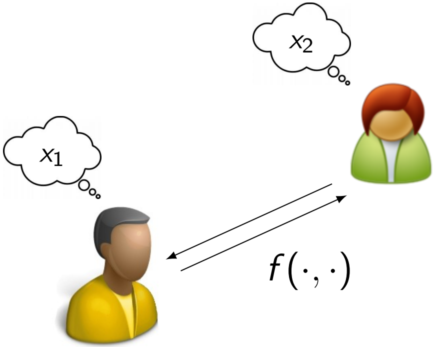
Encryption Switching Protocols

Geoffroy Couteau, Thomas Peters, and David Pointcheval

École Normale Supérieure, CNRS, INRIA, PSL

University of Aarhus
Thursday, June 3

Two-Party Computation



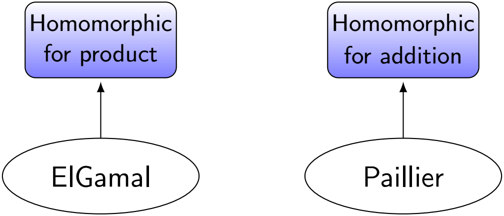
- *Correctness:* the output is $f(x_1, x_2)$
- *Privacy:* player i learns nothing on x_{2-i} (except $f(x_1, x_2)$)

Two-Party Computation from Homomorphic Encryption

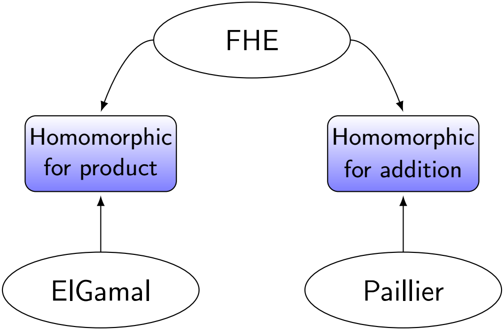
Homomorphic for product

Homomorphic for addition

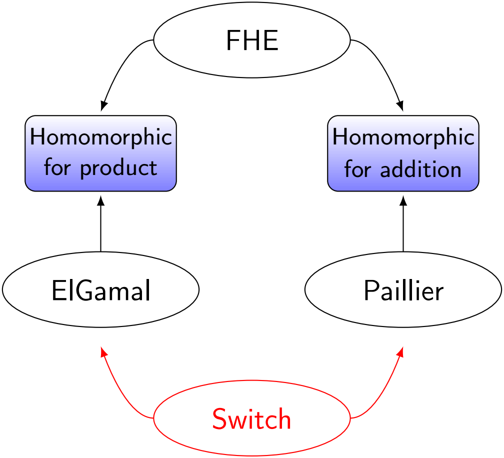
Two-Party Computation from Homomorphic Encryption



Two-Party Computation from Homomorphic Encryption



Two-Party Computation from Homomorphic Encryption

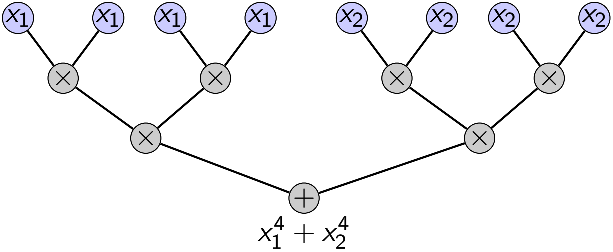


A Theoretical Example

Consider $f_{t,d} : (x_1, \cdots, x_t) \mapsto \sum_{i=1}^{t} x_i^d$

A Theoretical Example

Consider $f_{t,d} : (x_1, \cdots, x_t) \mapsto \sum_{i=1}^{t} x_i^d$



- Baur and Strassen (1983): *Any circuit computing $f_{t,d}$ has a size lower-bounded by $\Omega(t \log(d))$.*
- Most 2-PC protocols securely evaluating $f_{t,d}$ have a communication of $\Omega(t \log(d) poly(\kappa))$. (except FHE)

A Theoretical Example

Suppose we have:

- An additive scheme and a multiplicative scheme
- An ESP between them

How to evaluate $f_{t,d}$ with $O(t)$ communication?

