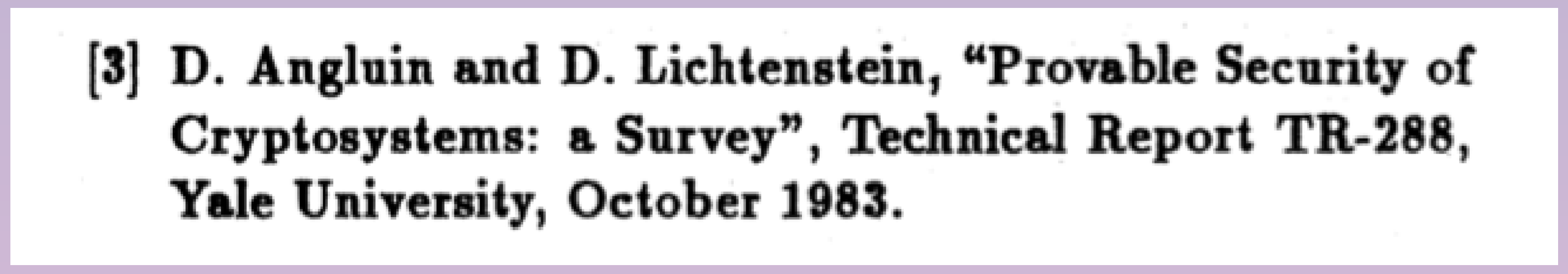
$enc_A(m_0) \rightarrow U_0$

$enc_A(m_1) \rightarrow U_1$

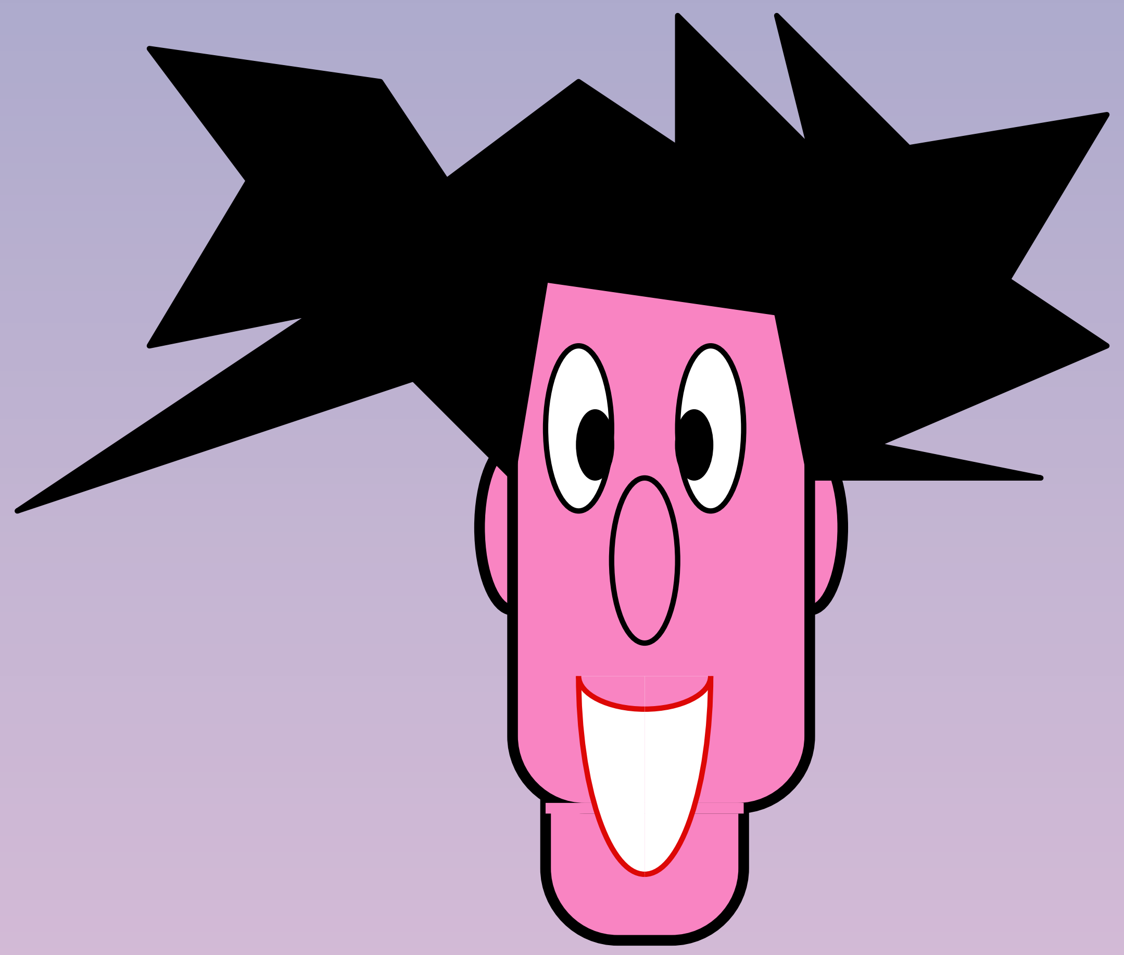
$RSR_A(®,U_c)$



[AL83]



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



Definition (*RSR* Encryption Scheme)

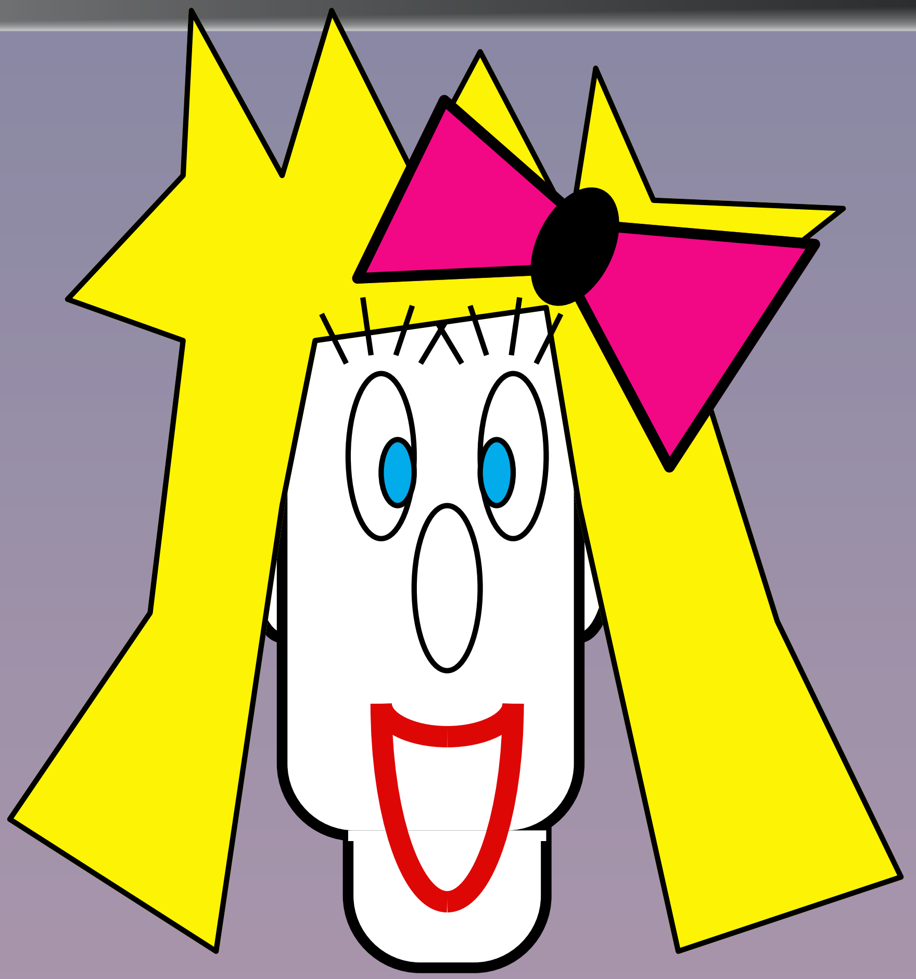
A public-key encryption scheme (enc,dec) is *Random-Self-Reducible* if there exists a pair of PPT algorithms $(\mathrm{RSR},\mathrm{RSR}^{-1})$ such that for all ®,m,

$$\mathrm{RSR}^{-1}(®,\mathrm{dec}(\mathrm{RSR}(®,\mathrm{enc}(m))) = m$$

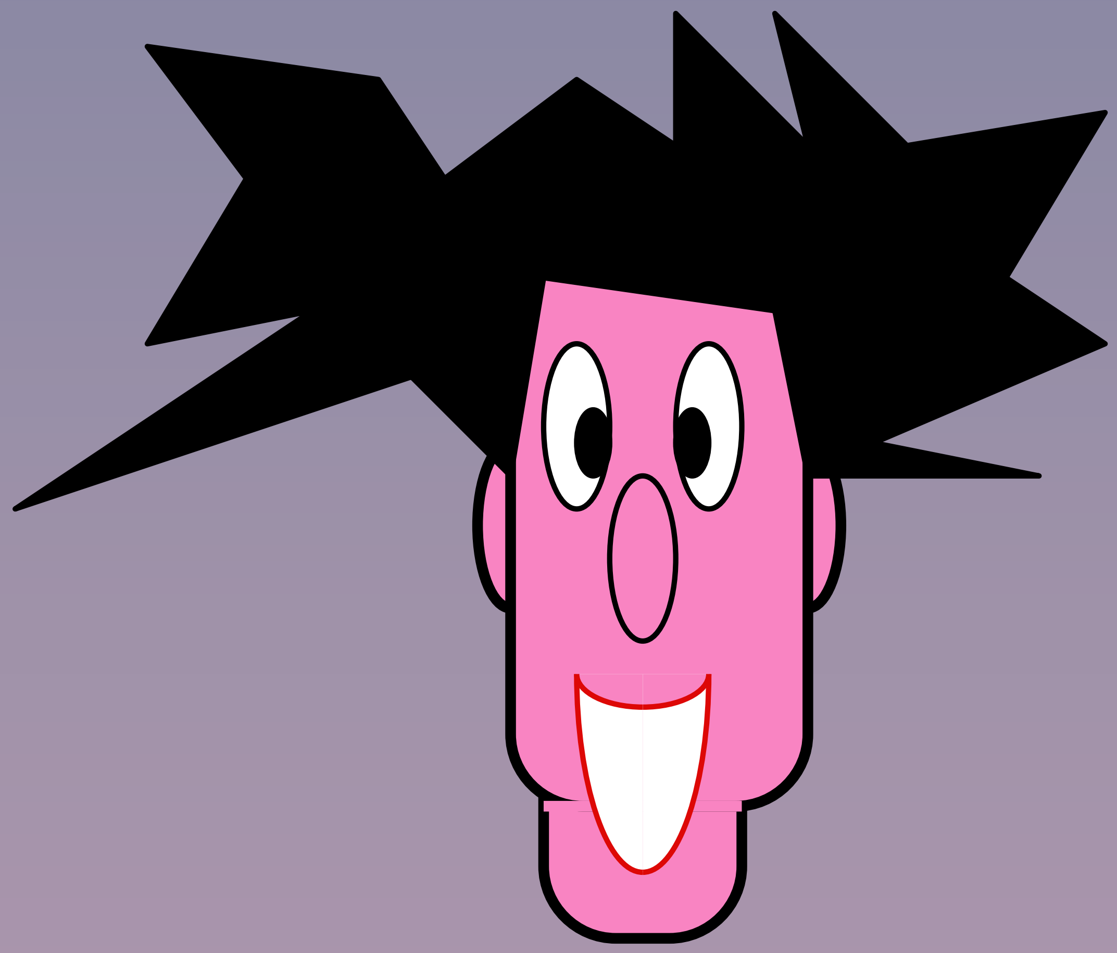
and

RSR(®,enc(m)) is a uniform ciphertext

when ® is uniform.



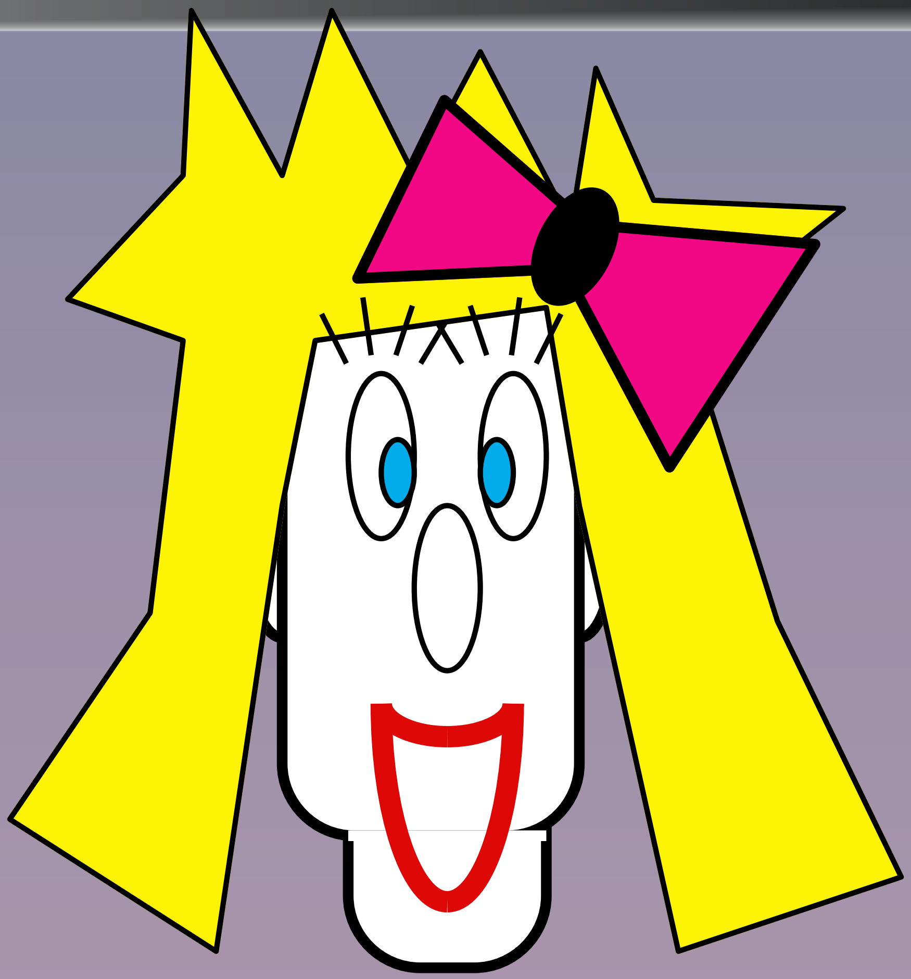
[GM84] [BCR86]



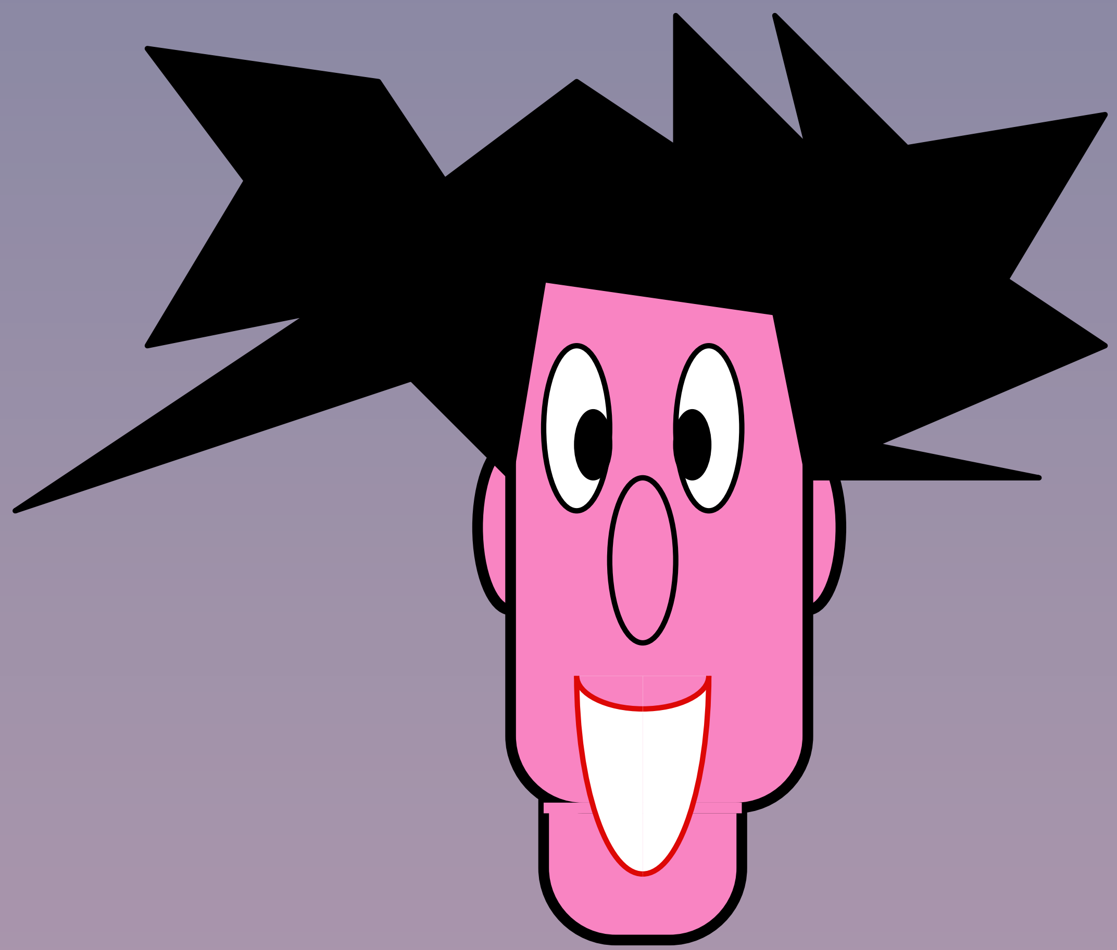
$$\mathrm{enc}_A(m_0) \longrightarrow U_0$$

$$\mathrm{enc}_A(m_1) \longrightarrow U_1$$

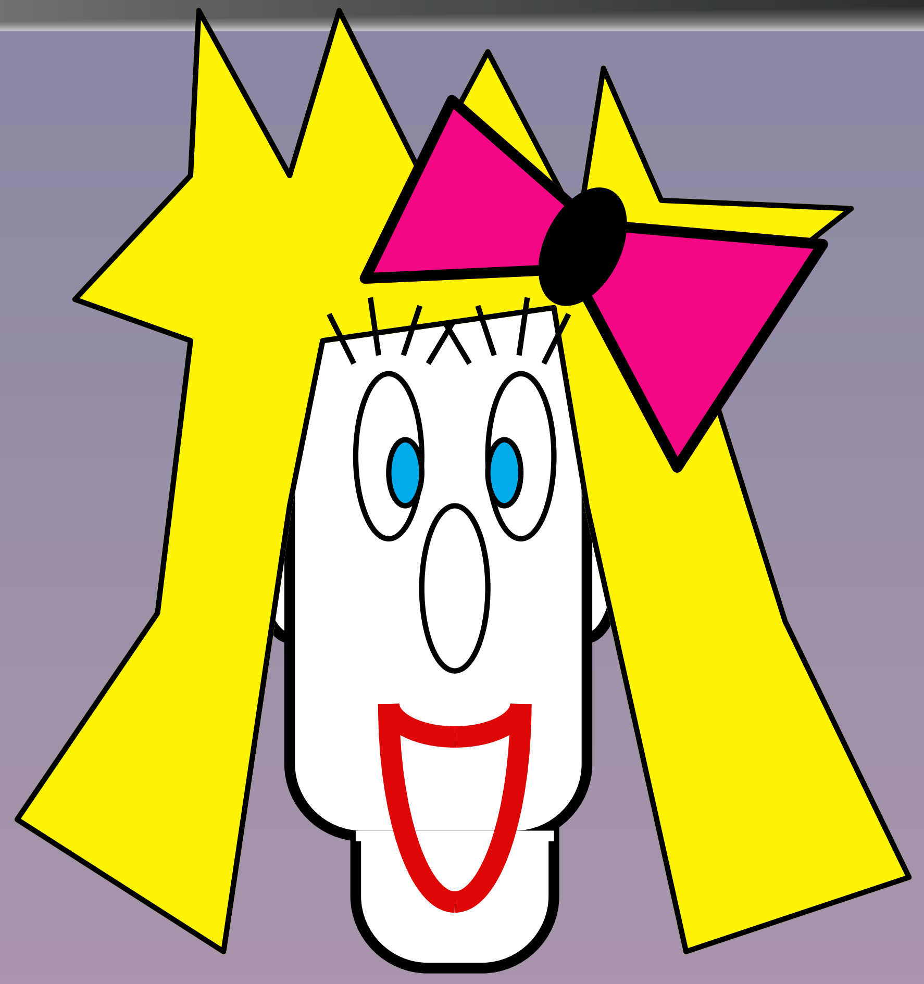
$$z \longleftarrow \mathrm{RSR}_A(®, U_c)$$



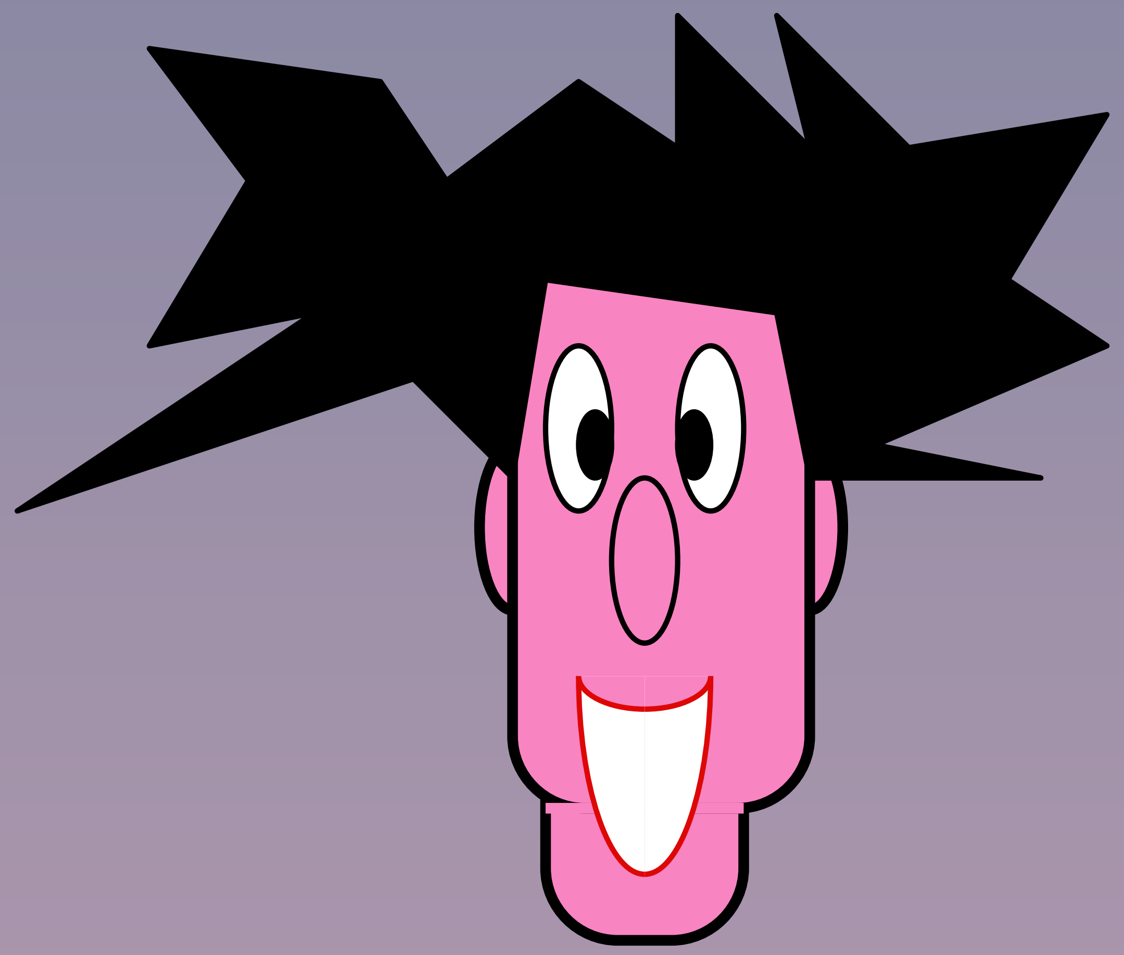
[GM84] [BCR86]



$$\begin{tikzcd} enc_A(m_0) \arrow[r] & U_0 \\ enc_A(m_1) \arrow[r] & U_1 \\ z & enc_A(®*m_c) = enc_A(®)\bullet U_c \arrow[l] \end{tikzcd}$$



