Property Preserving Symmetric Encryption

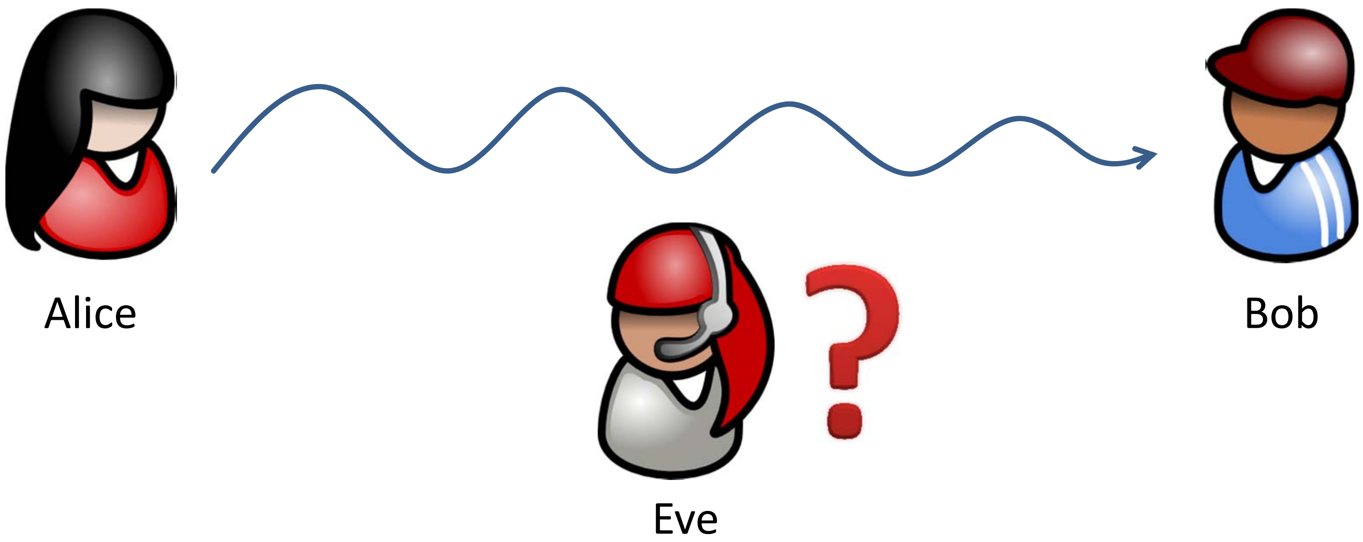
Omkant Pandey

Microsoft, Redmond

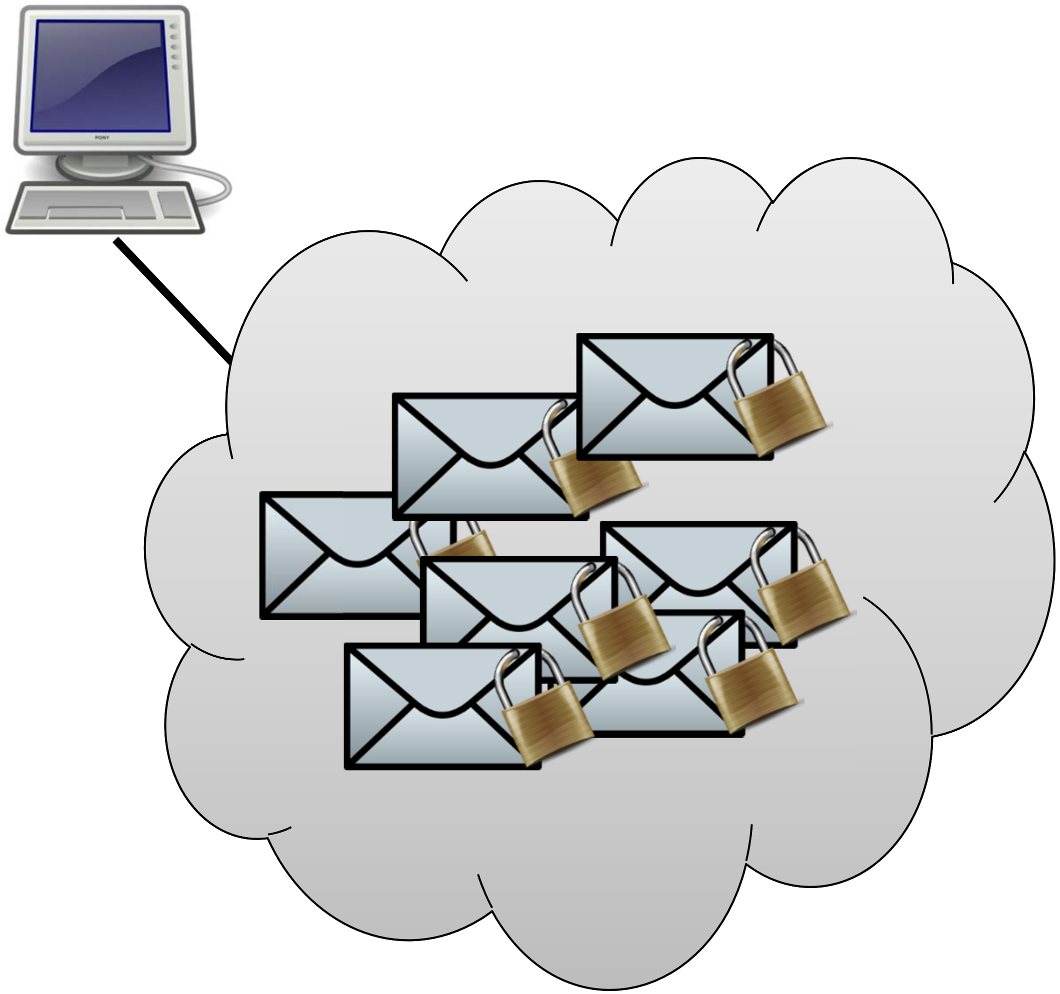
Yannis Rouselakis

University of Texas at Austin

Traditional Cryptography

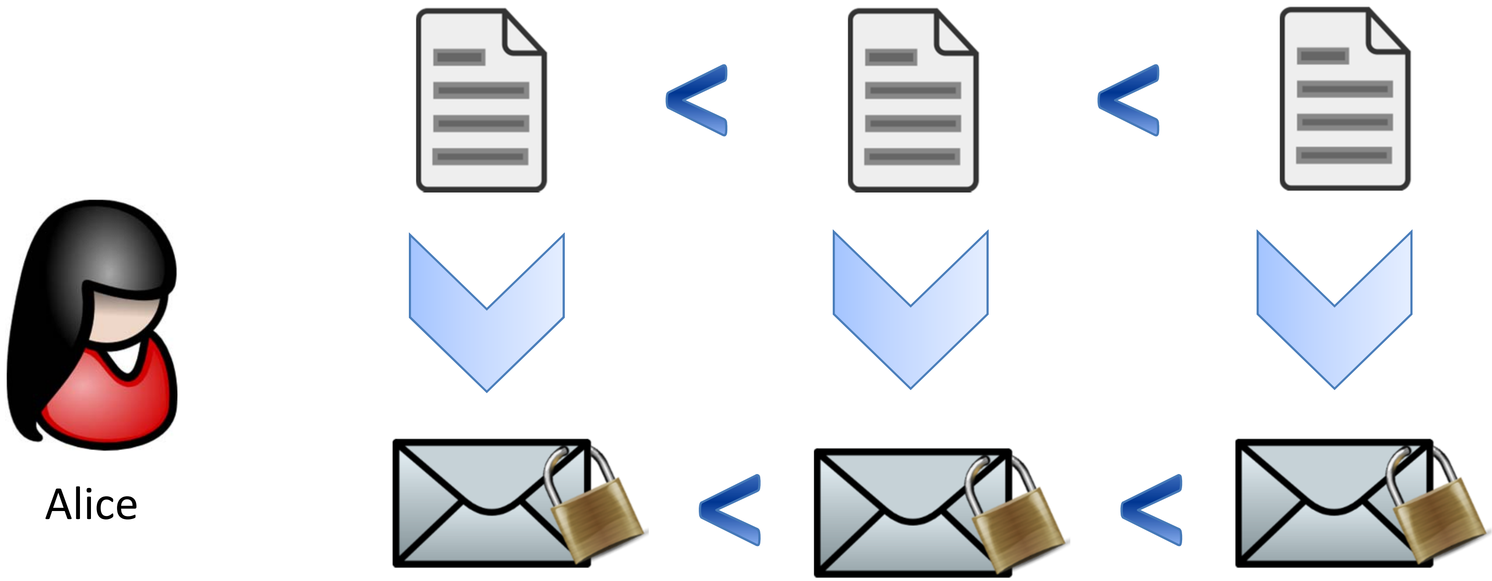


New Goal: Computations on Encrypted Data

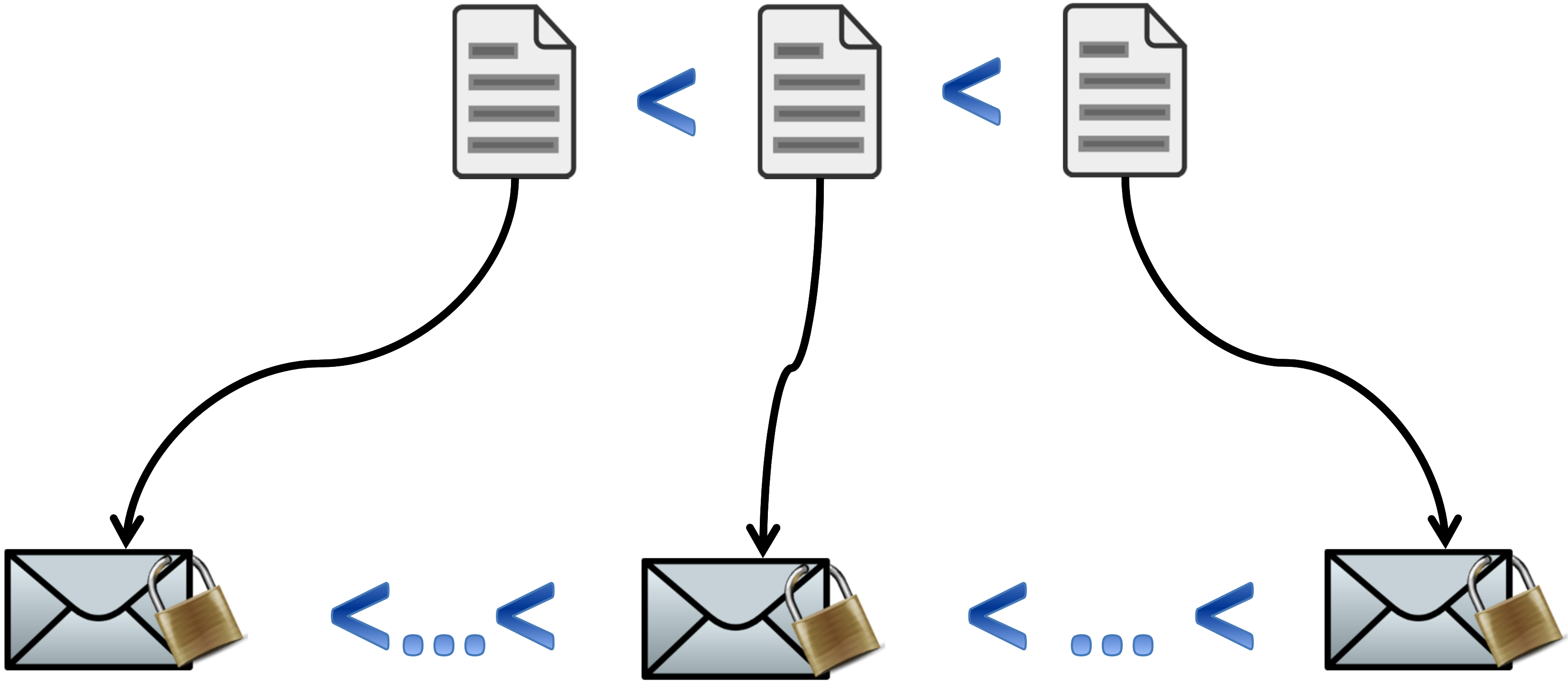


- Indexing
- Range queries
- Data clustering
- Keyword search