A Theoretical Example

Suppose we have:

- An additive scheme and a multiplicative scheme
- An ESP between them

How to evaluate $f_{t,d}$ with $O(t)$ communication?

x_0 x_1 x_2 x_3 x_4

A Theoretical Example

Suppose we have:

- An additive scheme and a multiplicative scheme
- An ESP between them

How to evaluate $f_{t,d}$ with $O(t)$ communication?

x_0 x_1 x_2 x_3 x_4

A Theoretical Example

Suppose we have:

- An additive scheme and a multiplicative scheme
- An ESP between them

How to evaluate $f_{t,d}$ with $O(t)$ communication?

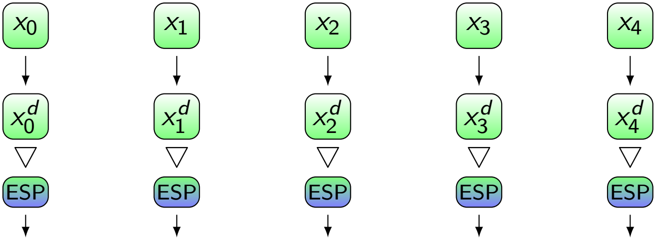
$$\begin{tikzcd}
x_0 \arrow[d] & x_1 \arrow[d] & x_2 \arrow[d] & x_3 \arrow[d] & x_4 \arrow[d] \\
x_0^d & x_1^d & x_2^d & x_3^d & x_4^d
\end{tikzcd}$$

A Theoretical Example

Suppose we have:

- An additive scheme and a multiplicative scheme
- An ESP between them

How to evaluate $f_{t,d}$ with $O(t)$ communication?

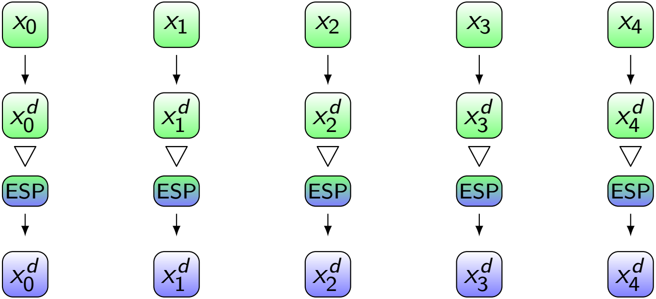


A Theoretical Example

Suppose we have:

- An additive scheme and a multiplicative scheme
- An ESP between them

How to evaluate $f_{t,d}$ with $O(t)$ communication?

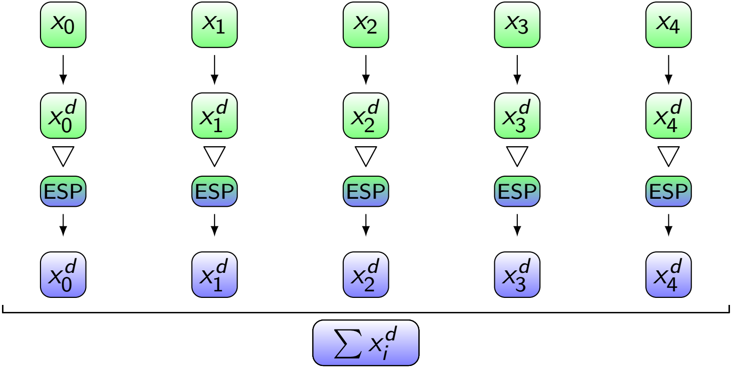


A Theoretical Example

Suppose we have:

- An additive scheme and a multiplicative scheme
- An ESP between them

How to evaluate $f_{t,d}$ with $O(t)$ communication?