[GM84]
[BCR86]

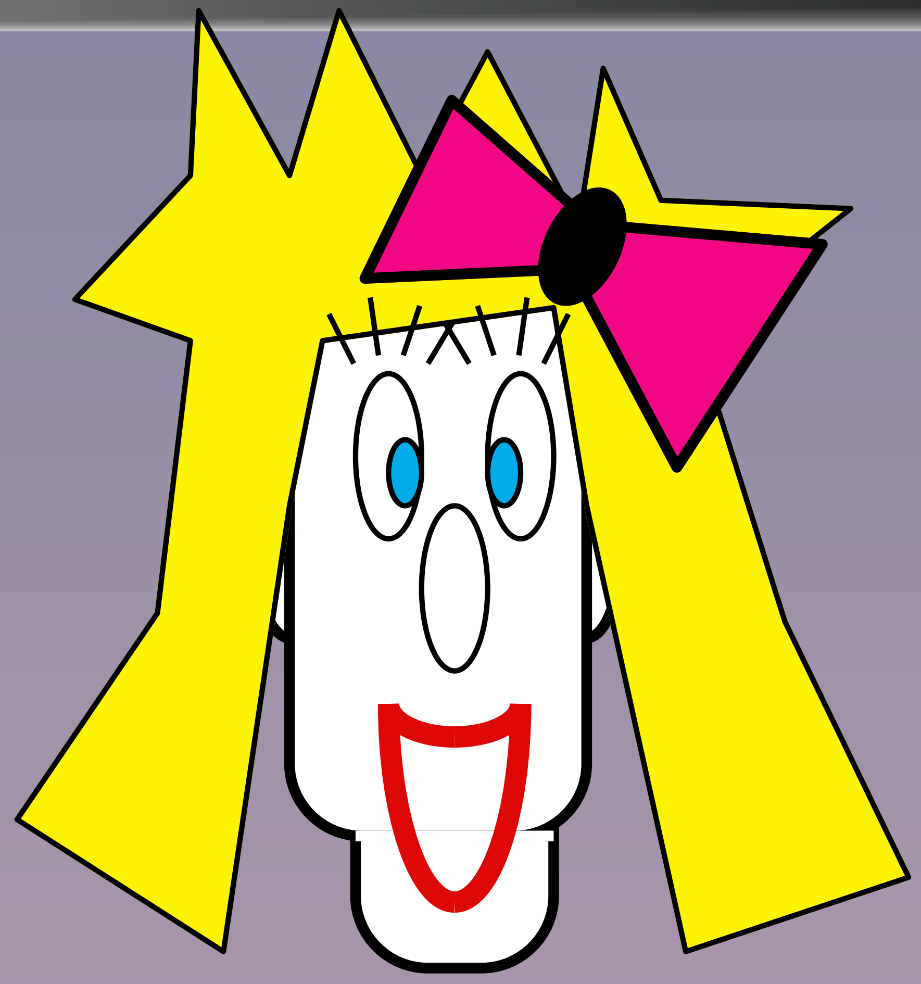


$\mathrm{enc}_A(m_0) \longrightarrow$ U0

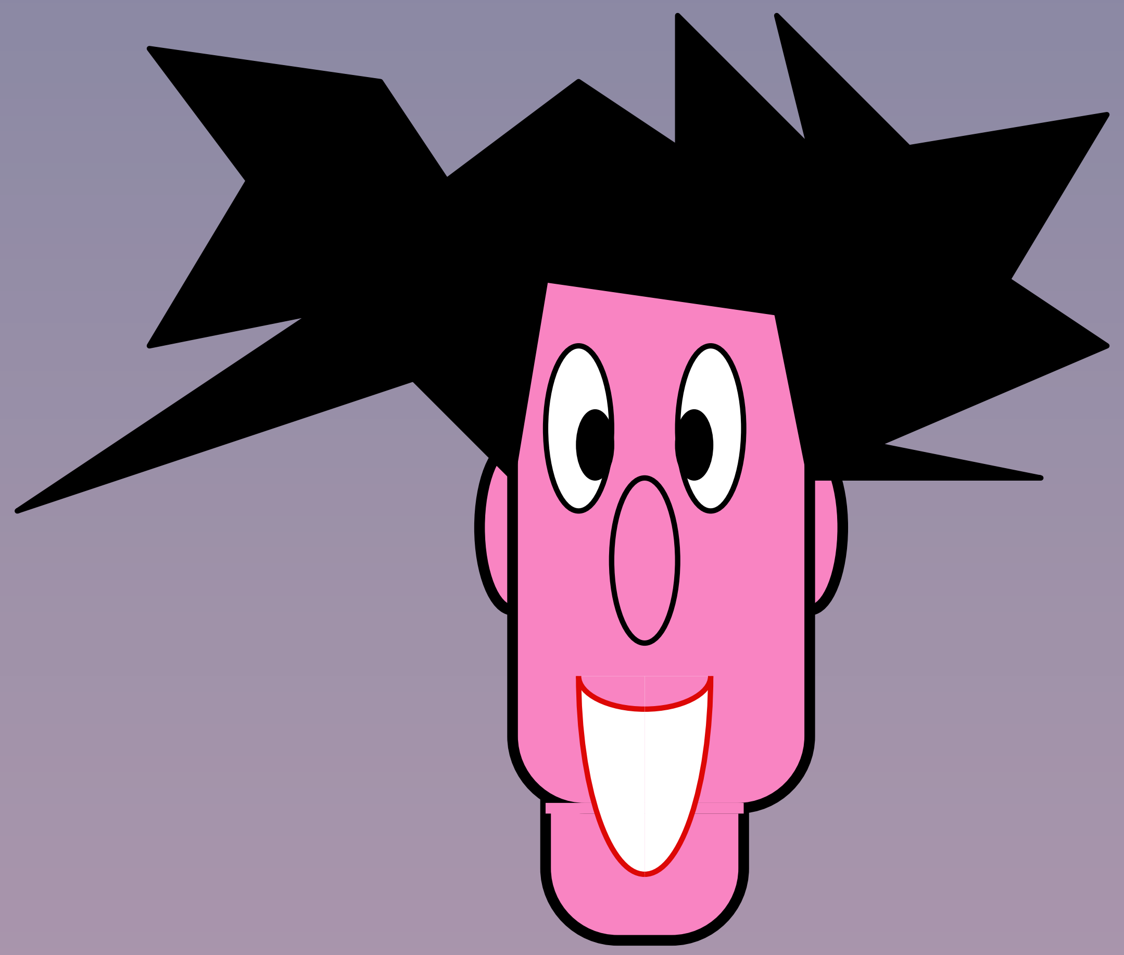
$\mathrm{enc}_A(m_1) \longrightarrow$ U1

$z \longleftarrow$ RSR_A(®,Uc)

$\mathrm{dec}_A(z) \longrightarrow$



[GM84] [BCR86]



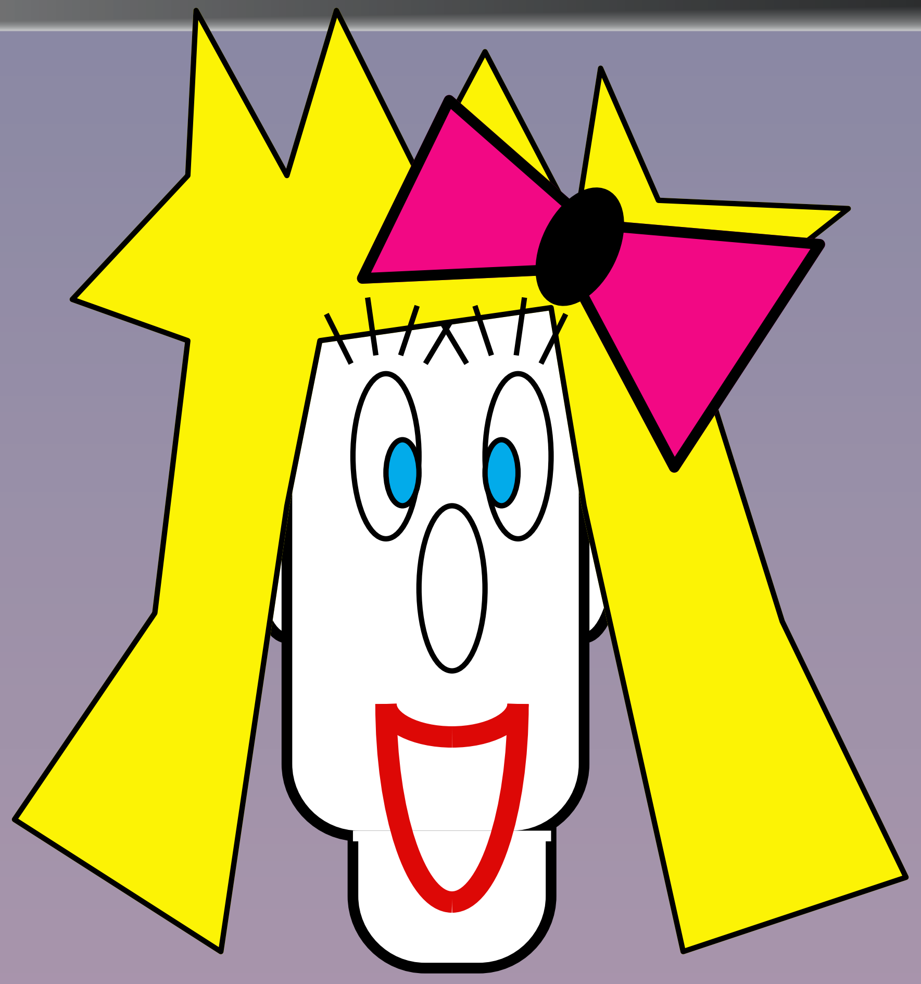
$\mathrm{enc}_A(m_0) \longrightarrow U_0$

$\mathrm{enc}_A(m_1) \longrightarrow U_1$

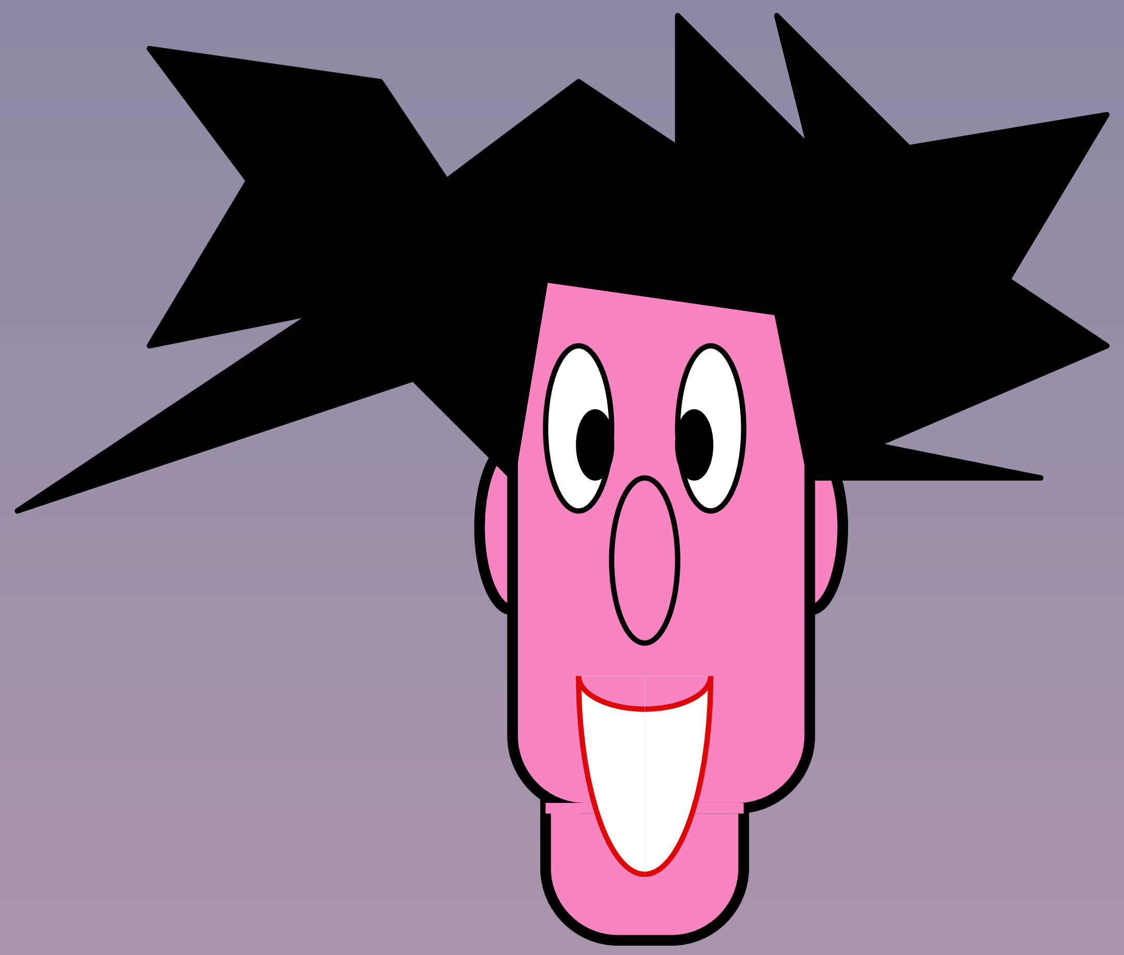
$z \longleftarrow RSR_A(®,U_c)$

$\mathrm{dec}_A(z) \longrightarrow y$

$B_c=\pi(RSR_A^{-1}(®,y))$



[GM84] [BCR86]



1/2-OT

$$\mathrm{enc}_A(m_0) \longrightarrow U_0$$

$$\mathrm{enc}_A(m_1) \longrightarrow U_1$$

$$z \longleftarrow \mathrm{enc}_A(®)\bullet U_c$$

$$\mathrm{dec}_A(z) \longrightarrow ®*m_c$$

$$B_c = \pi(®^{-1}*®*m_c)$$

OT Implementations

RSA

(* = x mod n, • = x mod n)

El-Gammal

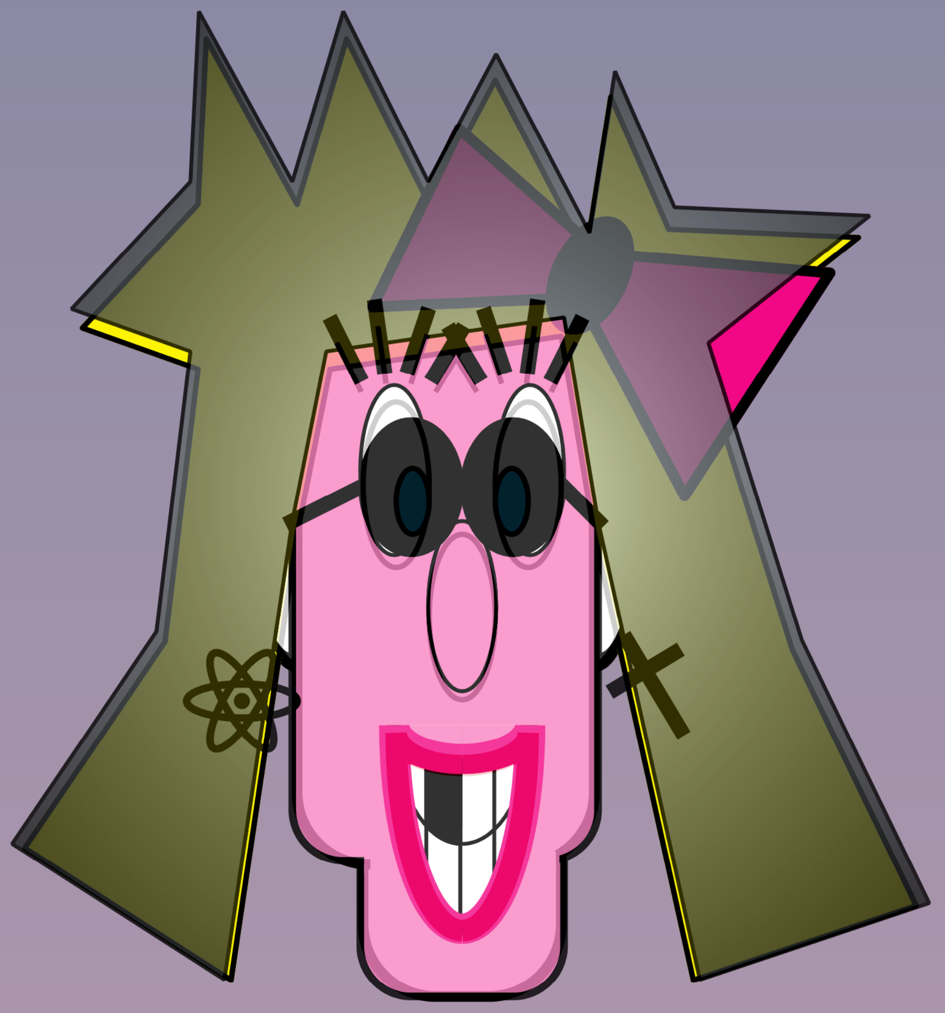
(* = x mod p, • = x mod p)

Goldwasser-Micali

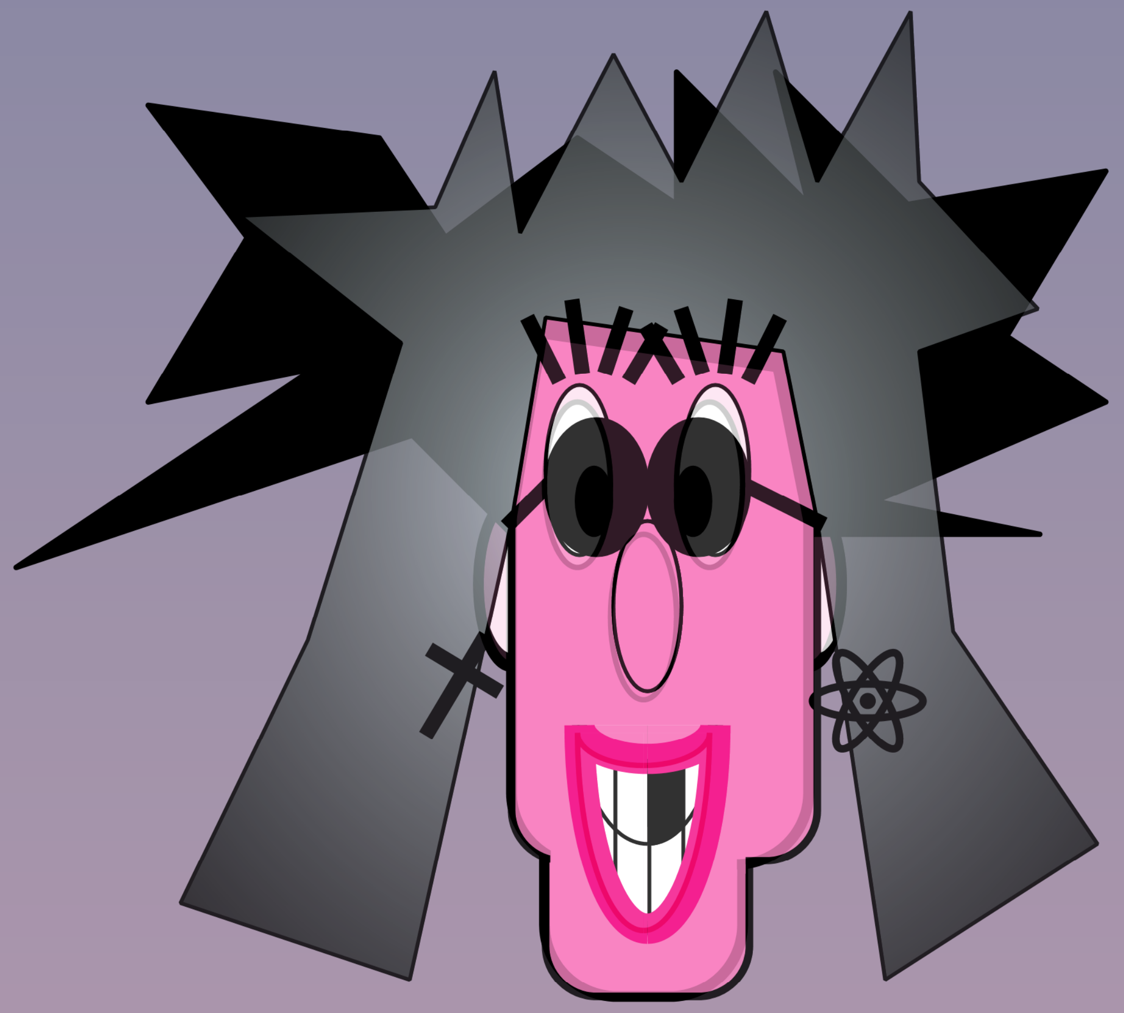
(* = (+), • = x mod n)

Paillier

(* = + mod N, • = x mod N^2)



[BCR86]