- General computations

Order-Preserving Encryption [BCLO09, BCO11]



Order-Preserving Encryption [BCLO09, BC011]



- Exponential size ciphertext space

  Or
- High min-entropy plaintext distributions

Are there properties / schemes with no such restrictions?

Property Preserving Encryption

A property P is a function of arity k

$$P(m_1, m_2, \ldots, m_k) = 0 \text{ or } 1$$

A Property Preserving Encryption (PPE) scheme contains

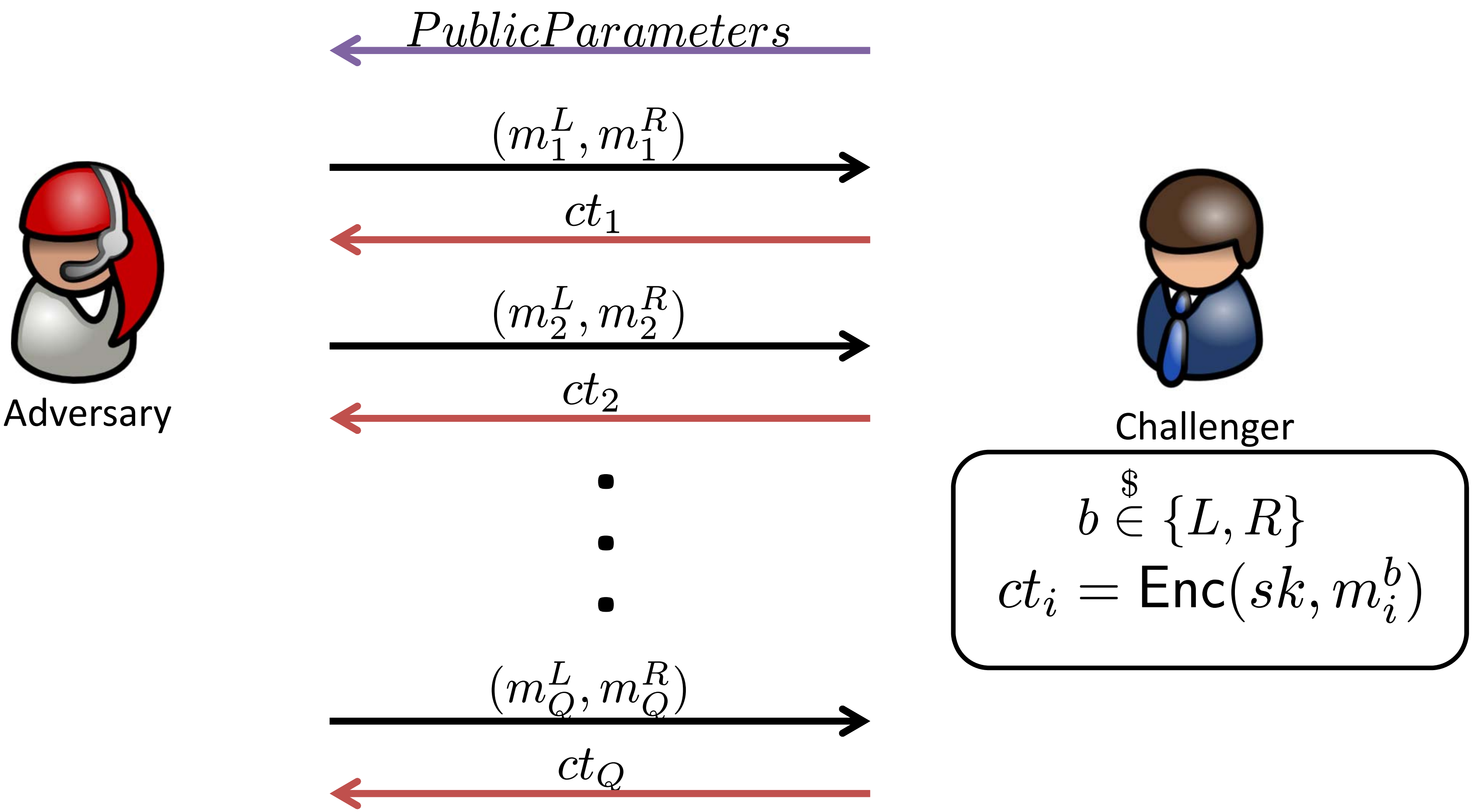
- $\mathsf{Setup} \rightarrow (pp, sk)$
- $\mathsf{Encrypt}(sk, m) \rightarrow ct$
- $\mathsf{Decrypt}(sk, ct) \rightarrow m$
- $\mathsf{Test}(pp, ct_1, ct_2, \ldots, c_k) \rightarrow \{0, 1\}$

Test should satisfy:

$$\mathsf{Test}(pp, ct_1, ct_2, \ldots, ct_k) = P(m_1, m_2, \ldots, m_k)$$

publicly computable $\rightarrow$ symmetric key encryption.