Homomorphic Cryptosystems

ElGamal Cryptosystem

- Semantic security: DDH assumption
- Homomorphic for $\times$

Paillier Cryptosystem

- Semantic security: DCR assumption
- Homomorphic for $+$

DDH assumption over $\mathbb{G}$:
Given $(g, g^a, g^b, g^c) \in \mathbb{G}^4$, find out whether $c = ab$.

DCR assumption for $n = pq$, with (p, q) safe primes:
Given $x \in \mathbb{Z}_{n^2}$ find out whether it is a nth power.

Homomorphic Cryptosystems

ElGamal Cryptosystem

- Semantic security: DDH assumption
- Homomorphic for $\times$
- Encrypts over any suitable $\mathbb{G}$

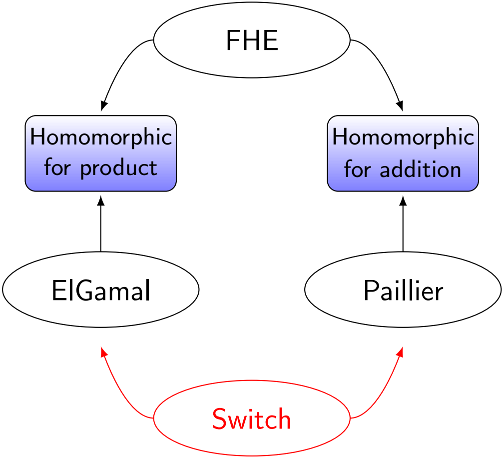
Paillier Cryptosystem

- Semantic security: DCR assumption
- Homomorphic for $+$
- Encrypts over $\mathbb{Z}_n$

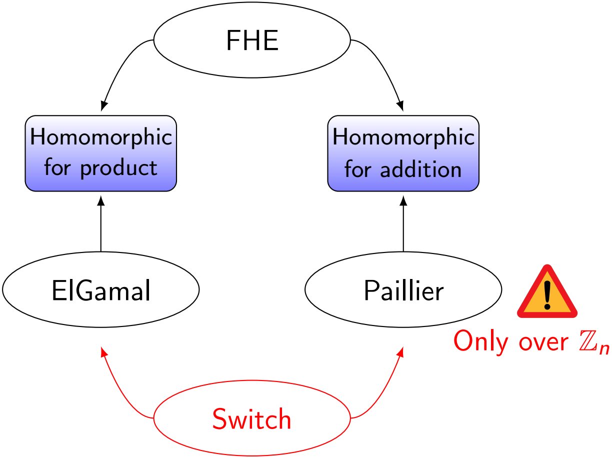
DDH assumption over $\mathbb{G}$:
Given $(g, g^a, g^b, g^c) \in \mathbb{G}^4$, find out whether $c = ab$.

DCR assumption for $n = pq$, with (p, q) safe primes:
Given $x \in \mathbb{Z}_{n^2}$ find out whether it is a nth power.

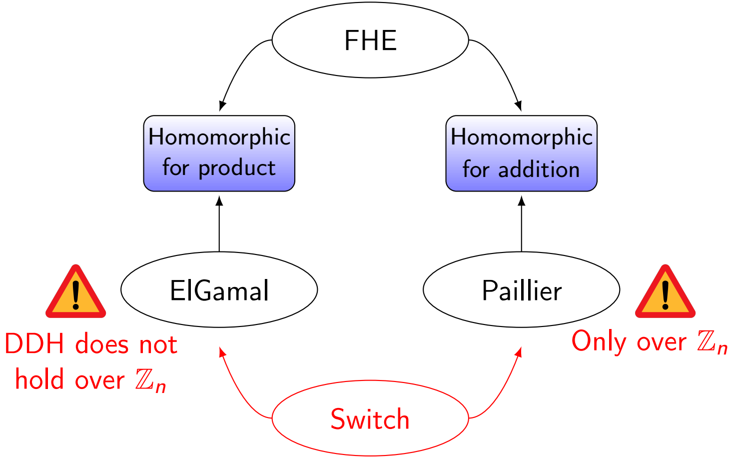
Multiparty Computation from Homomorphic Encryption