1/2-OT

$$\mathrm{enc}_A(m_0) \longrightarrow U_0$$

$$\mathrm{enc}_A(m_1) \longrightarrow U_1$$

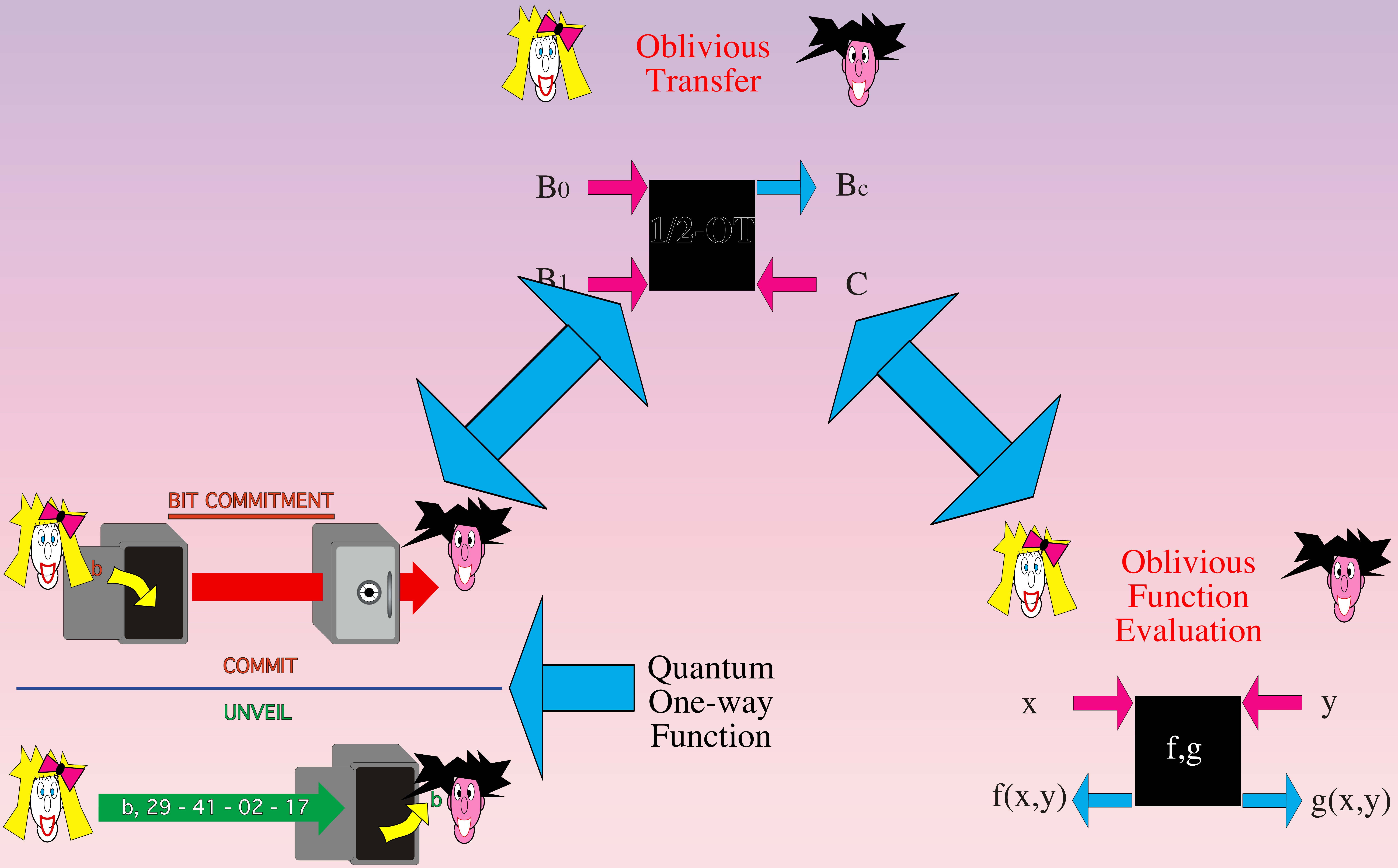
$$z \longleftarrow RSR_A(\text{®}, U_c)$$

$$\mathrm{dec}_A(z) \longrightarrow$$

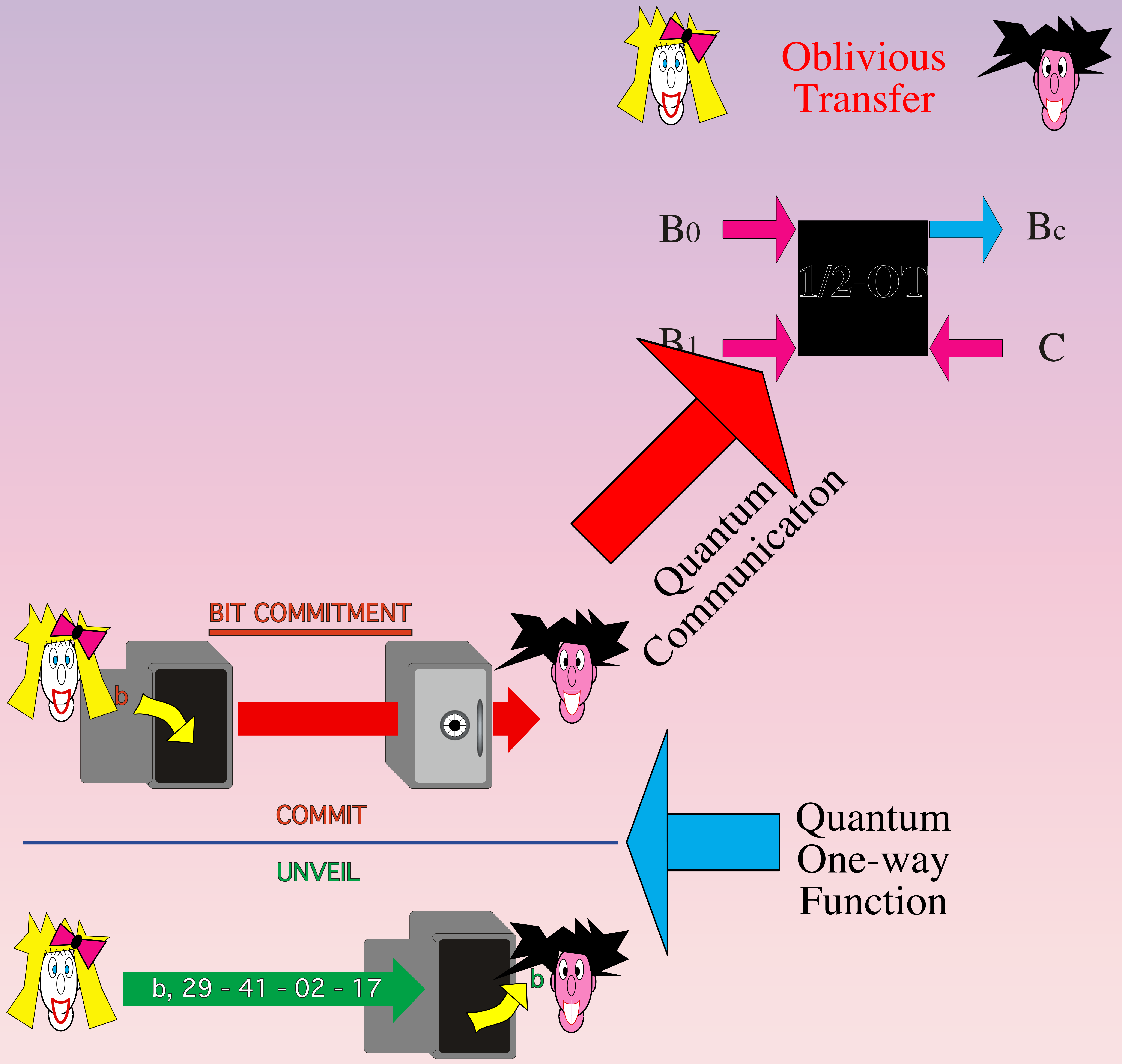
Use ZK proofs to make sure both party follow the protocol.

(2.5) Quantum Secure OT Implementations

Quantumly

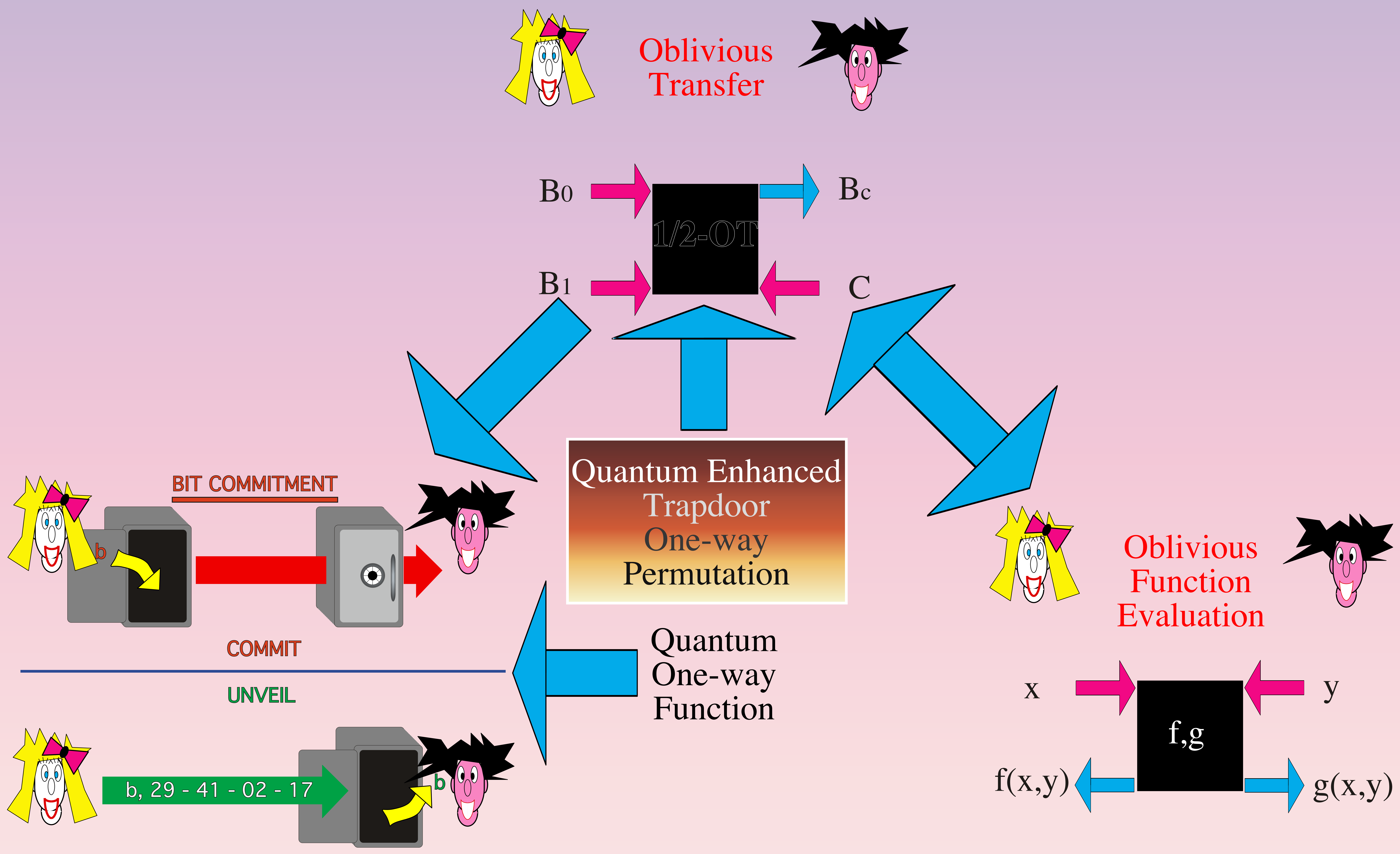

Oblivious Transfer
B0
Bc
1/2-OT
B1
C
BIT COMMITMENT
b
COMMIT
UNVEIL
b, 29 - 41 - 02 - 17
b
Quantum One-way Function
Oblivious Function Evaluation
x
y
f,g
f(x,y)
g(x,y)

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 Enhanced Trapdoor One-way Permutation

$$\{\mathrm{QETOP}\} = \emptyset ?$$

Quantum Enhanced Trapdoor One-way Permutation

RSA

Discrete Logarithm

Elliptic Curves

Factoring

Quantum *Enhanced Trapdoor One-way Permutation*

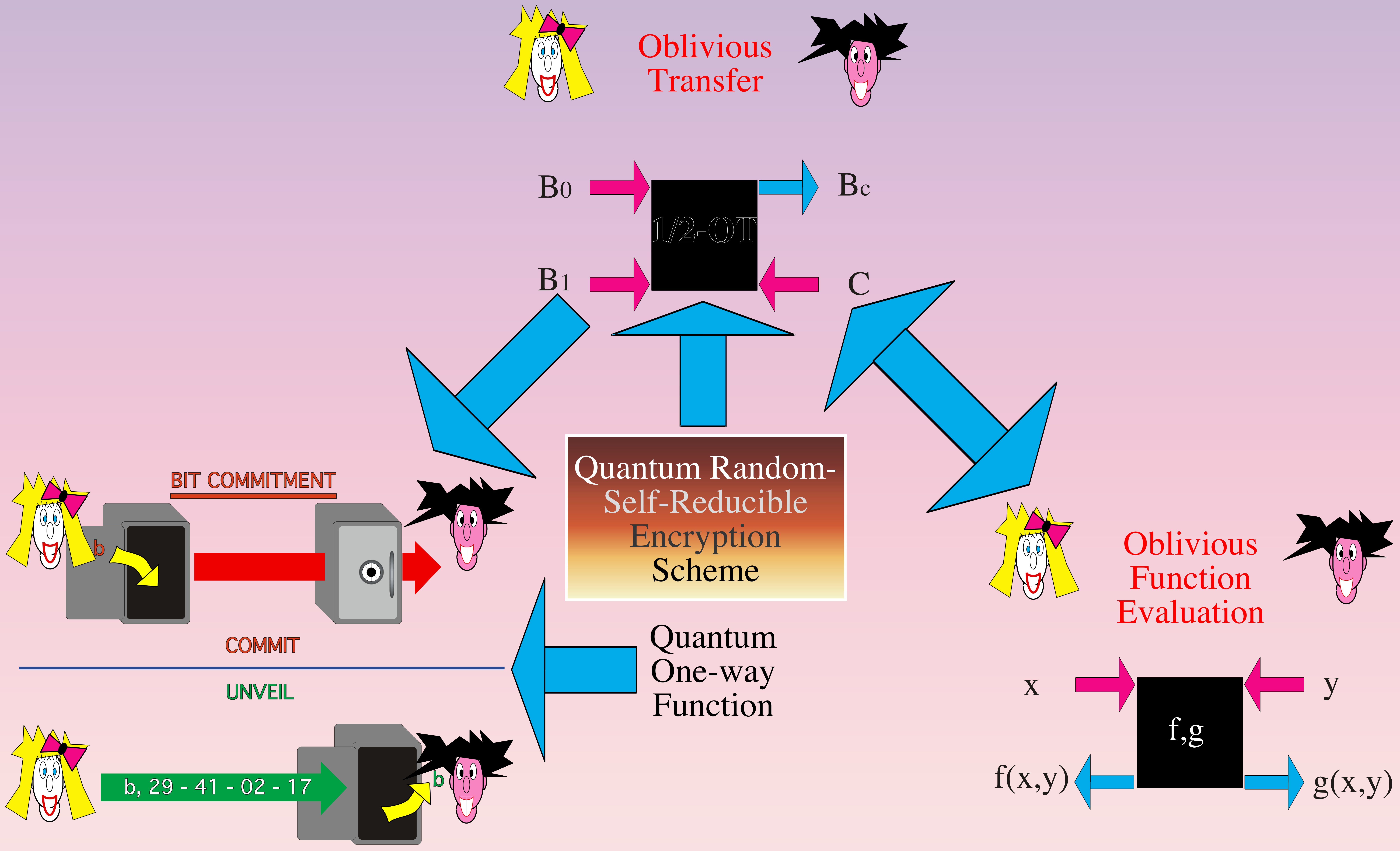
Lattices

McEliece

Integer GCD

LWE

Quantumly Secure Classically Implemented



Quantum Random-Self-Reducible Encryption Scheme

$$\{\text{QRSRES}\} = \emptyset ?$$

Quantum Random-Self-Reducible Encryption Schemes

RSA

El-Gammal

Elliptic Curves

Goldwasser Micali

Paillier

Quantum *Random-Self-Reducible Encryption Schemes*

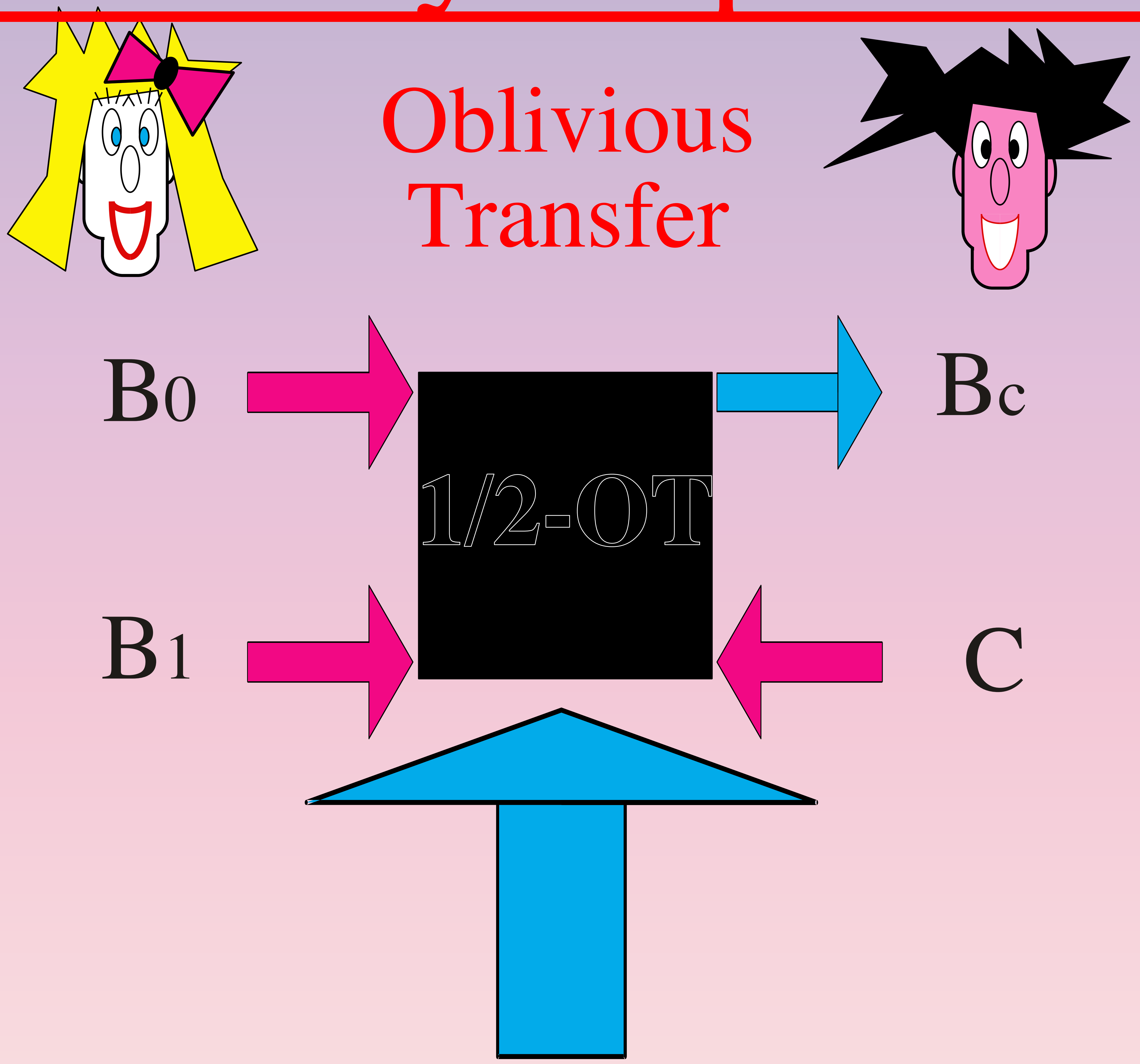
Lattices

McEliece

Integer GCD

LWE

Quantumly Secure Classically Implemented



Quantum Weakly Random-Self-Reducible Encryption Scheme

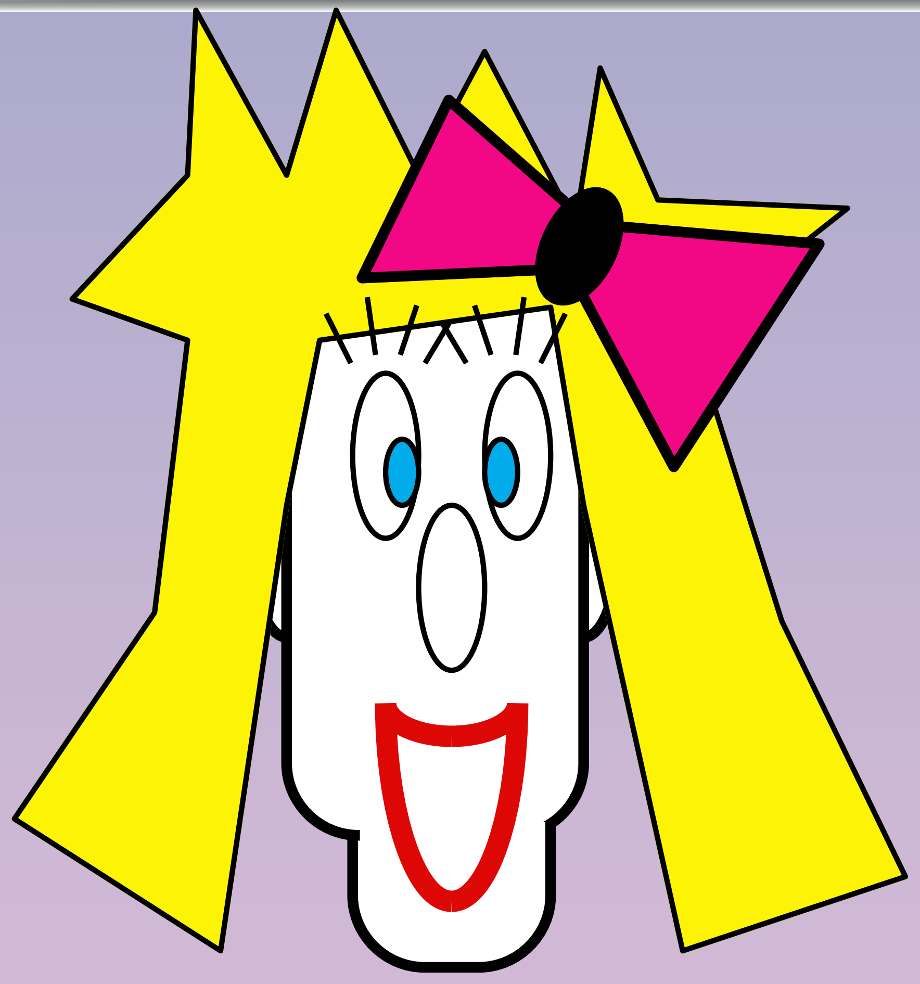
Quantum Weakly Random-Self-Reducible Encryption Scheme