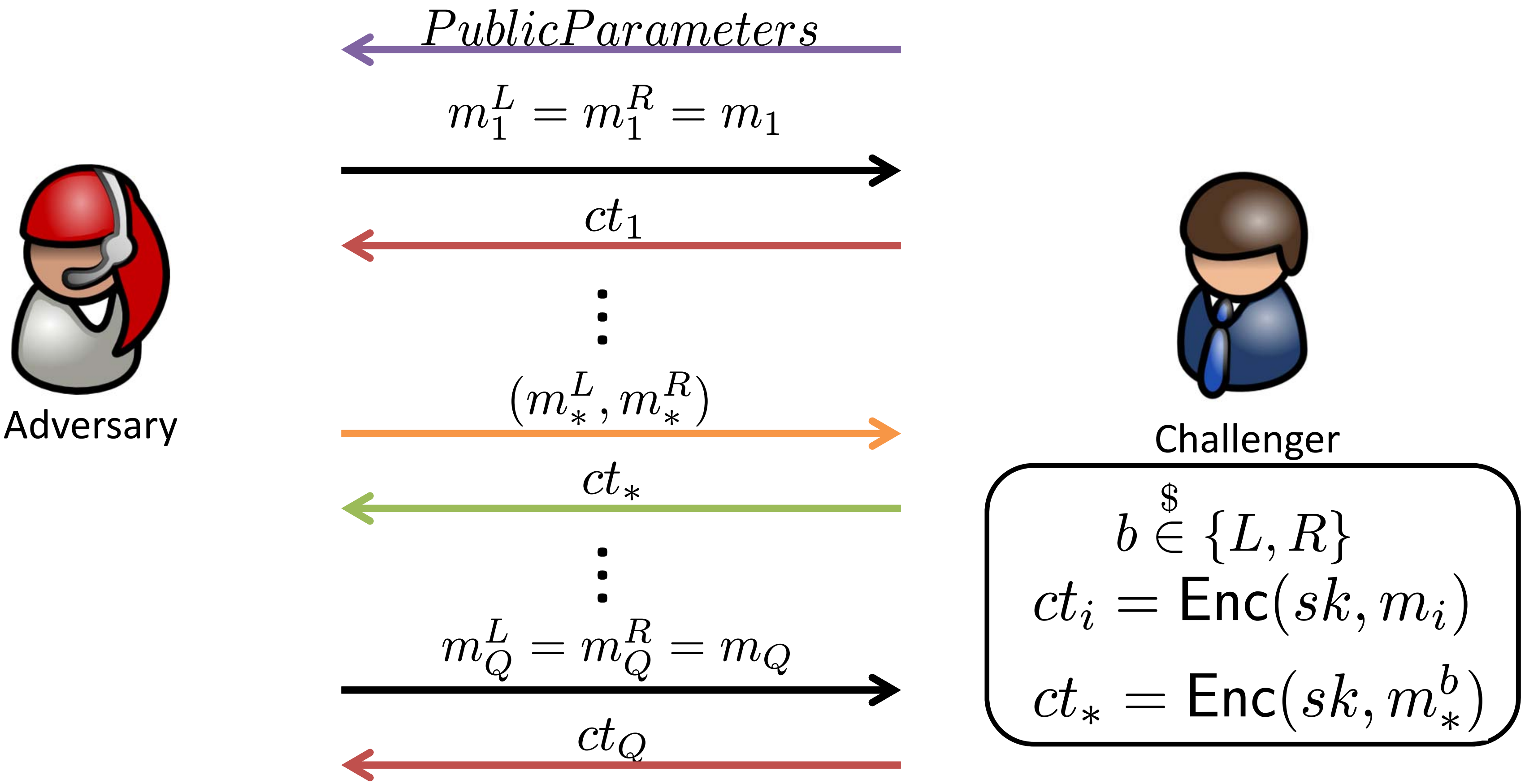
Left or Right Security [BDJR97]



Restriction: For all $(i_1, i_2, \ldots, i_k) \in [Q]^k$:

$$P(m_{i_1}^L, m_{i_2}^L, \ldots, m_{i_k}^L) = P(m_{i_1}^R, m_{i_2}^R, \ldots, m_{i_k}^R)$$

Find Then Guess Security [BDJR97]