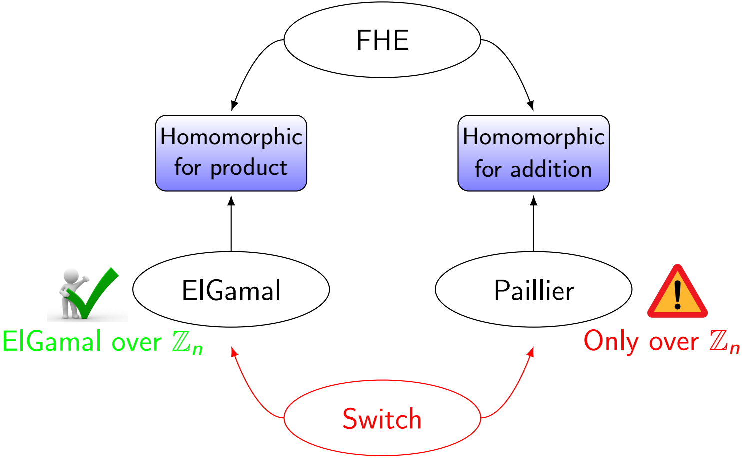
Multiparty Computation from Homomorphic Encryption



Multiparty Computation from Homomorphic Encryption

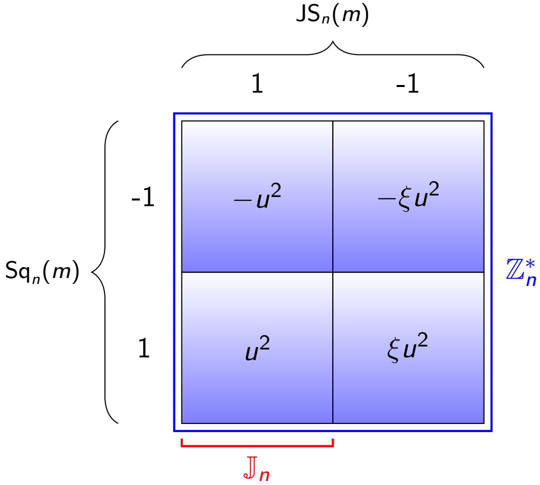


Multiparty Computation from Homomorphic Encryption



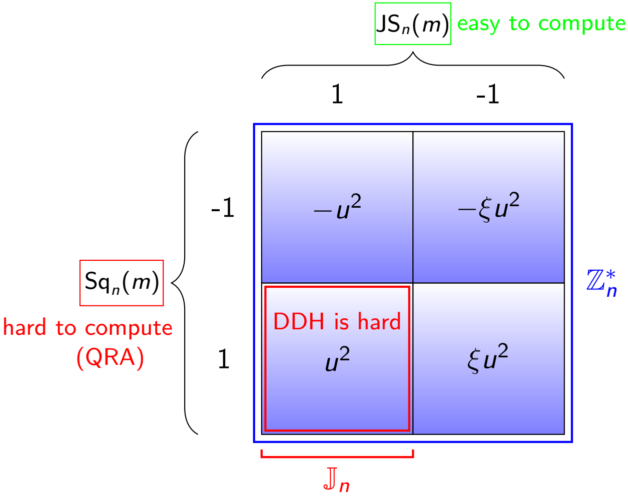
Structure of $(\mathbb{Z}_n^*, \times)$

- $n = p \cdot q, (p, q)$ are safe primes.
- 1 has four square roots: $(1, -1, \xi, -\xi)$.