~~McEliece~~

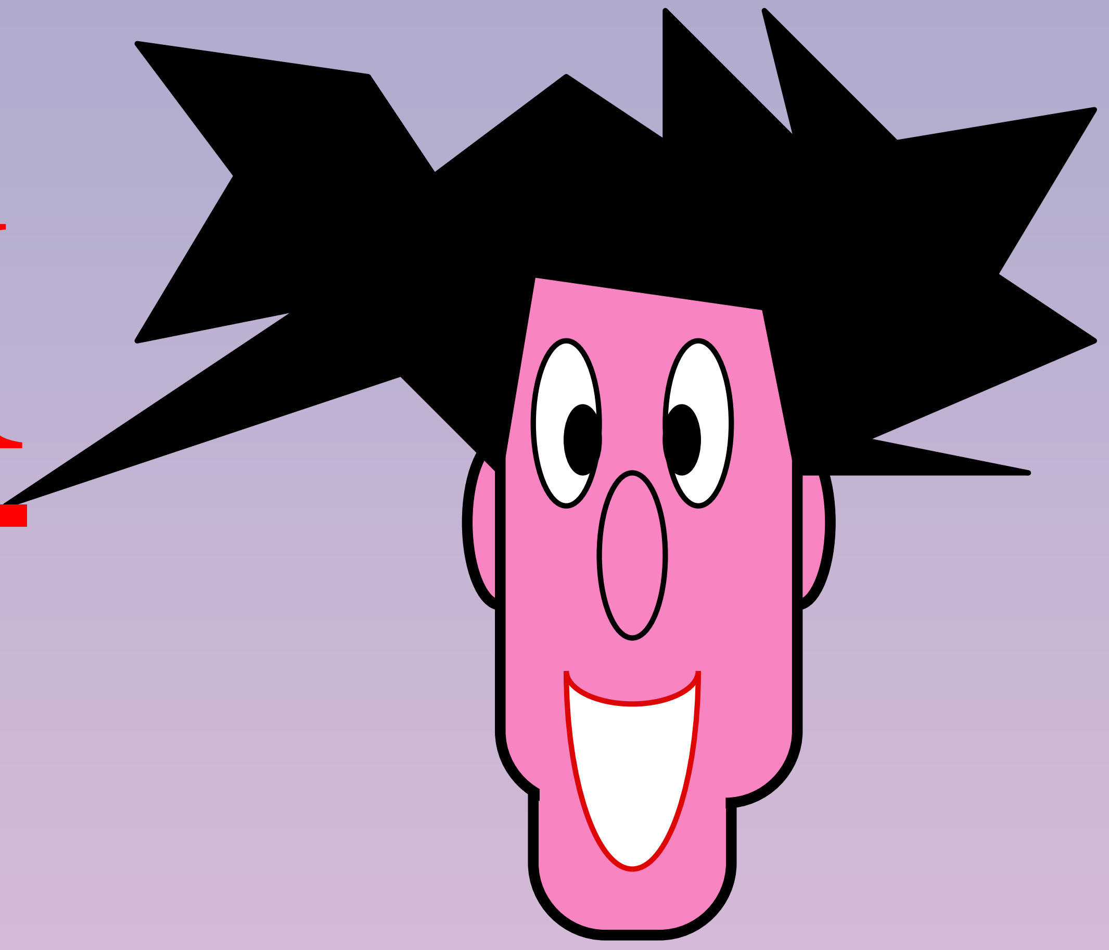
Lattices

Integer GCD

LWE



THIS WORK



Definition (*Weakly* RSR Encryption Scheme)

A public-key encryption scheme (enc,dec) is *Weakly* Random-Self-Reducible if there exists a pair of PPT algorithms (RsR, RsR^{-1}) such that for all ®,m,

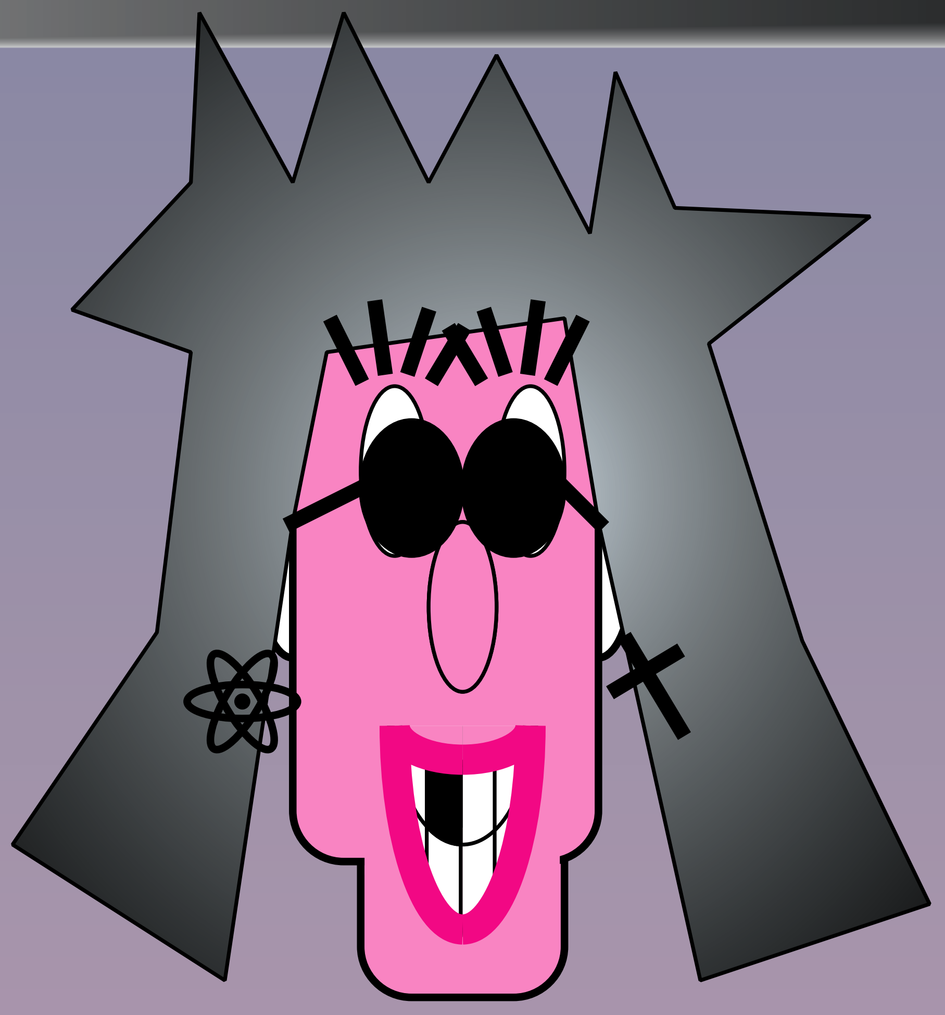
$$RsR^{-1}(®,dec(RsR(®,enc(m))) = m$$

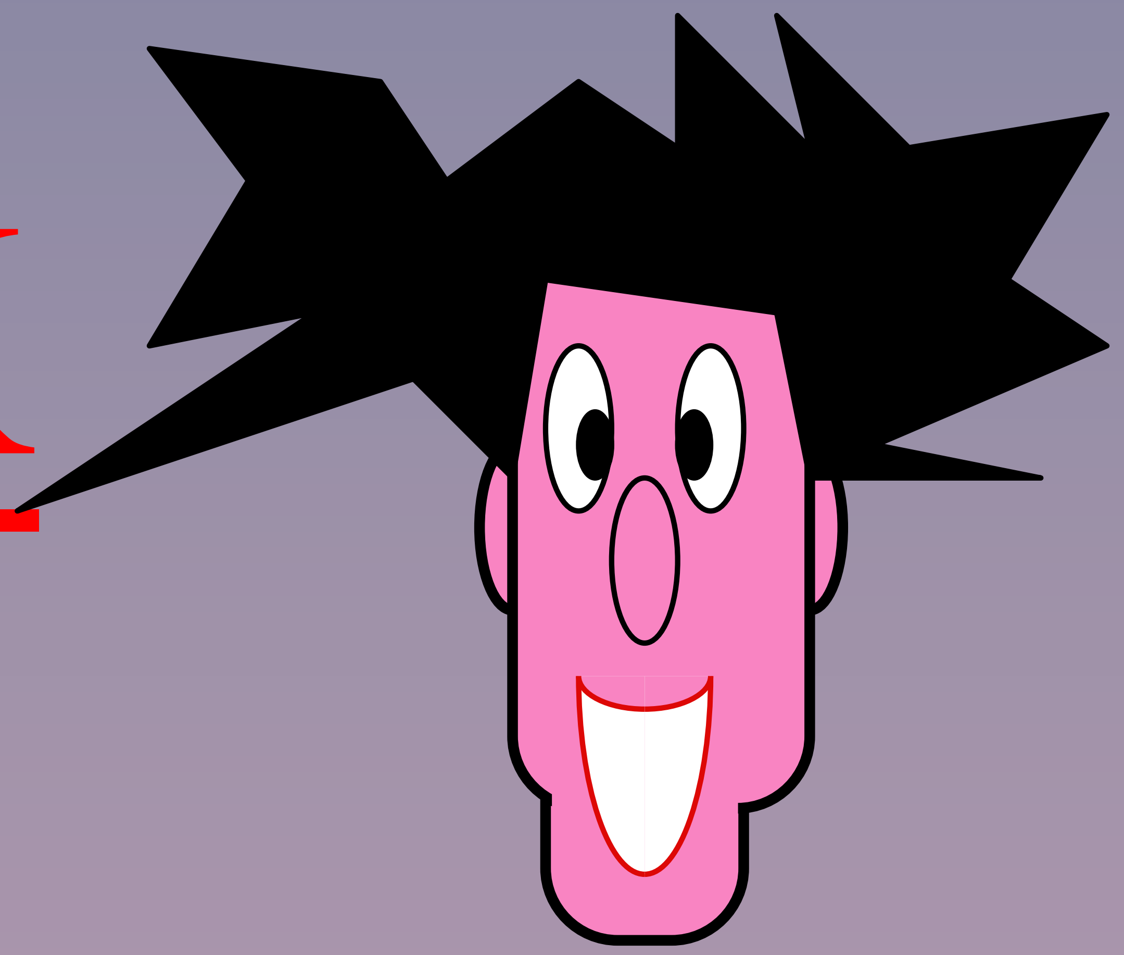
and there exists a PPT ditribution on ® s.t. for all m,m’

$$RsR(®,enc(m)) \sim RsR(®,enc(m’)).$$

(3) Implementation from QwRsRES

THIS WORK



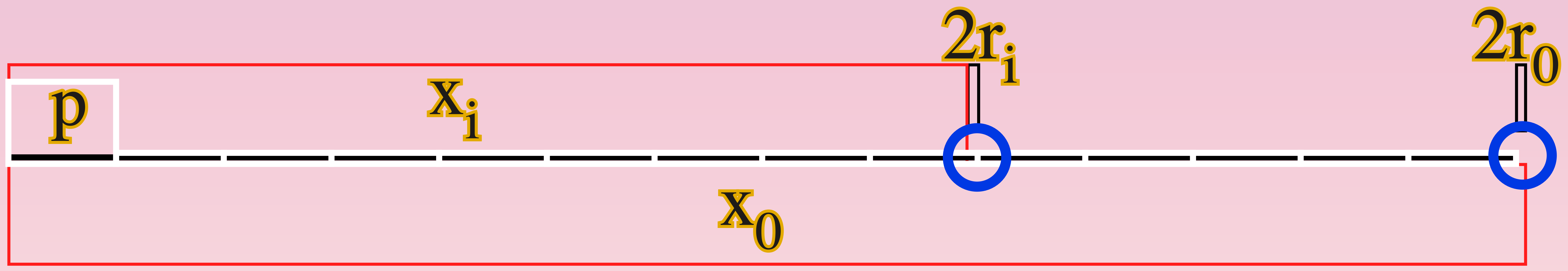


$$\begin{tikzcd}
enc_A(m_0) \arrow[r] & U0 \\
enc_A(m_1) \arrow[r] & U1 \\
\mathfrak{Z}_c & RsR_A(®,Uc) \arrow[l]
\end{tikzcd}$$

Approximate Integer GCD

Let p be a large odd integer. Define several (k) x_i's as follows

$$x_i = pq_i + 2r_i \qquad \text{with } x_i \gg p \gg \sum |r_i|$$



w.l.o.g. assume x_0 = largest x_i and q_0 is odd.

Approximate Integer GCD

Let p be a large odd integer. Define several (k) x_i's as follows

$$x_i = pq_i + 2r_i \qquad \text{with } x_i \gg p \gg \sum|r_i|$$

Define the following public-key encryption function :

$$\varepsilon_x(s,e,b) = (\sum_{i>0} x_i s_i + 2e + b) \bmod x_0$$

where b = input bit, s = rand bin vector, and rand error $|e| \ll p$

$$b = \text{parity}(\varepsilon_x(s,e,b) - \text{nearest multiple of } p)$$

Approximate Integer GCD

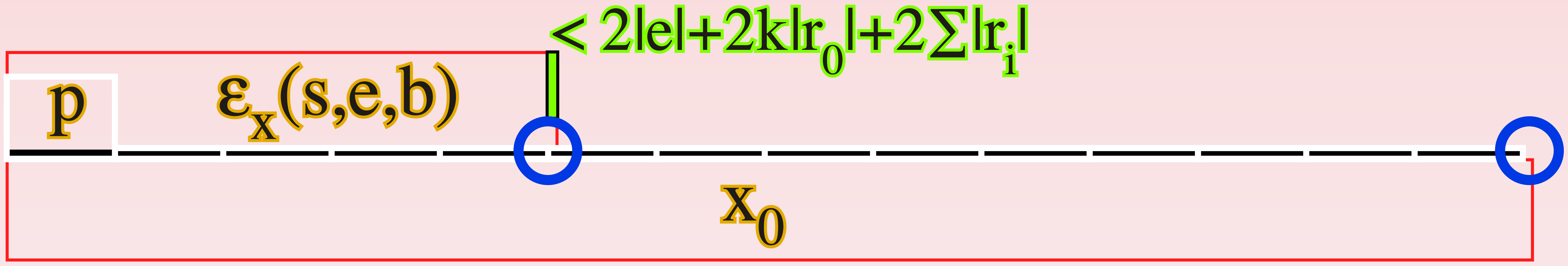
Let p be a large odd integer. Define several (k) x_i's as follows

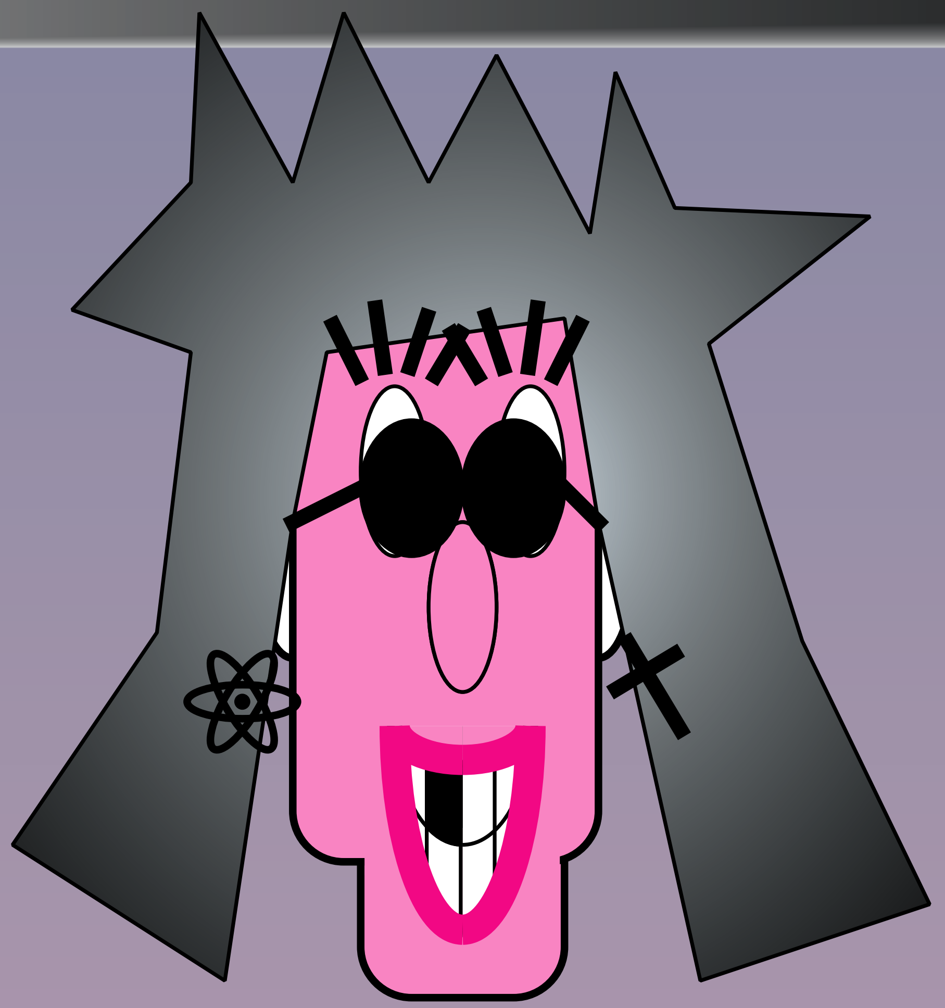
$$x_i = pq_i + 2r_i \qquad \text{with } x_i \gg p \gg \sum|r_i|$$

Define the following public-key encryption function :

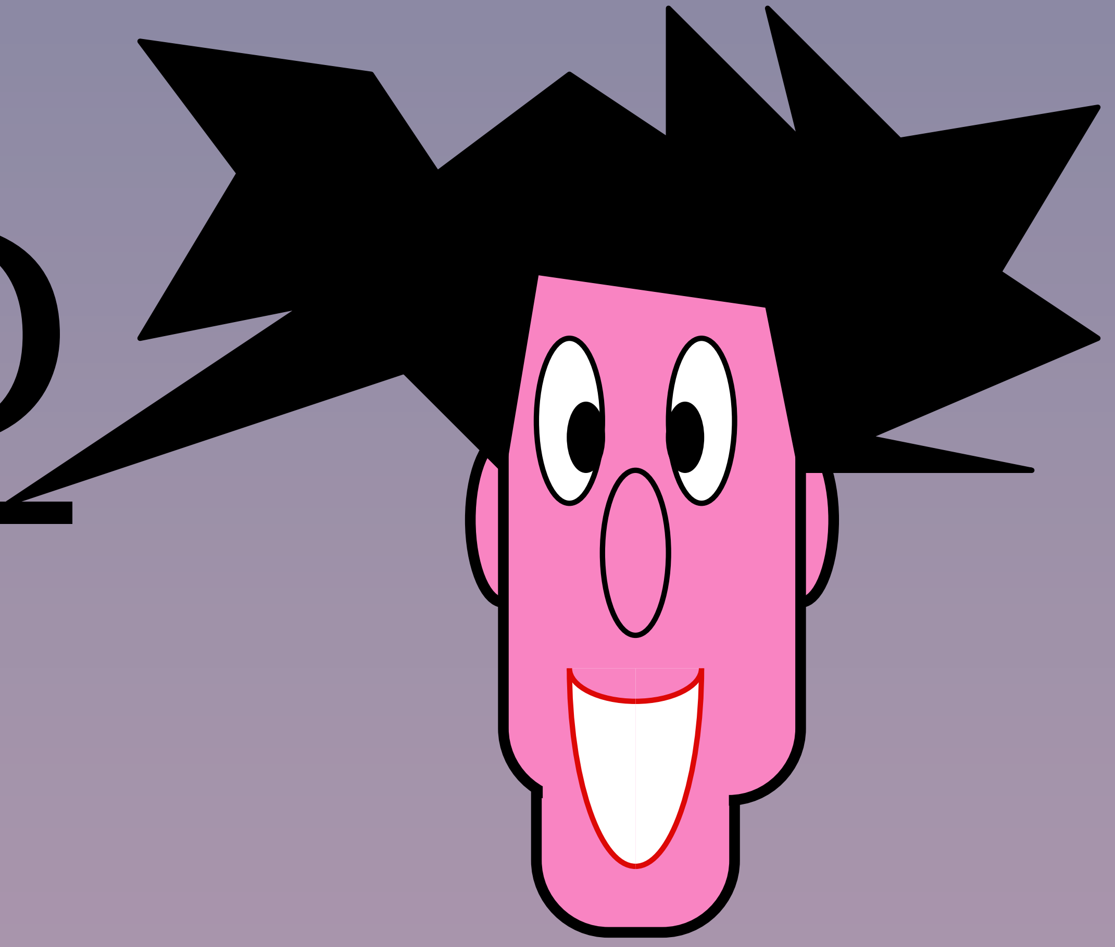
$$\varepsilon_x(s,e,b) = (\, \sum_{i>0} x_i s_i + 2e + b \,) \bmod x_0$$

where b = input bit, s = rand bin vector, and rand error $|e| \ll p$





Appr Int GCD



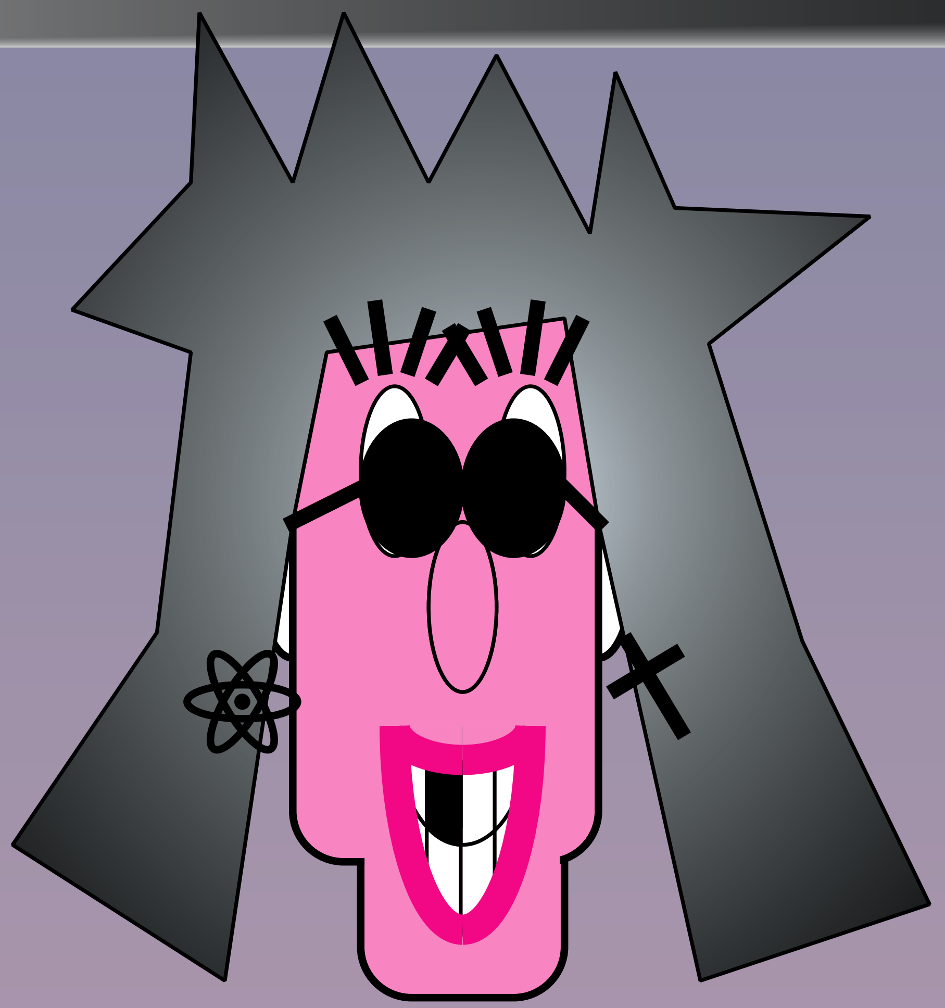
$$\varepsilon_x(R_0,E_0,B_0) \longrightarrow U0$$

$$\varepsilon_x(R_1,E_1,B_1) \longrightarrow U1$$

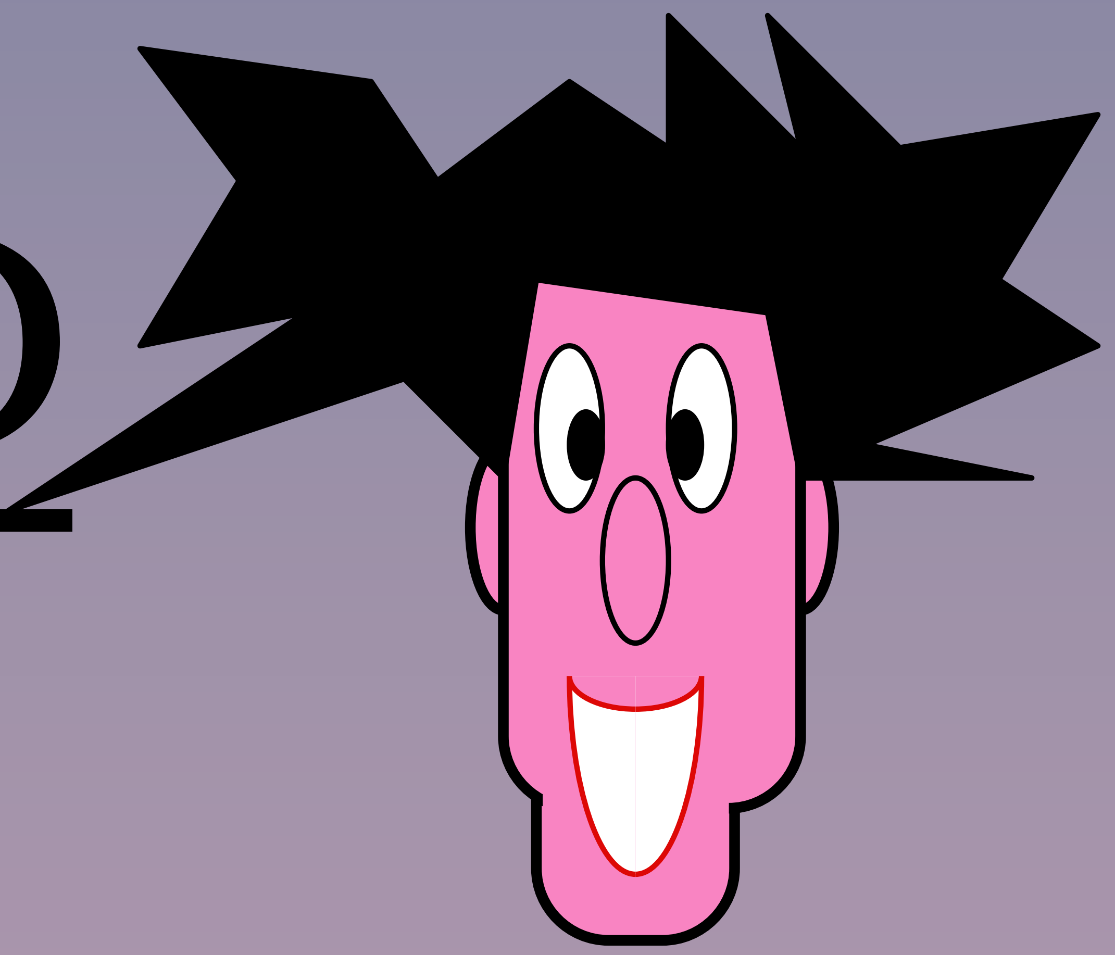
$$\mathcal{Z}_c \longleftarrow \varepsilon_x(s,e,b)+U_c$$