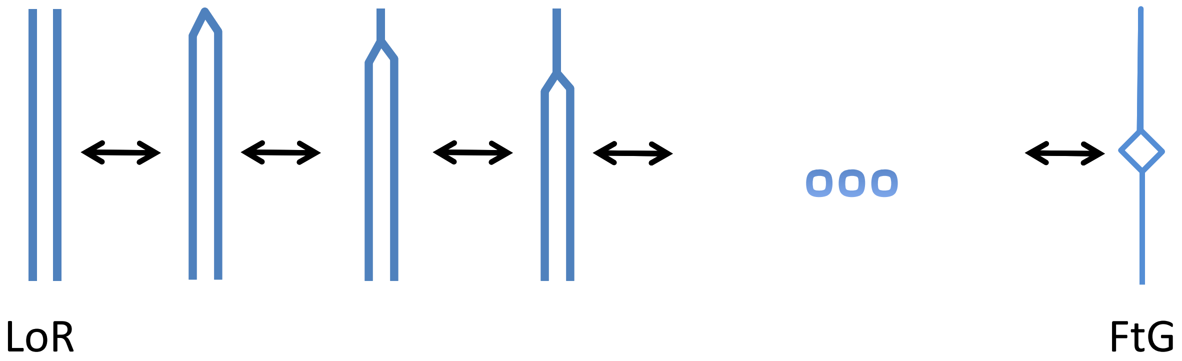
Restriction: For all $(i_1, i_2, \ldots, i_k) \in ([Q] \cup \{*\})^k$:

$$P(m_{i_1}^L, m_{i_2}^L, \ldots, m_{i_k}^L) = P(m_{i_1}^R, m_{i_2}^R, \ldots, m_{i_k}^R)$$

Definitional Relationships

Standard Symmetric Key Cryptography [BDJR97]:

Hybrid Argument



(Symmetric) Property Preserving Encryption:

Not Possible

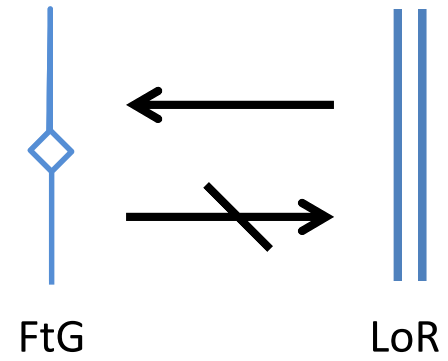
Left Sequence not "reachable" from Right Sequence

- Same equality pattern – Different "reachability" class
- Depends on the property at hand

Definitional Relationships

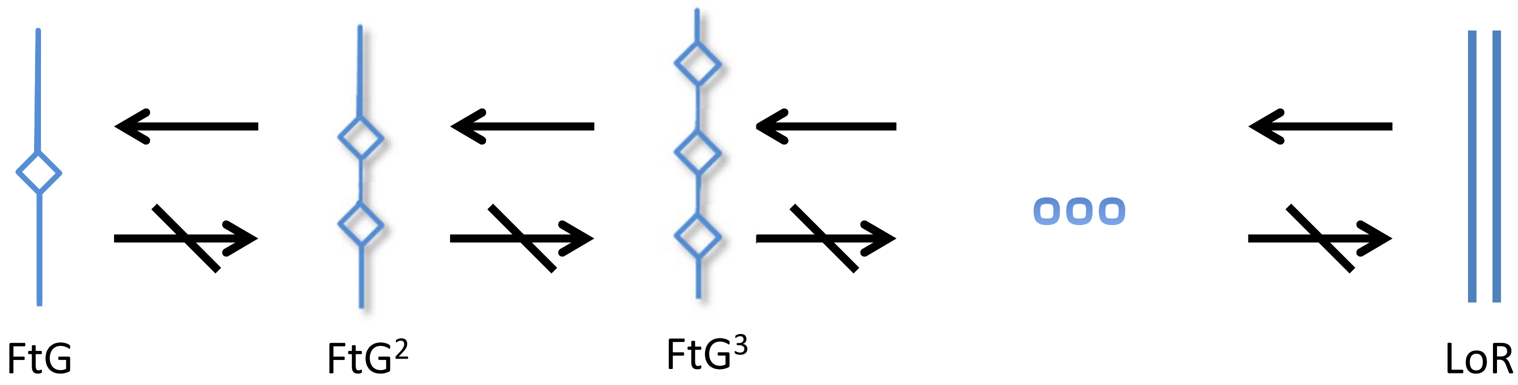
Theorem (informal):

Left or Right security strictly stronger than Find then Guess

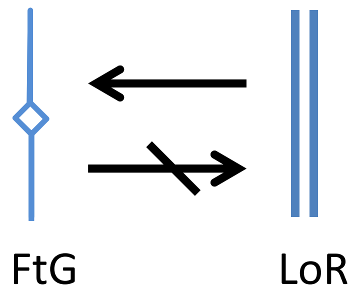


Theorem (informal):

There exists a hierarchy of Find then Guess



Proving the Separation



We will assume that there exists an FtG secure scheme

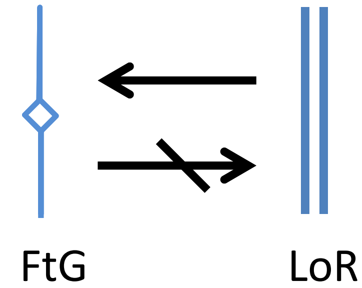
$$\Pi = (\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Test})$$

We will construct a new scheme

$$\Pi^* = (\mathsf{Setup}^*, \mathsf{Encrypt}^*, \mathsf{Decrypt}^*, \mathsf{Test}^*)$$

Such that: Π^* is FtG secure, **but** not LoR secure.