Structure of $(\mathbb{Z}_n^*, \times)$

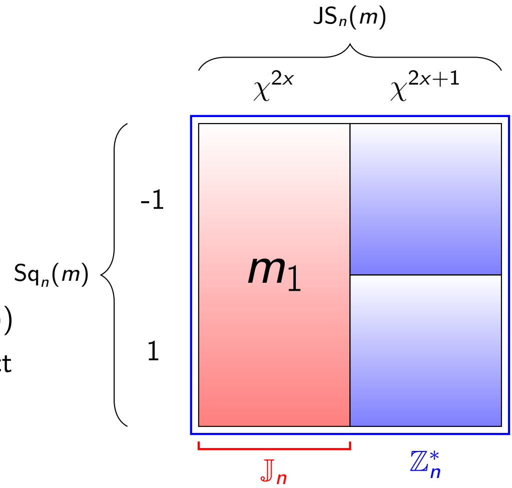
- $n = p \cdot q, (p, q)$ are safe primes.
- 1 has four square roots: $(1, -1, \xi, -\xi)$.



An ElGamal Variant over $\mathbb{Z}_n^*$

How to encrypt $m \in \mathbb{Z}_n^*$?

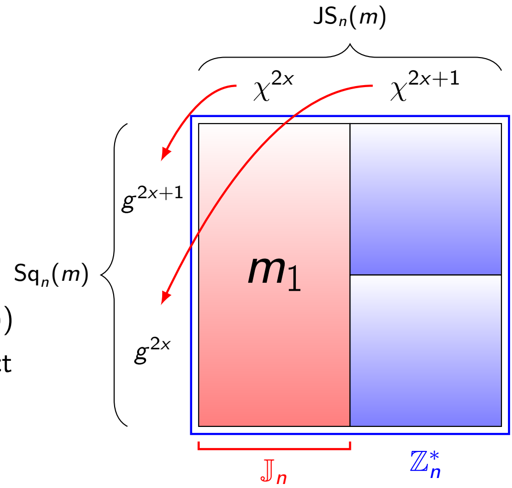
- $\chi \in \mathbb{Z}_n \setminus \mathbb{J}_n$
- g is a generator of $\mathbb{J}_n$
- $m = \chi^a \cdot m_1$
- $\mathsf{Enc}(m) = (g^a, \mathsf{EG}_{\mathbb{J}_n}(m_1))$
- Homomorphic for product



An ElGamal Variant over $\mathbb{Z}_n^*$

How to encrypt $m \in \mathbb{Z}_n^*$?

- $\chi \in \mathbb{Z}_n \setminus \mathbb{J}_n$
- g is a generator of $\mathbb{J}_n$
- $m = \chi^a \cdot m_1$
- $\mathsf{Enc}(m) = (g^a, \mathsf{EG}_{\mathbb{J}_n}(m_1))$
- Homomorphic for product



An ElGamal Variant over $\mathbb{Z}_n^* \simeq \mathbb{Z}_p^* \times \mathbb{Z}_q^*$

How to decrypt $\mathrm{Enc}(m) = (g^a, \mathrm{EG}_{\mathbb{J}_n}(m_1))$, with $m = \chi^a m_1$?

- Use the **chinese remainder theorem**
- Add discrete logs of χ in base g mod p and q to the secret key

	1	-1
-1	non-square mod p and non-square mod q	square mod p and non-square mod q
1	square mod p and square mod q	non-square mod p and square mod q

$\mathbb{Z}_n^*$

Extending the Variant over $\mathbb{Z}_n^* \cup \{0\}$

- Encoding $m \in \mathbb{Z}_n^* \cup \{0\}$ over $\mathbb{Z}_n^*$
- Preserving the homomorphic properties

0 is *absorbant* over $\mathbb{Z}_n^* \cup \{0\}$

$0 \times m = 0$

random is *absorbant* over $\mathbb{Z}_n^*$

random $\times m =$ random

Let $b = 1$ if $m = 0$, $b = 0$ else.