Secure if $|E_0|+|E_1|+k|r_0|+\sum|r_j| \ll |e|$

$$\varepsilon_x(s,e,b)+U_0 \sim \varepsilon_x(s',e,0) \sim \varepsilon_x(s,e,b)+U_1$$

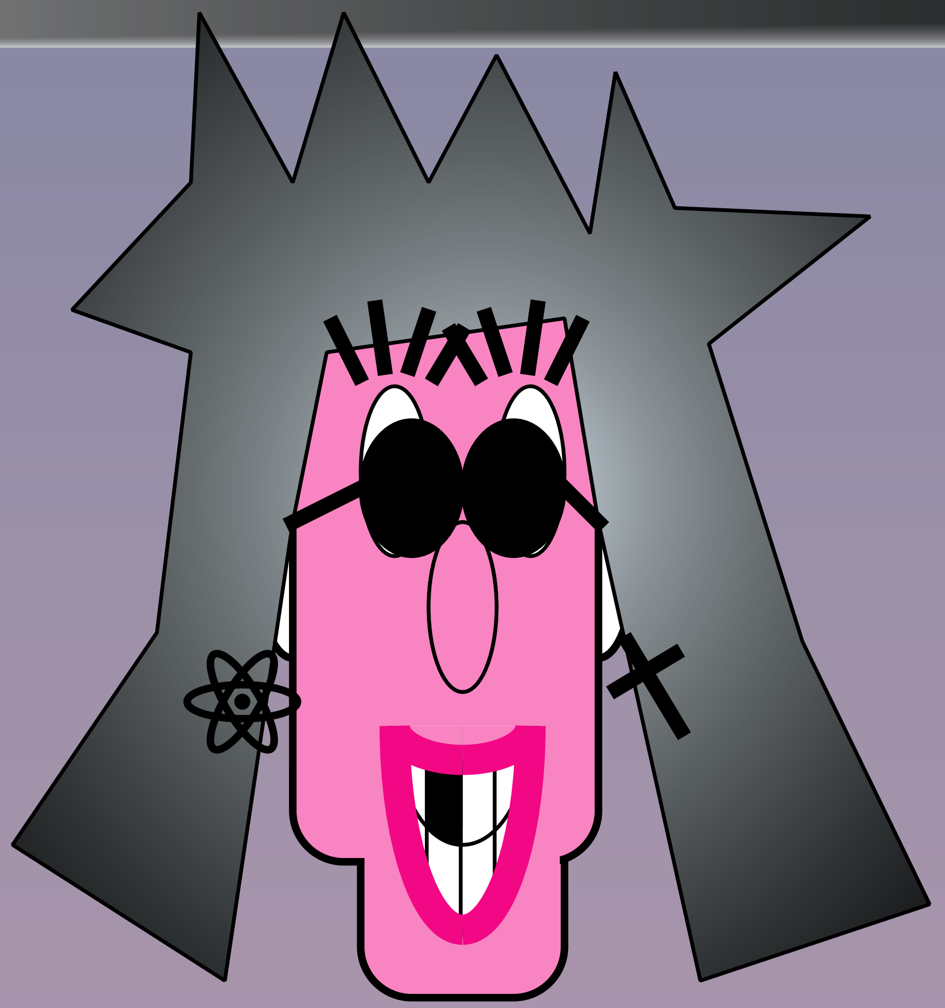


Appr Int GCD

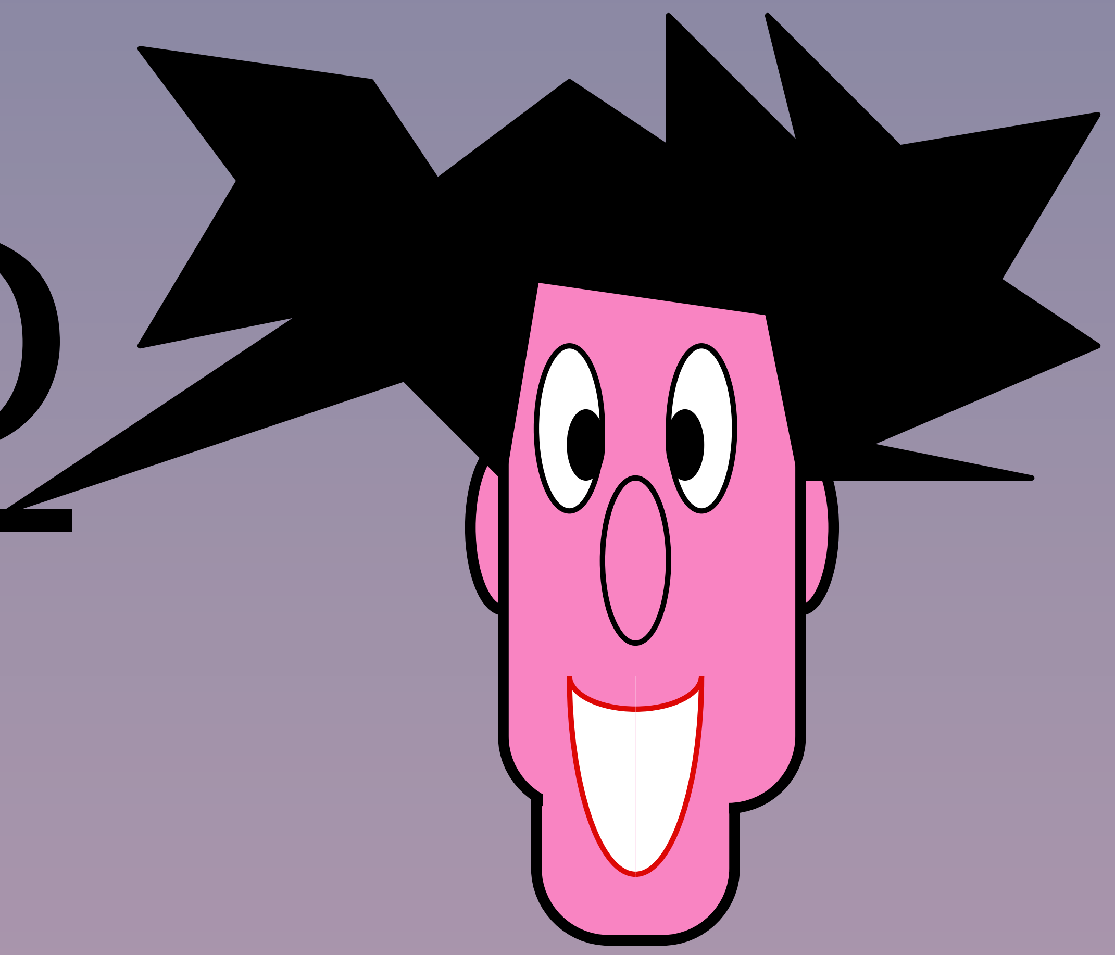


$$\begin{tikzcd}
\varepsilon_x(R_0,E_0,B_0) \arrow[r] & U_0 \\
\varepsilon_x(R_1,E_1,B_1) \arrow[r] & U_1 \\
\mathfrak{Z}_c & \varepsilon_x(s,e,b)+U_c \arrow[l]
\end{tikzcd}$$

Insecure if $|E_i| \gg |e|$ or $|r_j| \gg |e|$!



Appr Int GCD



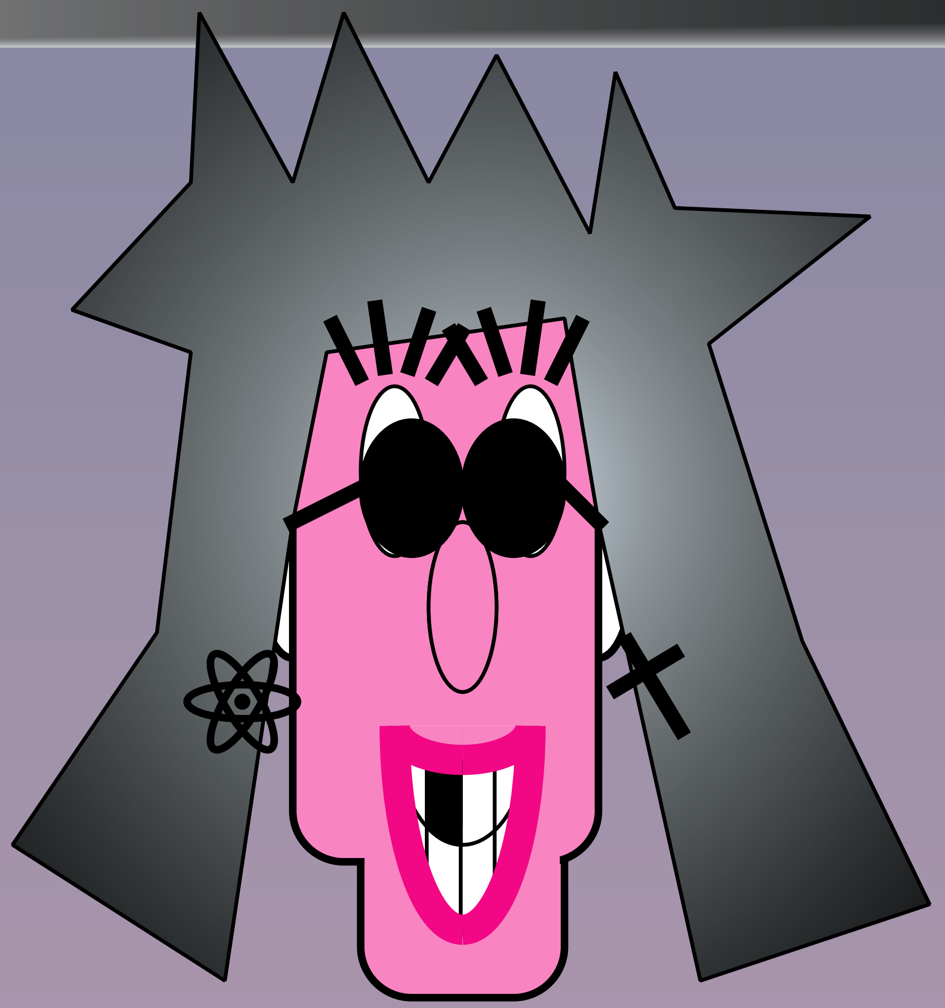
$$\varepsilon_x(R_0,E_0,B_0) \longrightarrow U_0$$

$$\varepsilon_x(R_1,E_1,B_1) \longrightarrow U_1$$

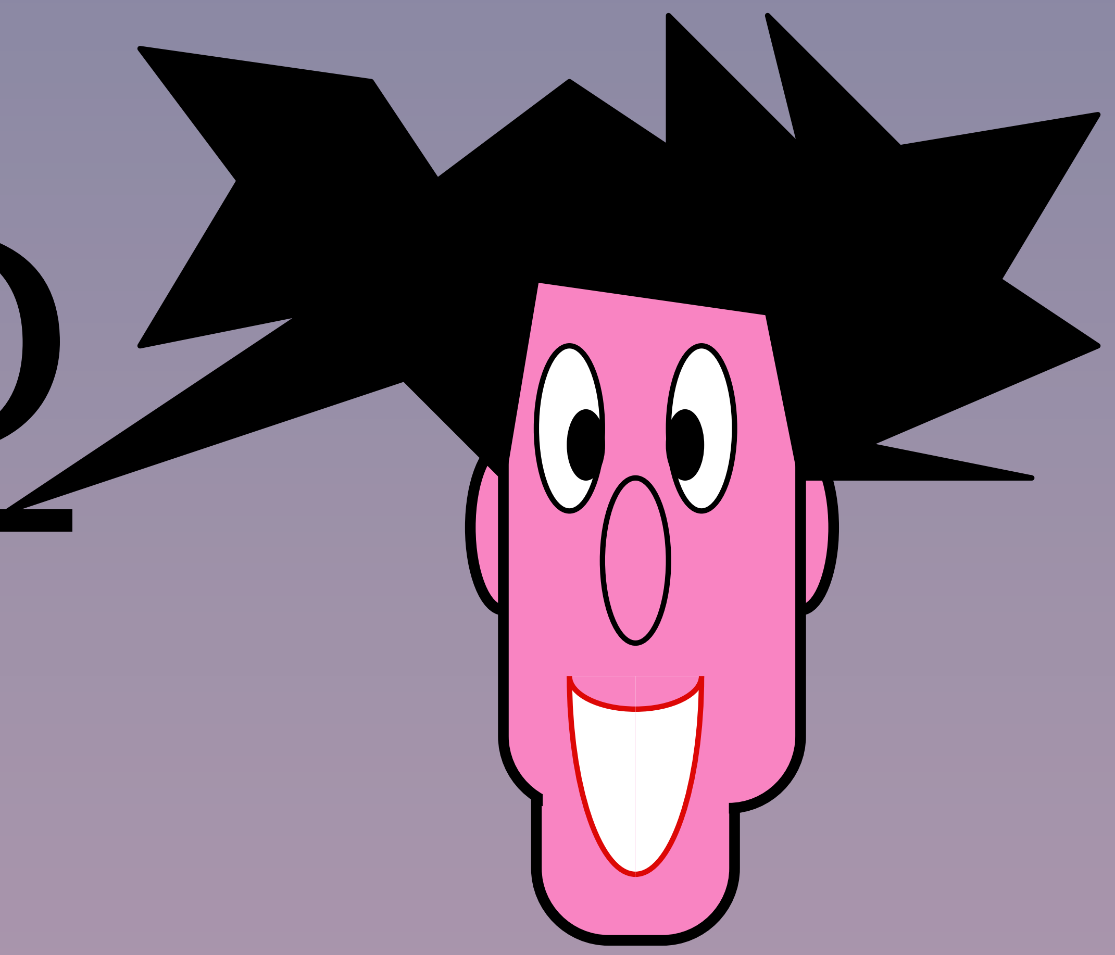
$$z \longleftarrow \varepsilon_x(s,e,b)+U_c$$

Prove in ZK that $|E_0|+|E_1|+k|r_0|+\sum|r_j| \ll$ Public Bound.

Use $|e| \gg$ Public Bound.



Appr Int GCD



$$\varepsilon_x(R_0,E_0,B_0) \Longrightarrow U_0$$

$$\varepsilon_x(R_1,E_1,B_1) \Longrightarrow U_1$$

$$z \Longleftarrow \varepsilon_x(s,e,b)+U_c$$

$$\delta_A(z) \Longrightarrow y$$

$$B_c=y(+)b$$

(4)

Conclusion

Open

Problems

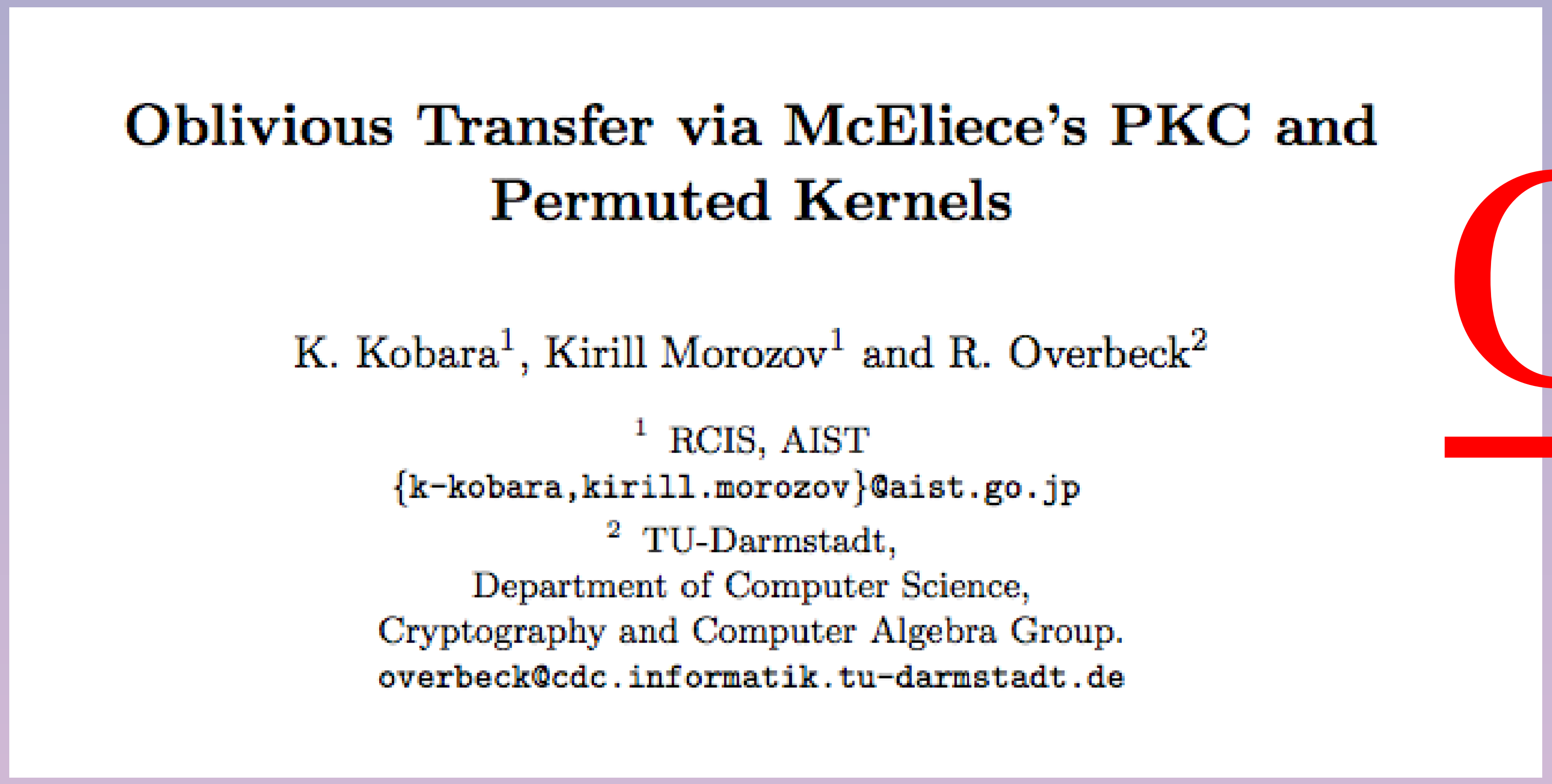
Quantum Weakly Random-Self-Reducible Encryption Scheme

~~McEliece~~

Lattices

Integer GCD

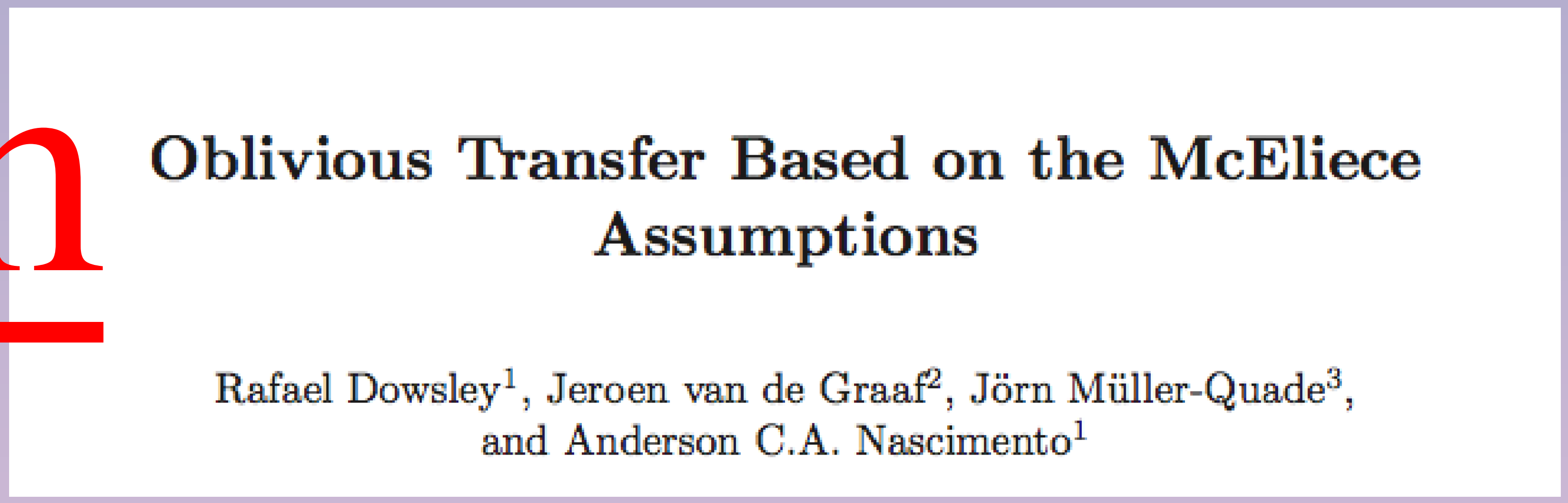
LWE


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

Open Problem


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]