Proving the Separation



Quadratic Residues

Consider $\mathcal{M} = \mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$, where p prime. We have that:

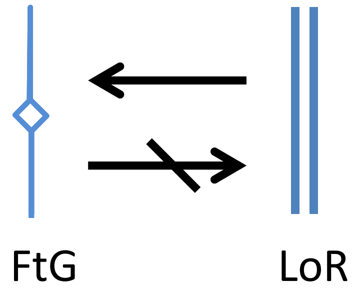
$$\mathcal{QR} = \{x \in \mathbb{Z}_p^* | \exists y \in \mathbb{Z}_p^* : x = y^2\}$$

$$\mathcal{QNR} = \mathbb{Z}_p^* \setminus \mathcal{QR}$$

For $z = xy$, where $x, y, z \in \mathbb{Z}_p^*$,

z is in $\mathcal{QR}$ if and only if Both x and y are in $\mathcal{QR}$ OR Both x and y are in $\mathcal{QNR}$

Proving the Separation



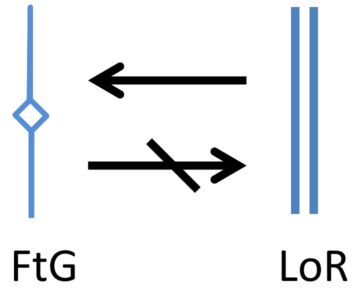
Consider the binary property:

$$P(x, y) = \begin{cases} 1 & \text{if } x \cdot y \in \mathcal{QR} \\ 0 & \text{if } x \cdot y \in \mathcal{QNR} \end{cases}$$

Suppose $\Pi = (\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Test})$ is FtG secure on property P:

$$\mathsf{Test}(\mathsf{Encrypt}(x), \mathsf{Encrypt}(y)) = P(x, y)$$

Proving the Separation



Create a new scheme

$$\Pi^* = (\mathsf{Setup}^*, \mathsf{Encrypt}^*, \mathsf{Decrypt}^*, \mathsf{Test}^*)$$

where:

Setup^*:

Calls $\mathsf{Setup} \rightarrow (pp, sk)$

Samples t from $\{0, 1\}$

Outputs $pp^* = pp$ and $sk^* = (sk, t)$

One-time pad

$\mathsf{Encrypt}^*(sk^*, m)$:

Calls $\mathsf{Encrypt}(sk, m) \rightarrow ct$

Samples b from $\{0, 1\}$

If $b = 0$ outputs

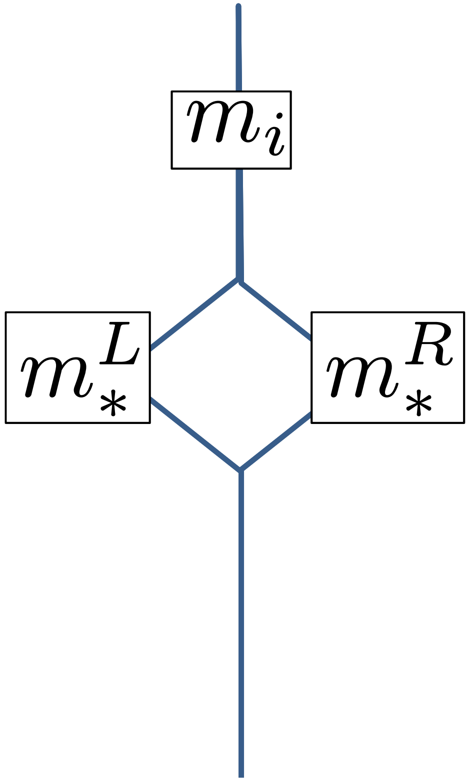
$$ct^* = (ct, b, t)$$

If $b = 1$ outputs

$$ct^* = (ct, b, t \oplus \mathcal{J}(m))$$

$$\mathcal{J}(m) = \begin{cases} 0 & \text{if } m \in \mathcal{QR} \\ 1 & \text{if } m \in \mathcal{QNR} \end{cases}$$

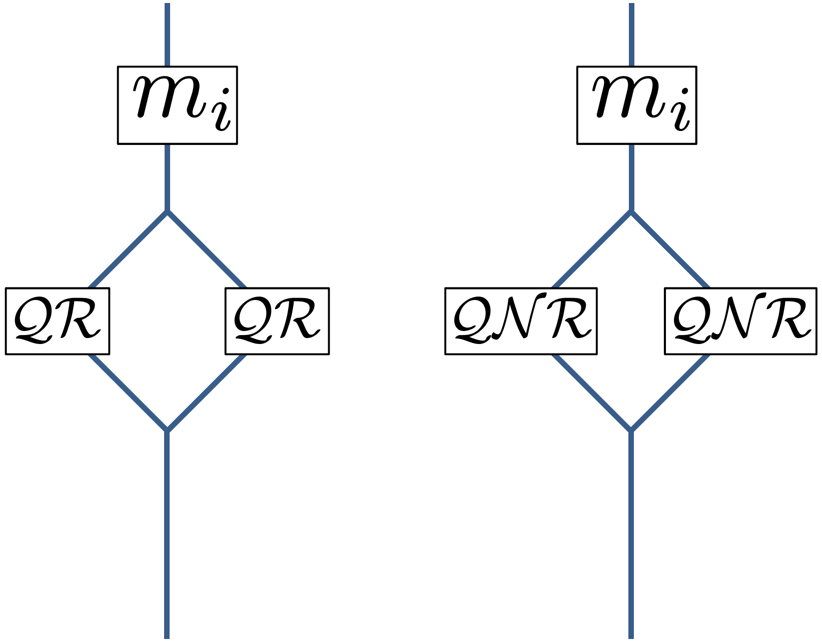
Proving the Separation: Π^* is FtG secure