$$\text{Encoding}(m) = (m + rb, R^b)$$

Putting Pieces Together

- We have an ElGamal-like scheme over $\mathbb{Z}_n^* \cup \{0\}$
- $\mathbb{Z}_n^* \cup \{0\}$ is “equivalent” to $\mathbb{Z}_n$ if the factorization is unknown
- We can use threshold schemes to ensure it

A toy scheme which does not handle the zero:

m

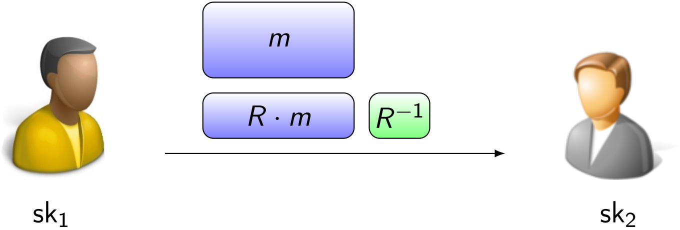
sk_1

sk_2

Putting Pieces Together

- We have an ElGamal-like scheme over $\mathbb{Z}_n^* \cup \{0\}$
- $\mathbb{Z}_n^* \cup \{0\}$ is "equivalent" to $\mathbb{Z}_n$ if the factorization is unknown
- We can use threshold schemes to ensure it

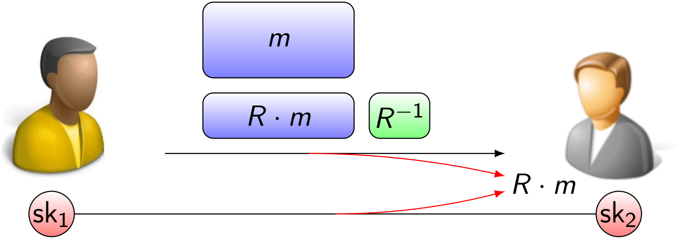
A toy scheme which does not handle the zero:



Putting Pieces Together

- We have an ElGamal-like scheme over $\mathbb{Z}_n^* \cup \{0\}$
- $\mathbb{Z}_n^* \cup \{0\}$ is "equivalent" to $\mathbb{Z}_n$ if the factorization is unknown
- We can use threshold schemes to ensure it

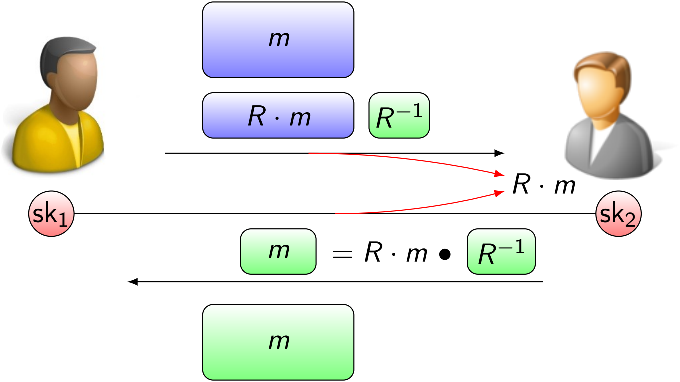
A toy scheme which does not handle the zero:



Putting Pieces Together

- We have an ElGamal-like scheme over $\mathbb{Z}_n^* \cup \{0\}$
- $\mathbb{Z}_n^* \cup \{0\}$ is "equivalent" to $\mathbb{Z}_n$ if the factorization is unknown
- We can use threshold schemes to ensure it

A toy scheme which does not handle the zero:



What Do We Do Next?

- Deal with the other direction
- Extend the construction to handle zeros
- Prove formally that it implies general 2-PC
- Add security against malicious adversaries

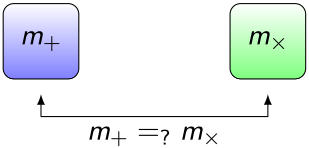
What Do We Do Next?