McEliece

Find a pair of PPT algorithms (RsR,RsR^{-1}) such that for all ®,m,

$$\text{RsR}^{-1}(®,\text{dec}(\text{RsR}(®,\text{enc}(m))) = m$$

and there exists a PPT ditribution on ® s.t. for all m,m'

$$\text{RsR}(®,\text{enc}(m)) \sim \text{RsR}(®,\text{enc}(m')).$$

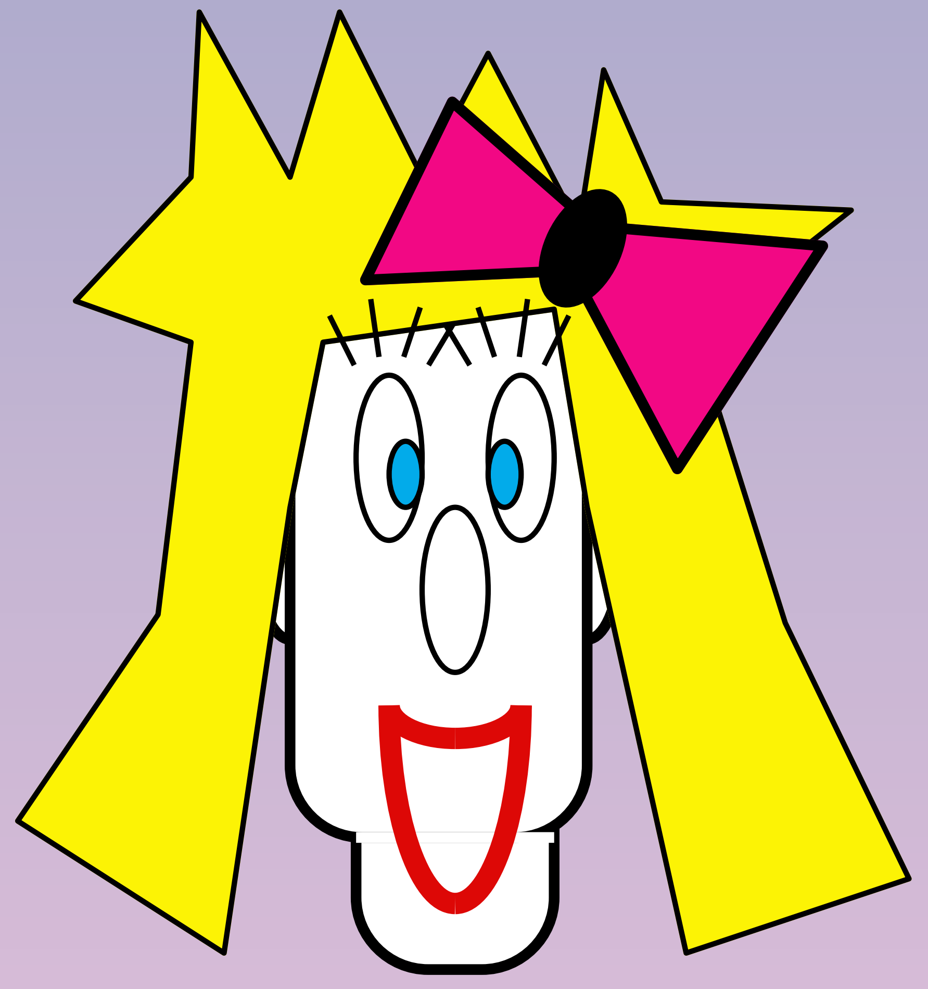
Oblivious Transfer from Weakly Random-Self-Reducible Encryption

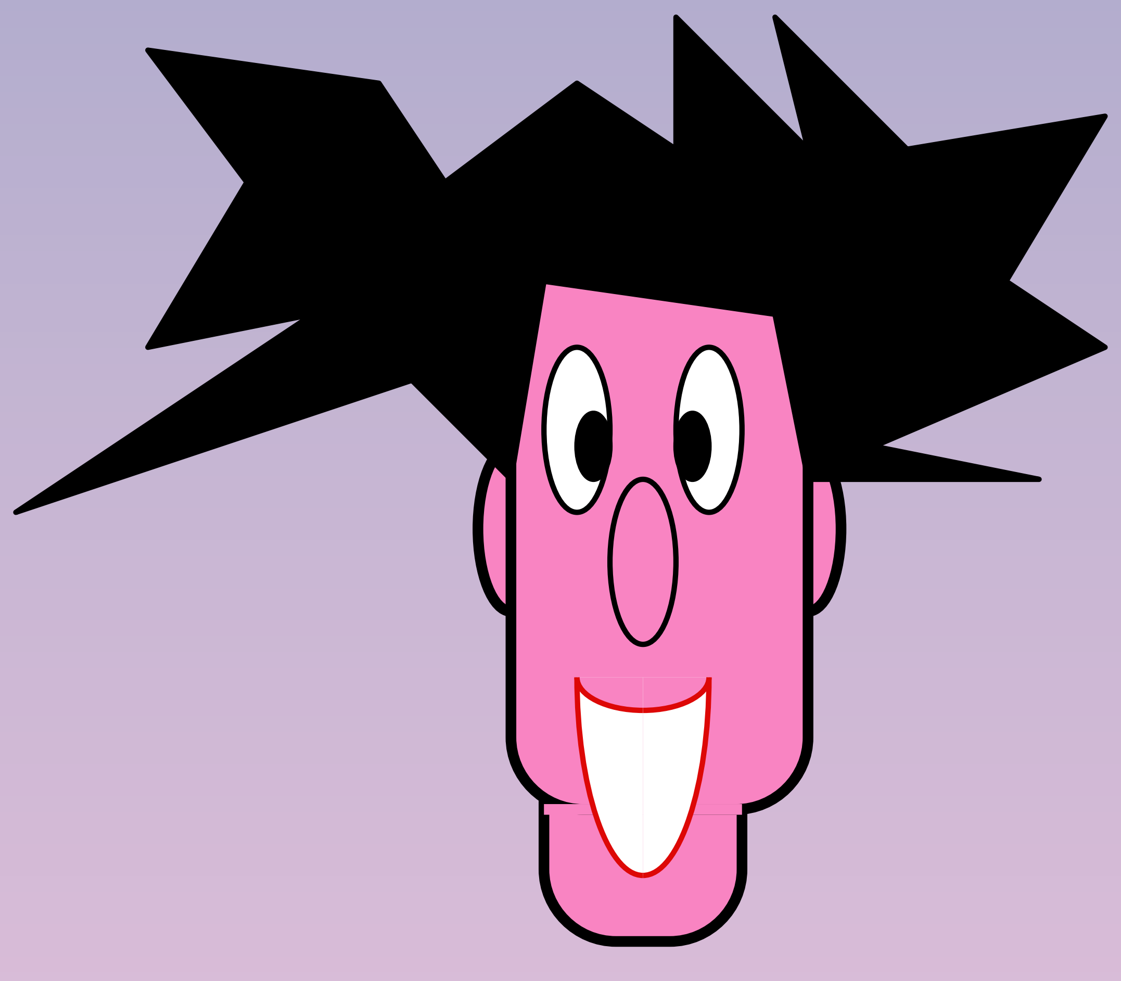
Claude Crépeau

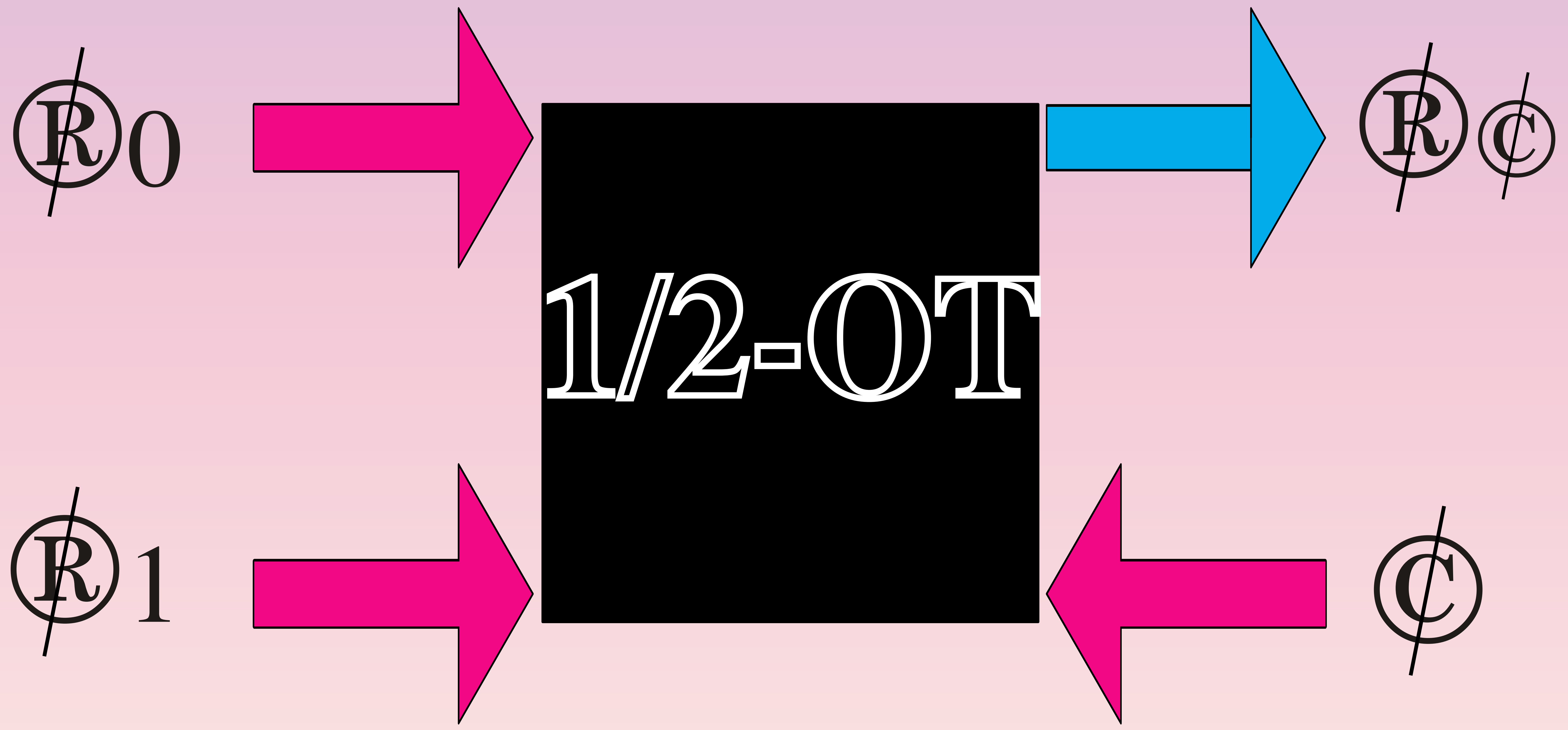
School of Computer Science
McGill University