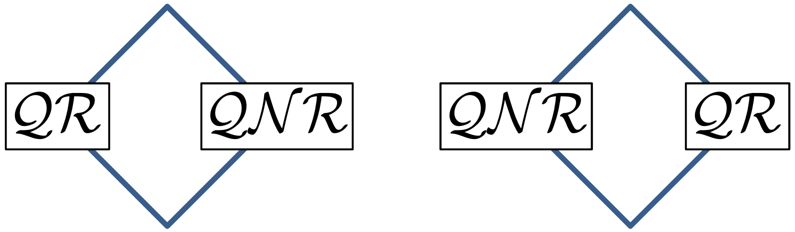
It is true:

$$P(m_i, m_*^L) = P(m_i, m_*^R)$$

Case 1: $\mathcal{J}(m_*^L) = \mathcal{J}(m_*^R)$



Case 2: $\mathcal{J}(m_*^L) \neq \mathcal{J}(m_*^R)$



Proving the Separation: Π^* is FtG secure

Encrypt$^*(sk^*, m)$:
  Encrypt$(sk, m) \rightarrow ct$
  $b \xleftarrow{\$} \{0,1\}$
  If $b = 0$ then $ct^* = (ct, b, t)$
  else $ct^* = (ct, b, t \oplus \mathcal{J}(m))$

Case 1: $\mathcal{J}(m_*^L) = \mathcal{J}(m_*^R)$

Simulator knows t and simulates the game perferctly by answering all single queries and the challenge query.

Case 2: $\mathcal{J}(m_*^L) \neq \mathcal{J}(m_*^R)$

Simulator responds to the *one* query with (ct, b_1, b_2) where b_1, b_2 uniformly random bits.

Proving the Separation: Π^* is not LoR secure

$\mathsf{Encrypt}^*(sk^*, m)$:

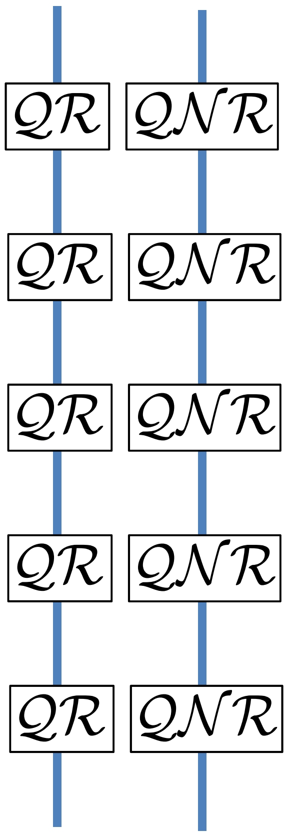
$\mathsf{Encrypt}(sk, m) \rightarrow ct$

$b \xleftarrow{\$} \{0,1\}$

If $b = 0$ then $ct^* = (ct, b, t)$

else $ct^* = (ct, b, t \oplus \mathcal{J}(m))$

Attack:



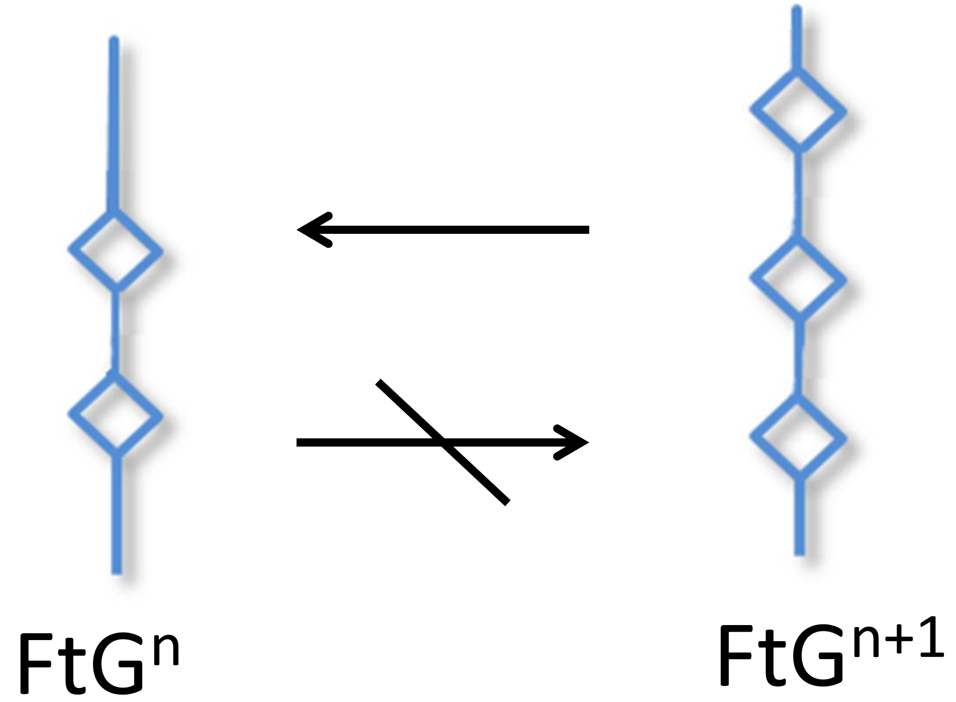
Valid:

$P(m_i^L, m_j^L) = P(m_i^R, m_j^R) = 1$ for all i, j

Successful:

W.h.p. the attacker learns t and can deduce $\mathcal{J}(m)$

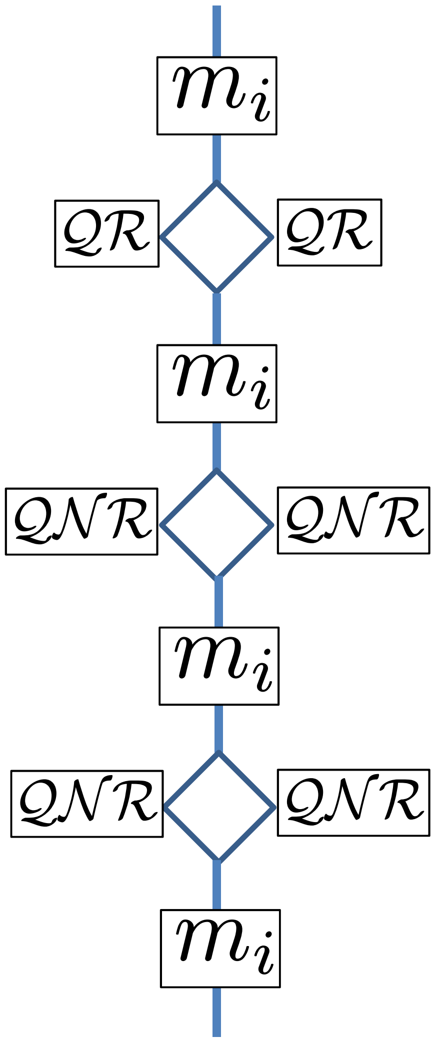
Proving the Hierarchy



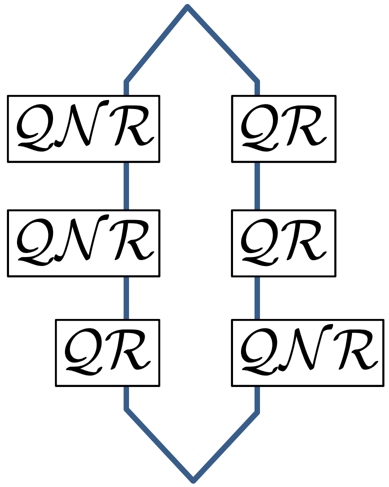
Assuming there exists an FtG^n secure scheme Π, we construct a scheme Π^* that is FtG^n secure, but not FtG^{n+1} secure.

Proving the Hierarchy: Main Ideas

Case 1:



Case 2:



- Use an n-time pad to encode information about sign.
- In case 1 simulate perfectly knowing the pad.
- In case 2 output suitable random integers.
- Correct simulation until n challenge queries.
- Break with constant probability at n+1 challenge queries.

Constructions

- Unary Properties: Trivial generic construction
- Binary Properties using Predicate Encryption [KSW08]:
  - Requires very strong security
  - No candidate construction known for non trivial properties
- Ternary properties and above: Open Problem

Pairings in Composite Order Groups

Let $\mathbb{G}$ be a group of *composite* order $N = p \cdot q$ with a bilinear mapping:

$$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T \text{ and } e(g^a, g^b) = e(g, g)^{ab}$$

Independence Property:
Let g_0, g_1 be generators of the subgroups of order p, q, respectively. Then:

$$e(g_0^a \cdot g_1^b, g_0^c \cdot g_1^d) = e(g_0, g_0)^{ac} \cdot e(g_1, g_1)^{bd}$$

In particular: $e(g_0^a, g_1^b) = 1$

Orthogonality

Property: Orthogonality of n-dimensional vectors in $\mathbb{Z}_p$.

$$\vec{a} = (a_1, a_2, \ldots, a_n) \qquad \vec{b} = (b_1, b_2, \ldots, b_n)$$

$$\vec{a} \cdot \vec{b} = a_1 \cdot b_1 + a_2 \cdot b_2 + \ldots + a_n \cdot b_n$$

$$P(\vec{a}, \vec{b}) = \begin{cases} 0 & \text{if } \vec{a} \cdot \vec{b} = 0 \pmod{p} \\ 1 & \text{otherwise} \end{cases}$$

Explicit Construction: Setup and Encrypt

Secret key: g_0, g_1 and $v, t_1, t_2, \ldots, t_n \in \mathbb{Z}_p$ such that:

$$v^2 = t_1^2 + t_2^2 + \ldots + t_n^2$$

Encryption of $\vec{a} = (a_1, a_2, \ldots, a_n)$:

Pick $r, s \in \mathbb{Z}_p$ and output

$$g_1^{rv},\ (g_0^{sa_1} \cdot g_1^{rt_1}, g_0^{sa_2} \cdot g_1^{rt_2}, \ldots, g_0^{sa_n} \cdot g_1^{rt_n})$$

Explicit Construction: Test

$$Enc(\vec{a}) : \quad g_1^{rv}, g_0^{sa_1} g_1^{rt_1}, g_0^{sa_2} g_1^{rt_2}, \ldots, g_0^{sa_n} g_1^{rt_n}$$

$$Enc(\vec{b}) : \quad g_1^{r'v}, g_0^{s'b_1} g_1^{r't_1}, g_0^{s'b_2} g_1^{r't_2}, \ldots, g_0^{s'b_n} g_1^{r't_n}$$

First pairing: $e(g_1, g_1)^{rr'v^2}$

Product of n pairings:

$$e(g_0, g_0)^{ss'a_1b_1} e(g_1, g_1)^{rr't_1^2} \cdot \ldots \cdot$$

$$e(g_0, g_0)^{ss'a_nb_n} e(g_1, g_1)^{rr't_n^2}$$

$$= e(g_0, g_0)^{ss' \cdot (\vec{a} \cdot \vec{b})} e(g_1, g_1)^{rr'(t_1^2 + \ldots + t_n^2)}$$

New Directions

- New interesting properties:
  - Ternary properties and above.
  - Arithmetic progressions.
  - Geometric shapes - Straight Lines.
  - General properties.
- Using lattices, since pairings seem suitable only for binary properties.
- "Privatizing" popular algorithms:
  - Clustering
  - Data classification
- Generalizing the properties to functions

  → Powerful public computations on encrypted data.

Thank you

Questions ?