- Deal with the other direction
- Extend the construction to handle zeros
- Prove formally that it implies general 2-PC
- Add security against malicious adversaries

Requires new techniques for ZK

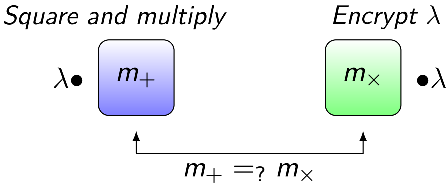
Twin Ciphertext Proof

- The core of the problem is a non-algebraic statement.



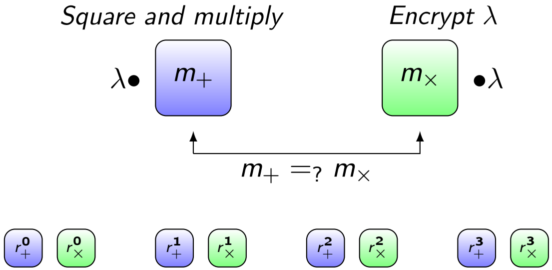
Twin Ciphertext Proof

- The core of the problem is a non-algebraic statement.
- However, there is some common algebraic structure.



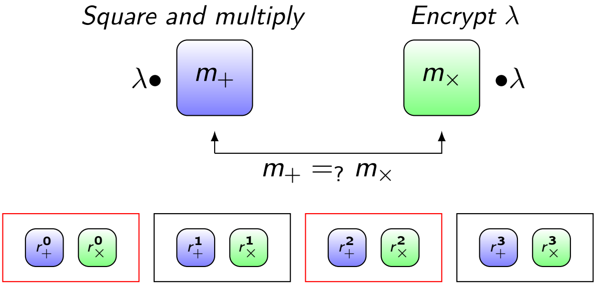
Twin Ciphertext Proof

- The core of the problem is a non-algebraic statement.
- However, there is some common algebraic structure.



Twin Ciphertext Proof

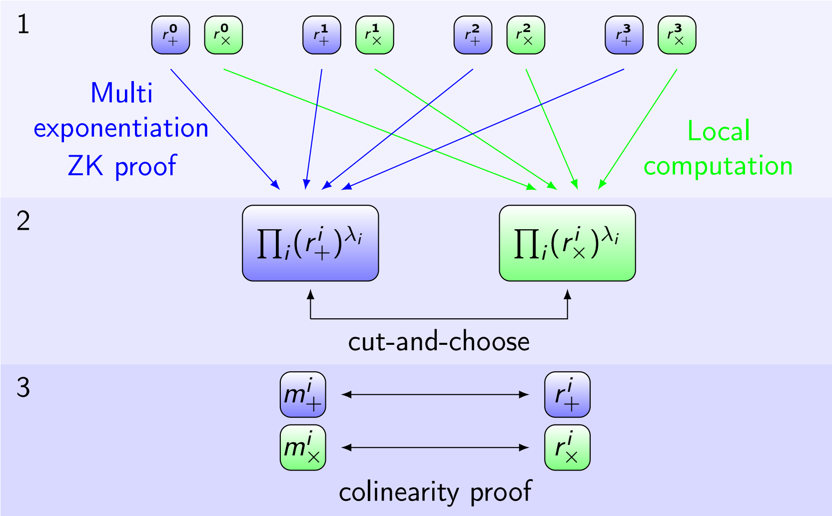
- The core of the problem is a non-algebraic statement.
- However, there is some common algebraic structure.



- Reveal the random coins
- Reveal m/r_i and play two plaintext-equality proofs

$\hookrightarrow$ Colinearity proof

Pool of Twin-Ciphertext Pairs



Applications

Given access to a pool of preproven twin-ciphertext pairs, the players can very efficiently perform various ZK proofs:

- Double-logarithms proofs
- Proofs of exponential relations (known or unknown exponents)
- Proofs that a committed number is a prime
- And so on...

Thank you for your attention