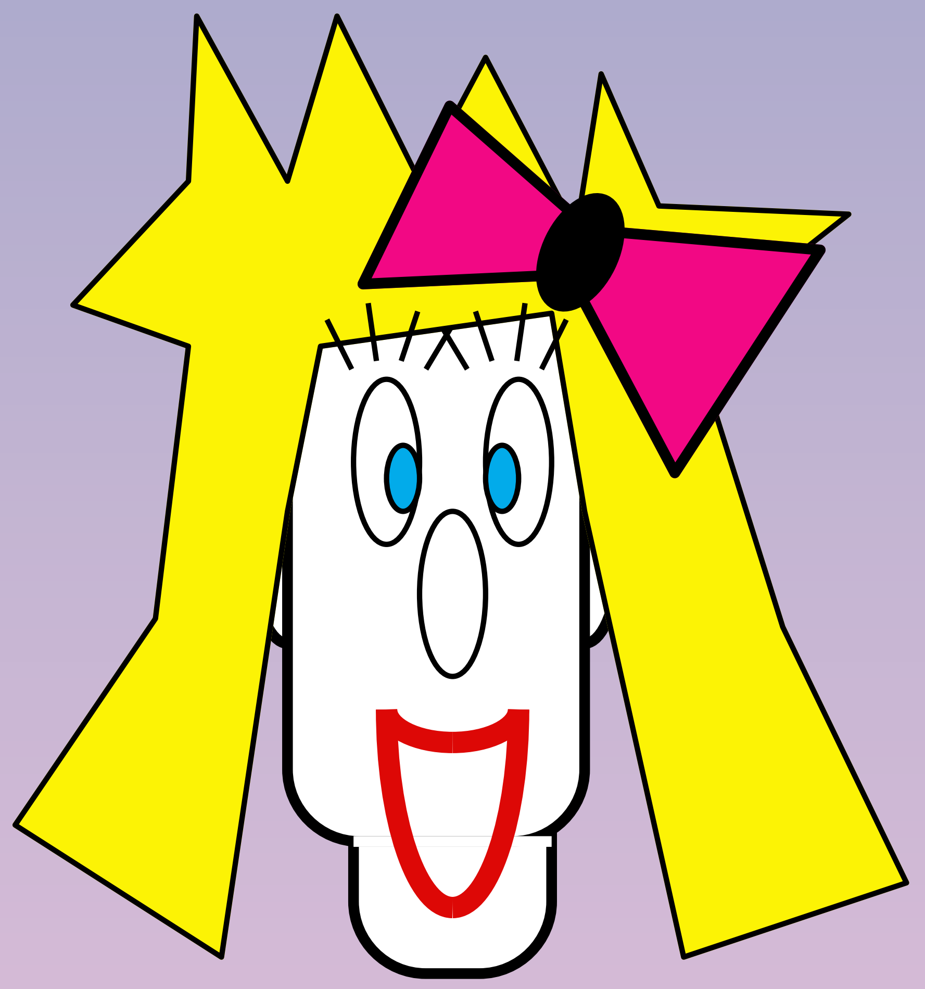
joint work with Raza Ali Kazmi

Randomized Oblivious Transfer

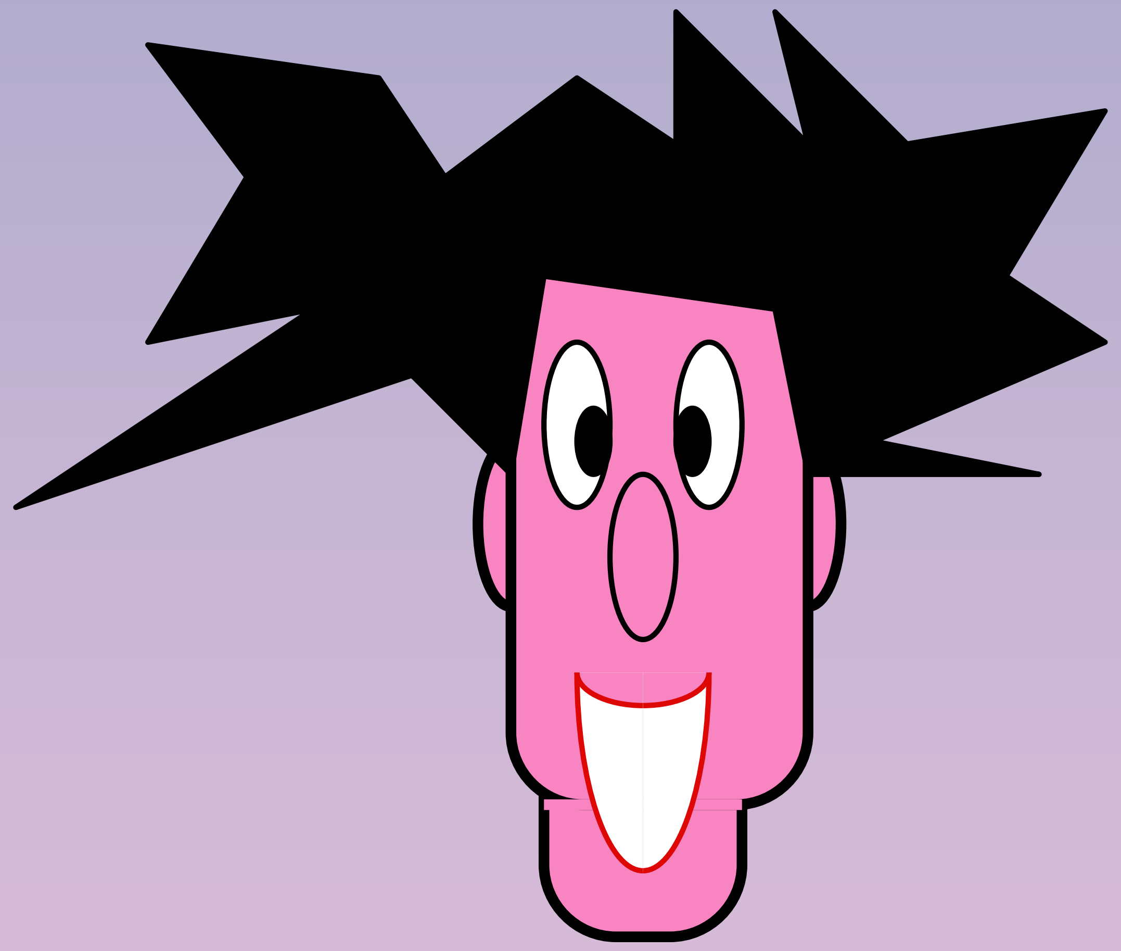


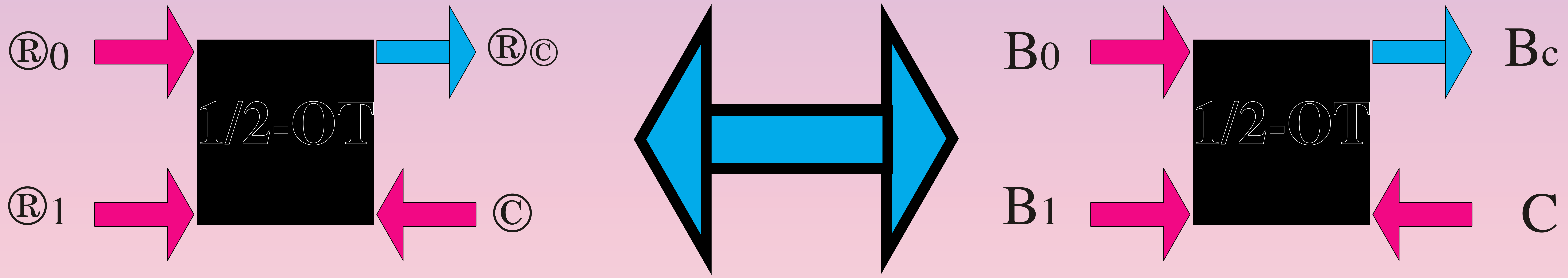


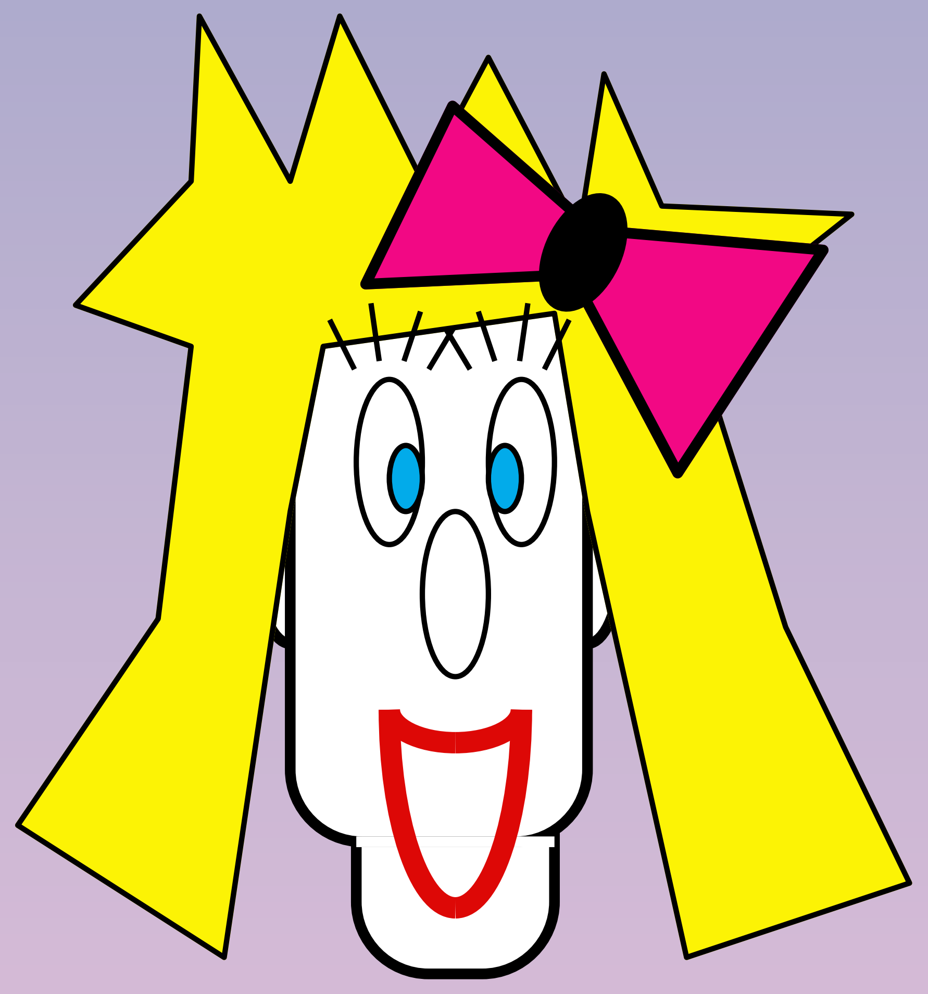




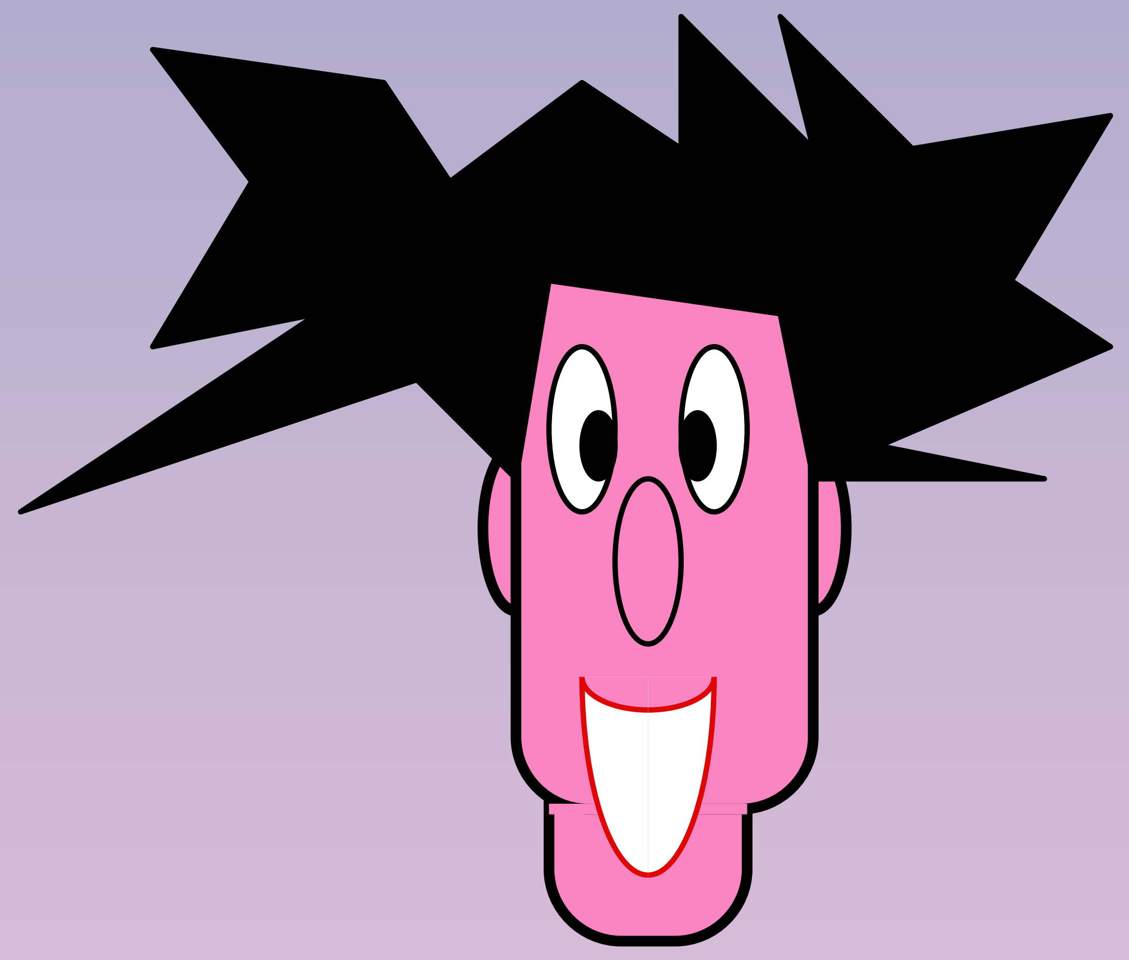
Randomized Oblivious Transfer

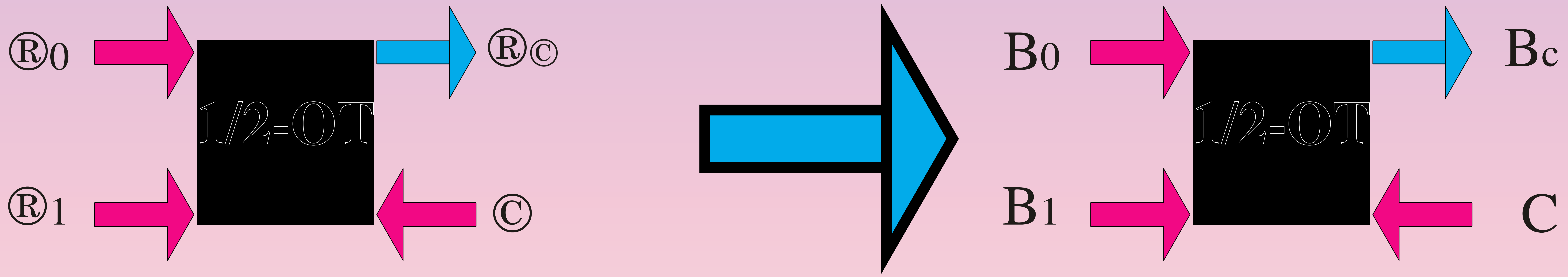


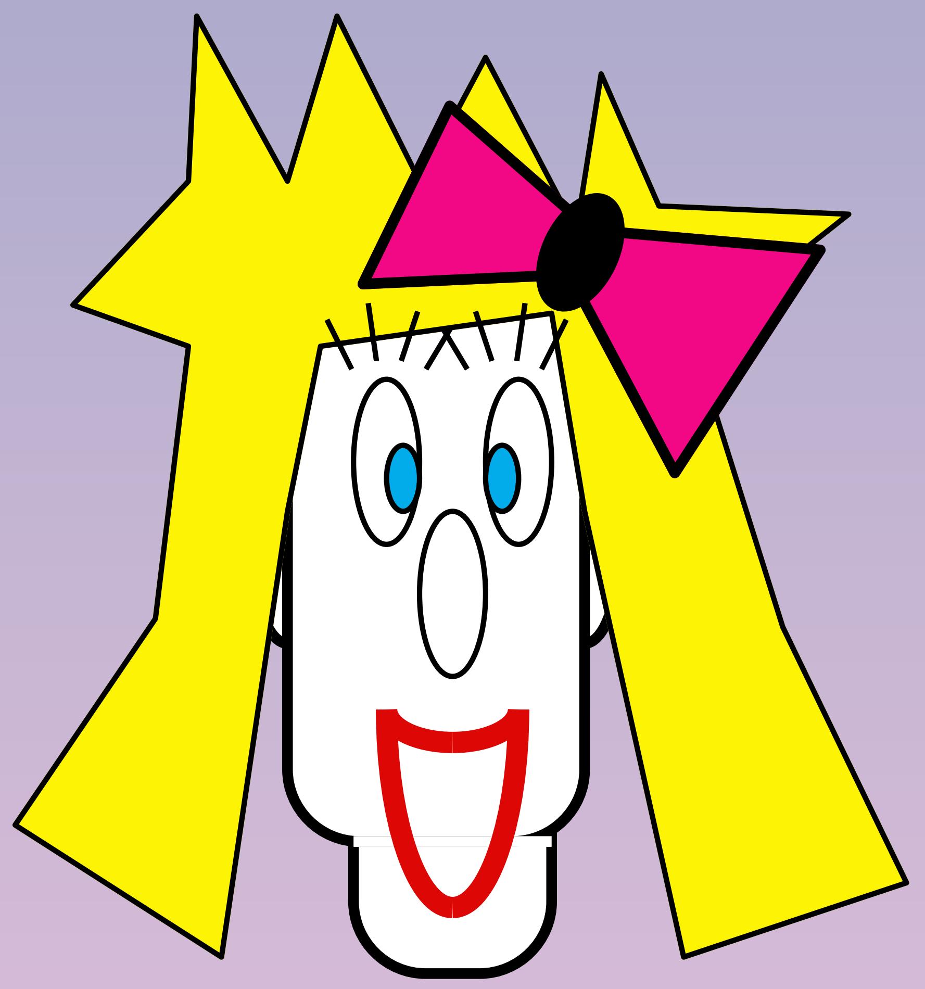




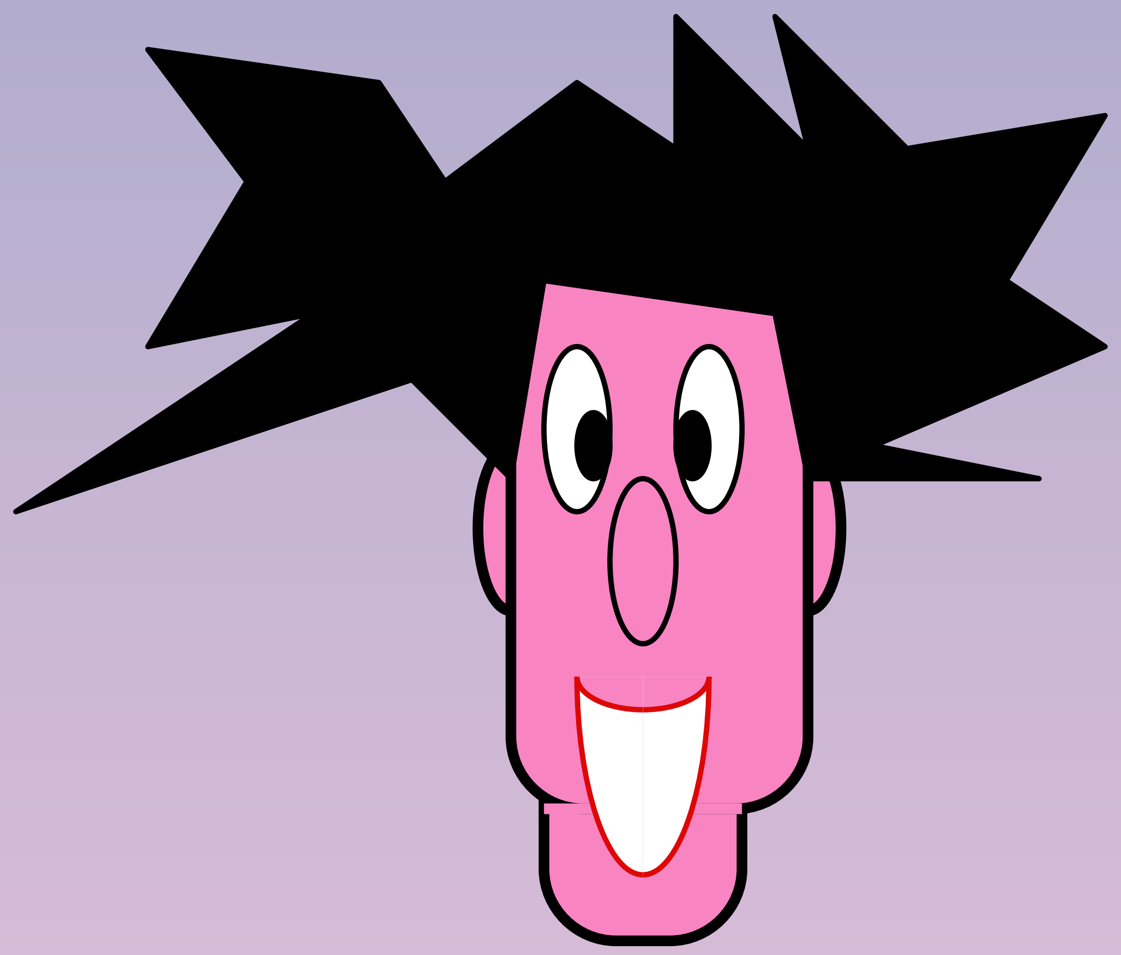
Randomized Oblivious Transfer

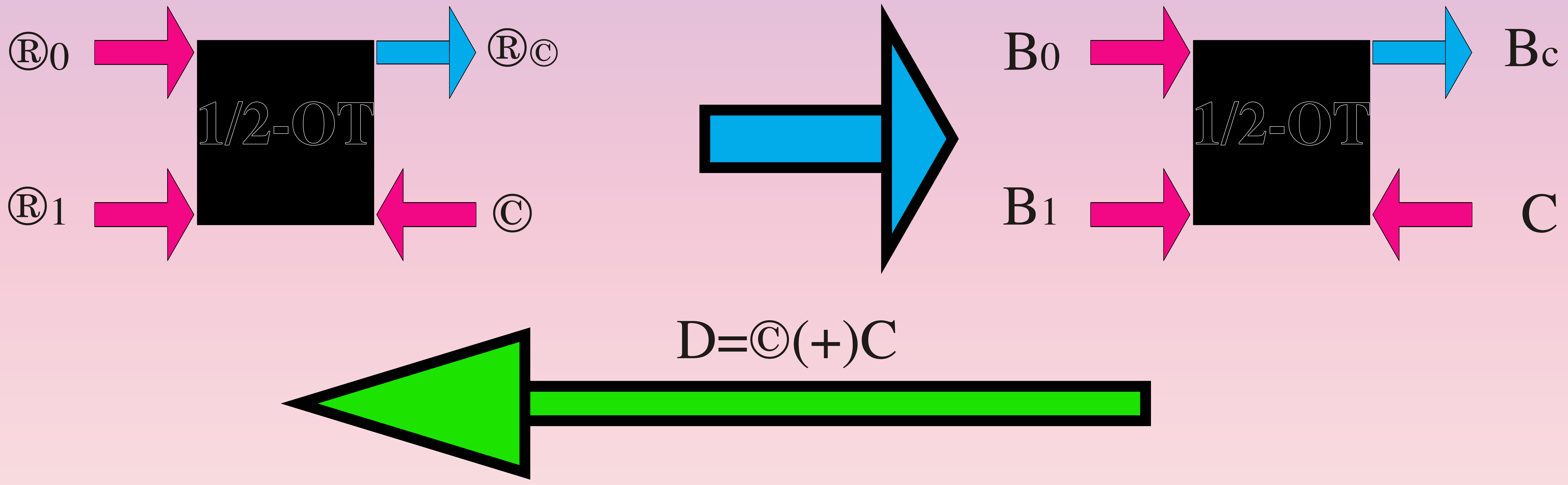


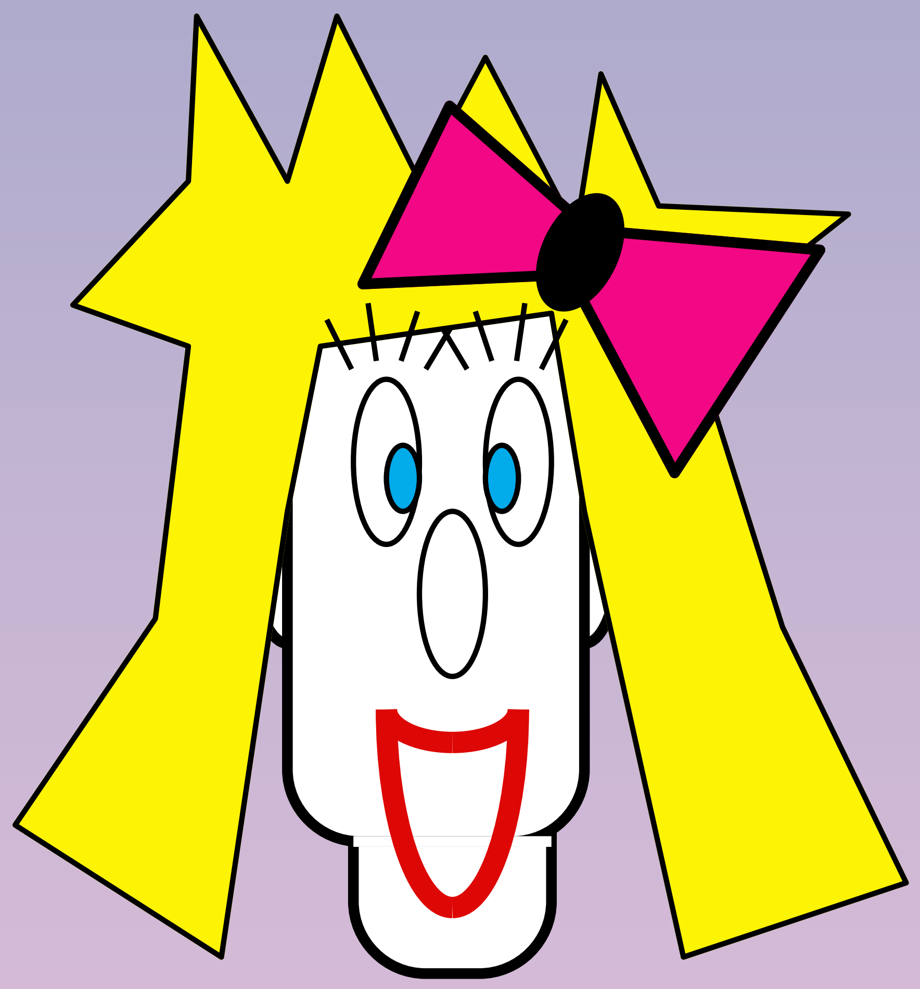




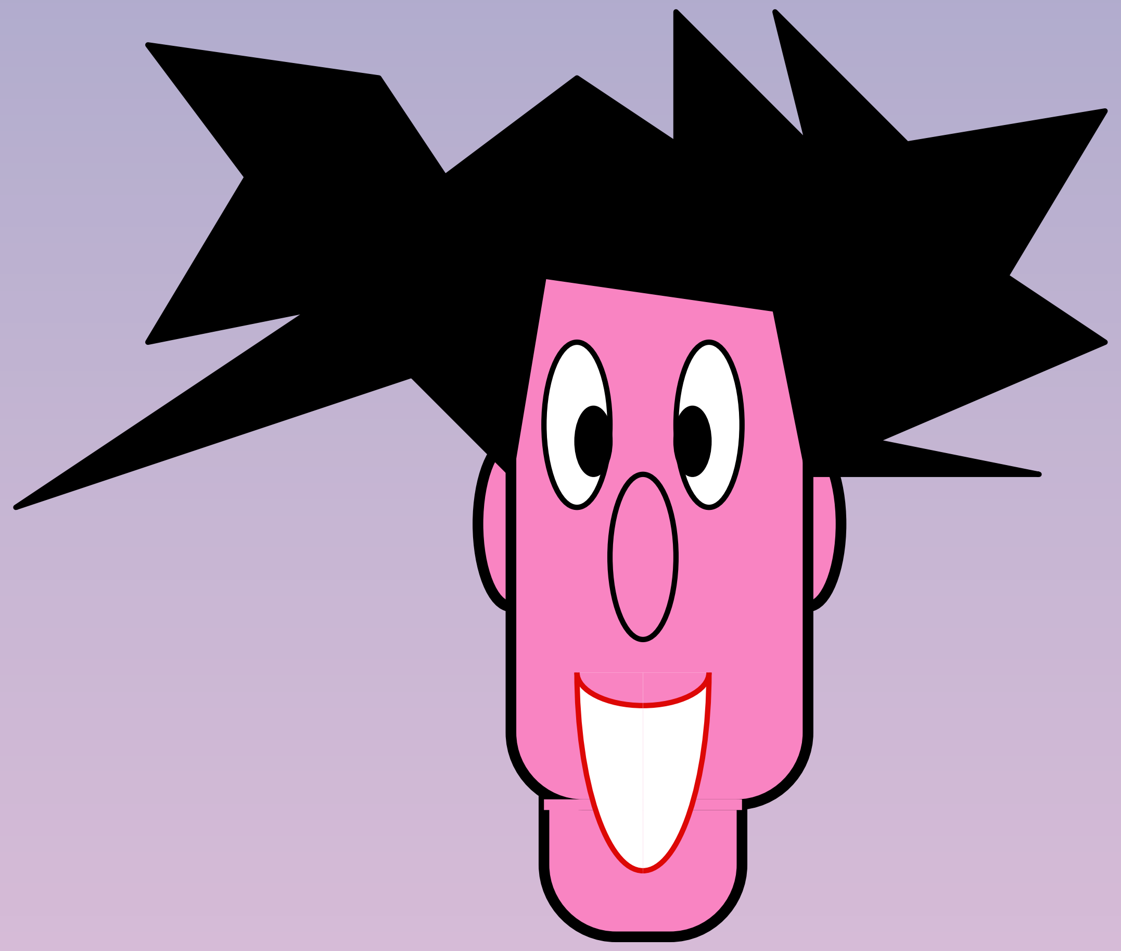
Randomized Oblivious Transfer

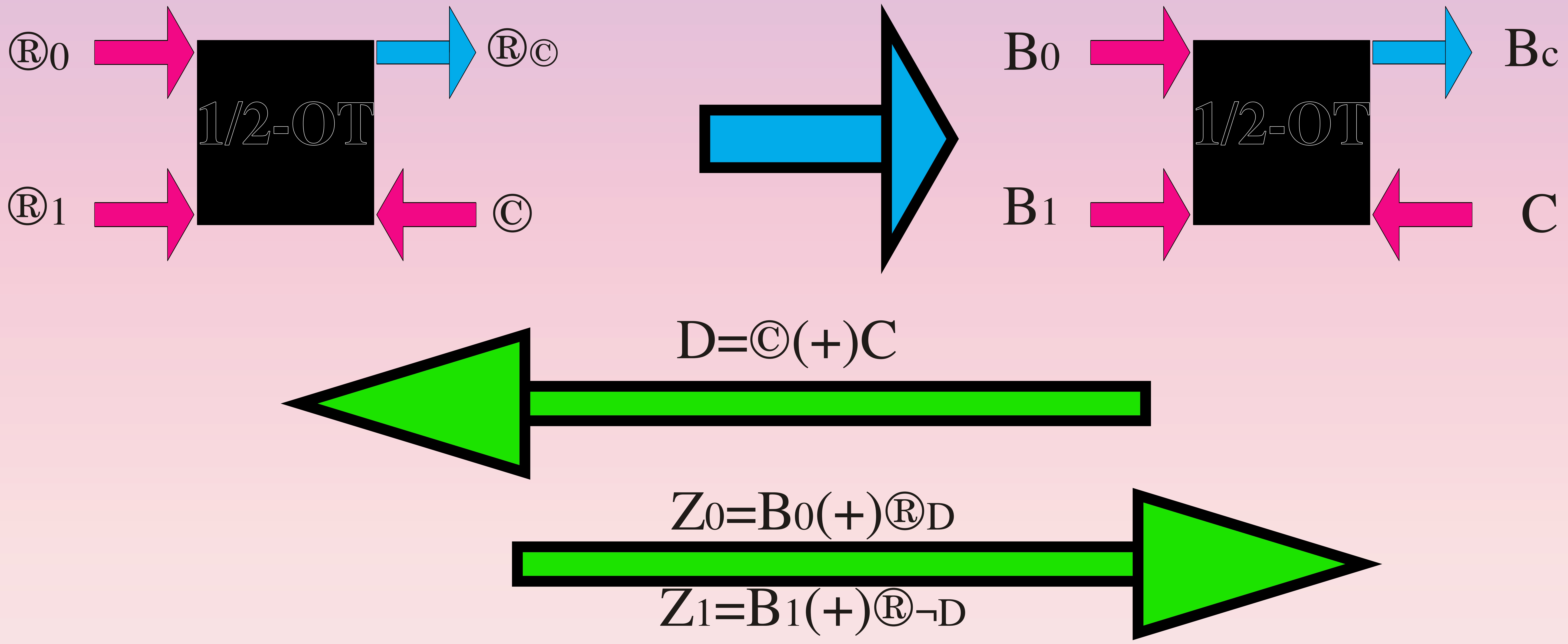




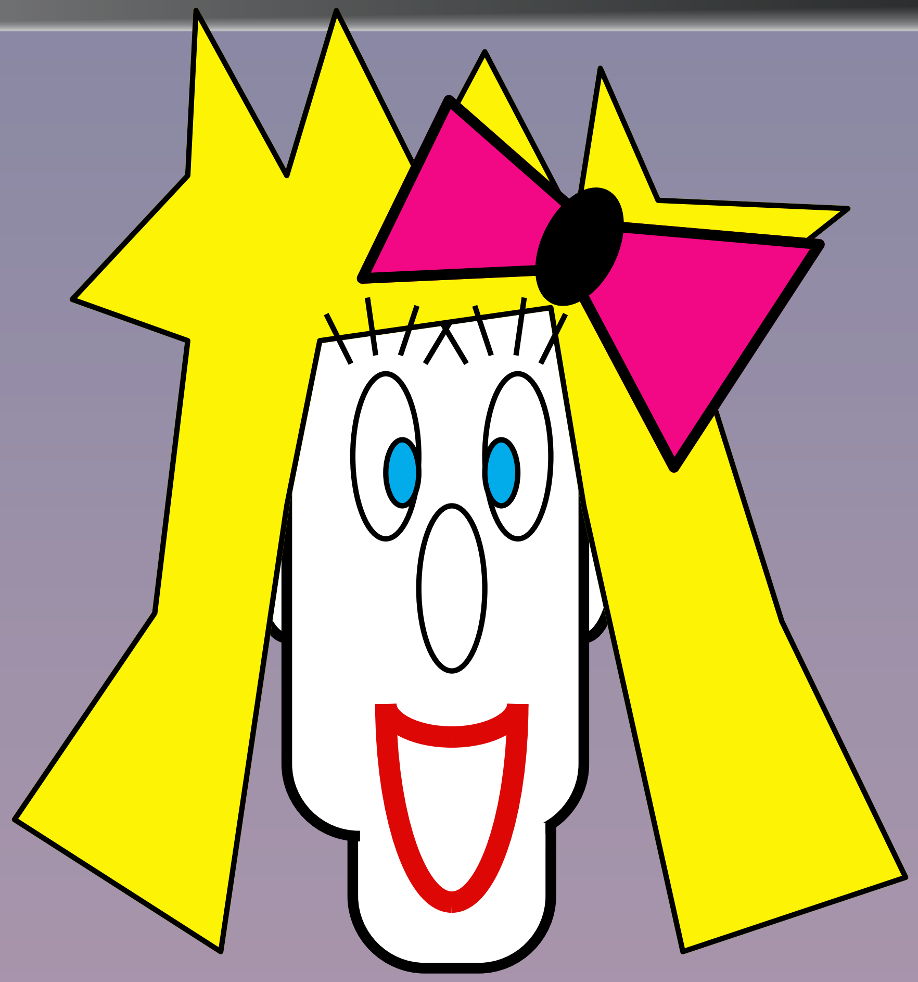


Randomized Oblivious Transfer

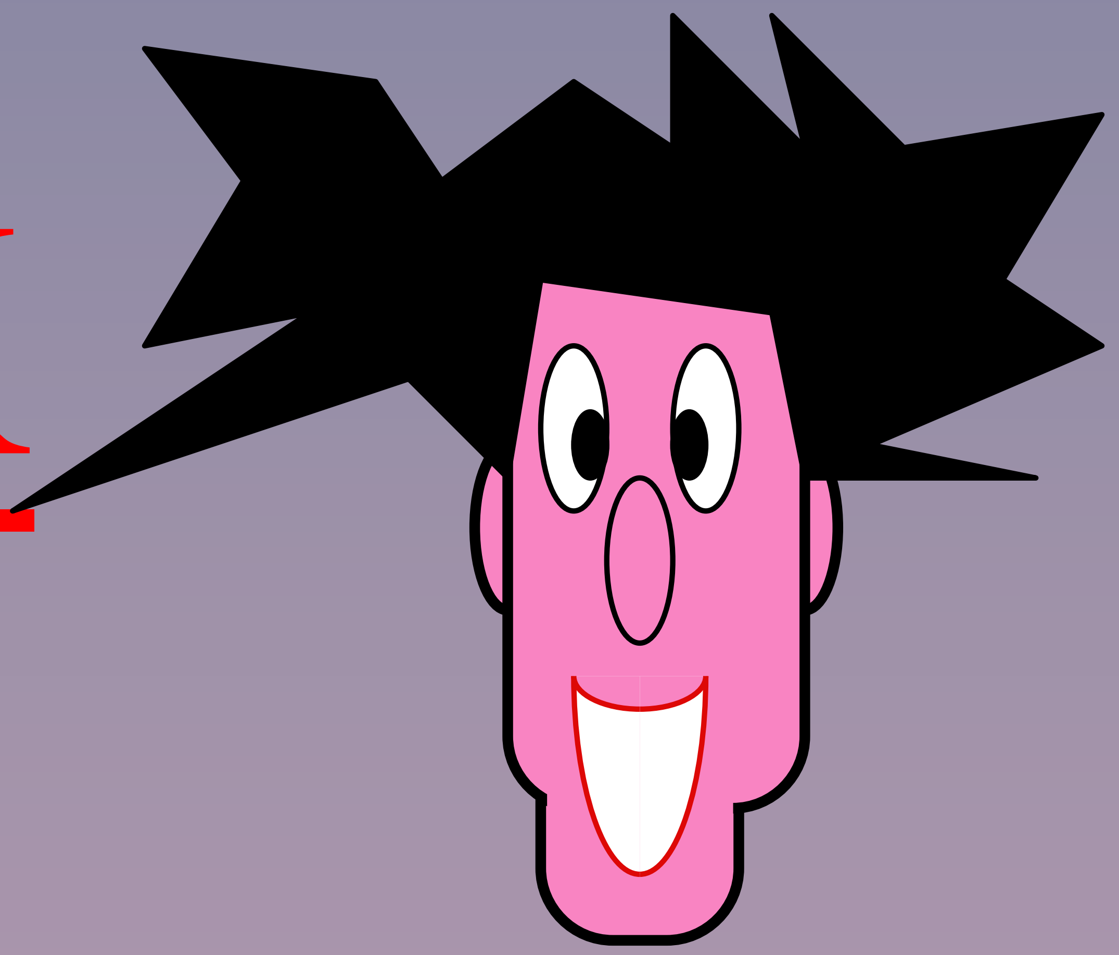




B$_c$=Z$_c$(+)®$_©$



THIS WORK

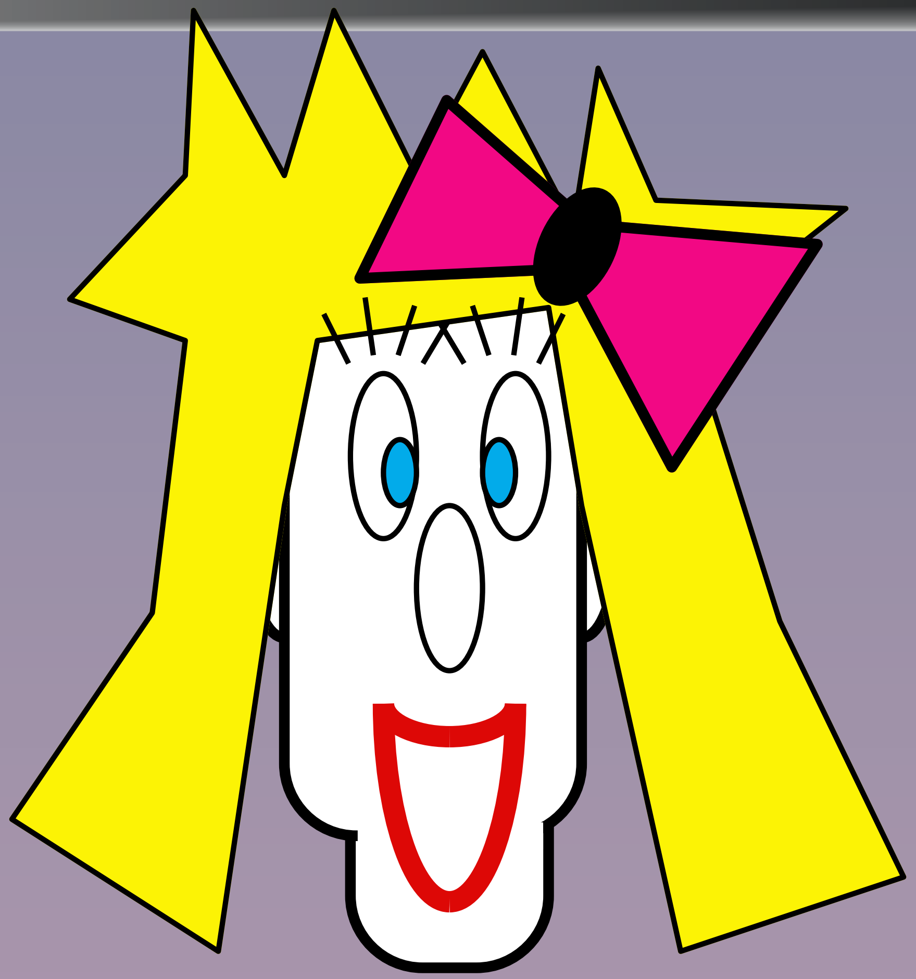


$$U \longleftarrow enc_B(c)$$

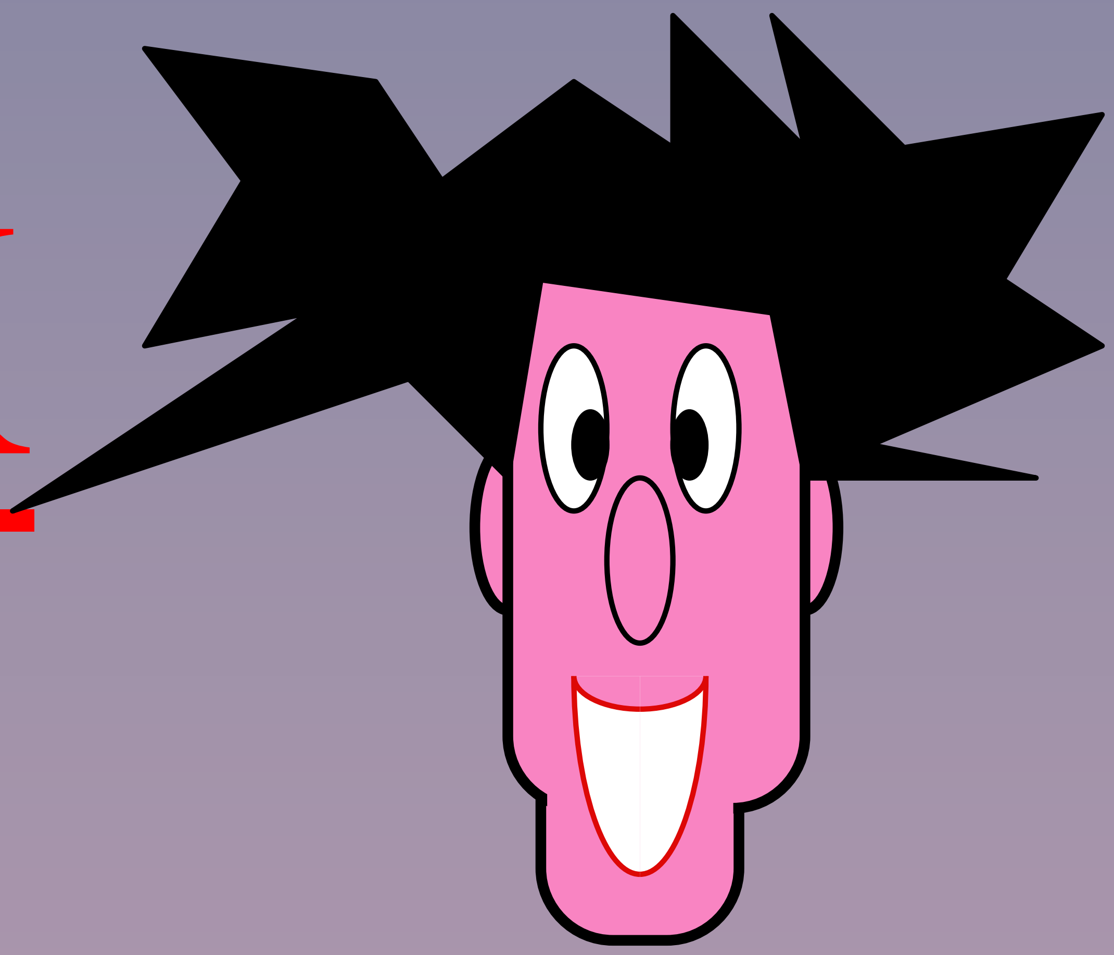
if $B_0=B_1$ then

$\underline{enc_B(B_0)}$

$$\longrightarrow$$



THIS WORK



$U \longleftarrow enc_B(c)$

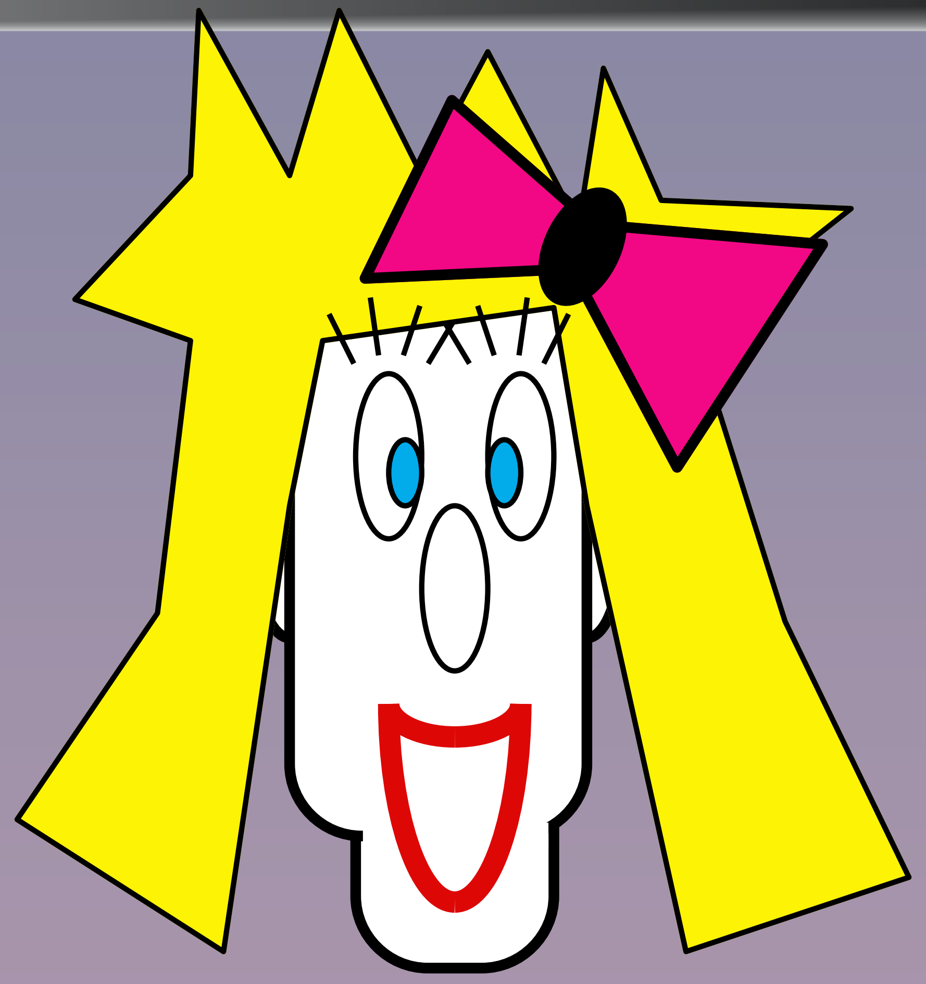
if $B_0=B_1$ then

$\underline{enc_B(B_0)}$

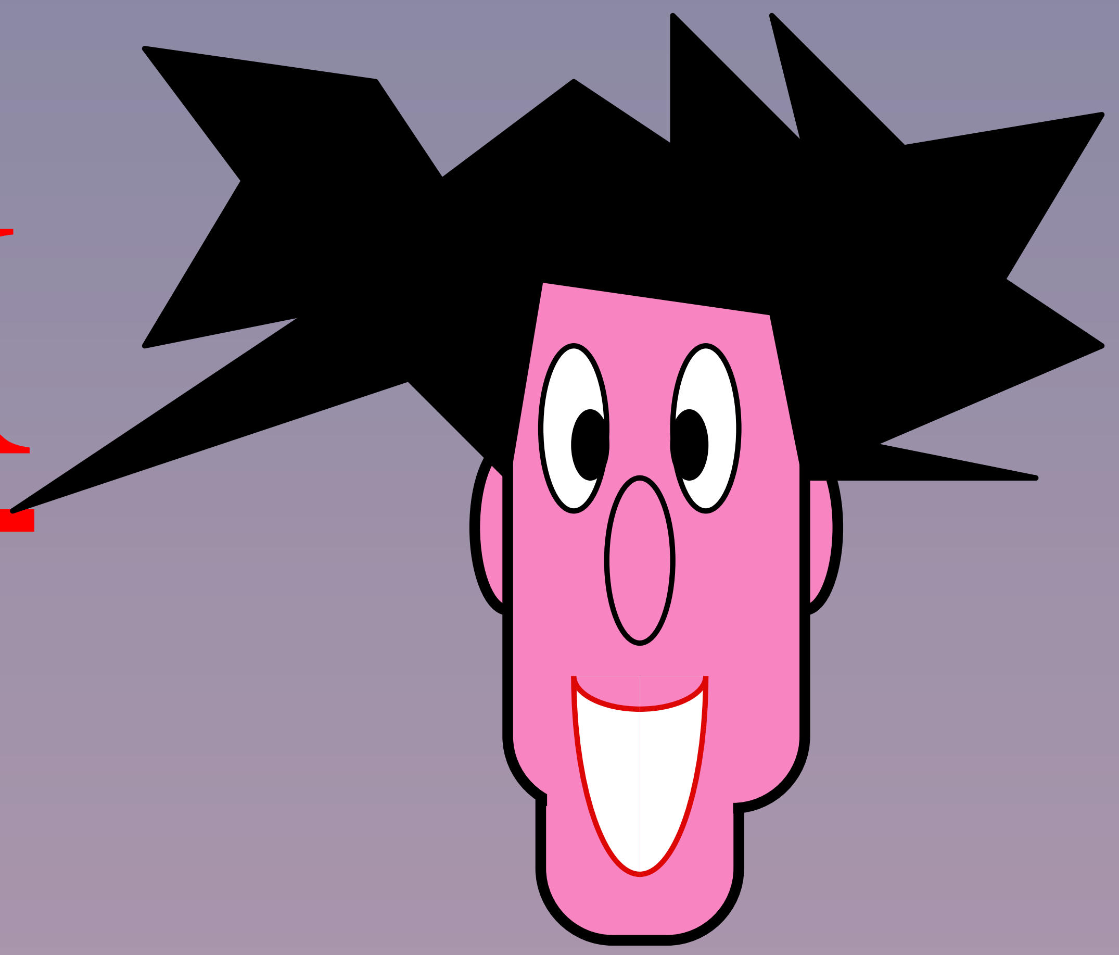
else $\longrightarrow$

$\underline{enc_B(B_0)\bullet U}$

$\sim enc_B(B_0 * c)$



THIS WORK



$U \longleftarrow \mathrm{enc}_B(c)$

if $B_0=B_1$ then

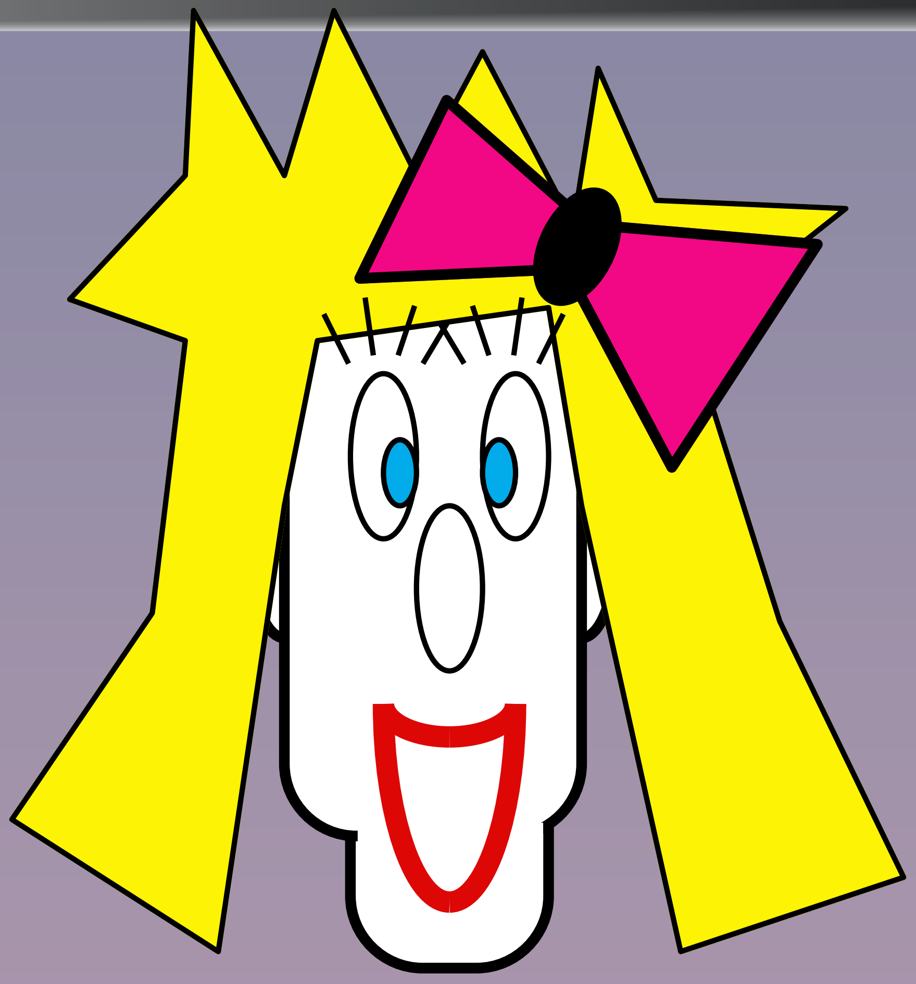
  $\underline{\mathrm{enc}_B(B_0)}$

else

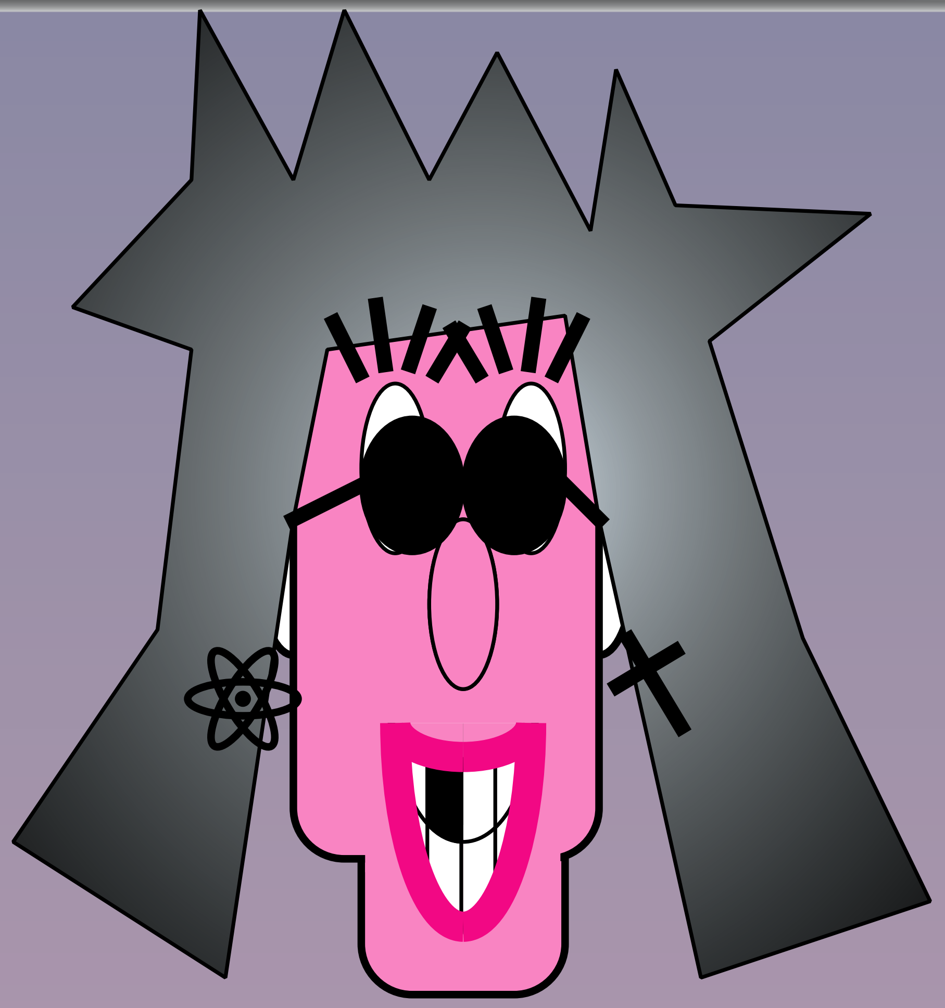
  $\underline{\mathrm{enc}_B(B_0)\bullet U} \longrightarrow z$

$\sim \mathrm{enc}_B(B_0 * c)$

$B_c=\mathrm{dec}_B(z)$



THIS WORK



$$U \longleftarrow \mathrm{enc}_B(c)$$

```
if $B_0 = B_1$ then
    $\underline{\mathrm{enc}_B(B_0)}$
else
    $\underline{\mathrm{enc}_B(B_0) \bullet U}$
$\sim \mathrm{enc}_B(B_0 * c)$
```

$$\longrightarrow z \text{ or } \mathfrak{z}$$