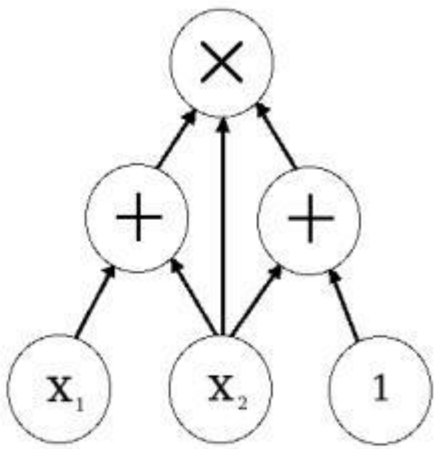
# Arithmetic Cryptography
# or
# what Garbled Circuits CAN'T do



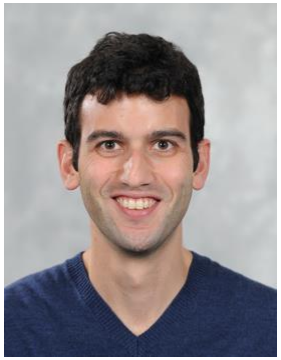

Benny Applebaum, Jonathan Avron, Christina Brzuska

Tel Aviv University

# Motivating Example

**FHE Factory**

**Clients**



Option 1:
Construct three different FHEs

FHE that supports operations overs **finite-precision reals**

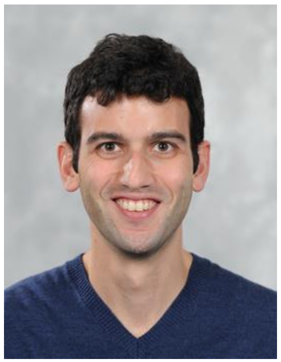
FHE that supports **mod-N** operations

FHE that supports operations over **some field or a ring F**

Too much work…

# Motivating Example

**FHE Factory**



Option 2:
Simulate computation via Boolean circuit

**Clients**

FHE that supports operations overs **finite-precision reals**

FHE that supports **mod-N** operations

FHE that supports operations over **some field or a ring F**

But Boolean simulation may be

- Expensive

cost may be much larger than log |**F**|

- Not Modular

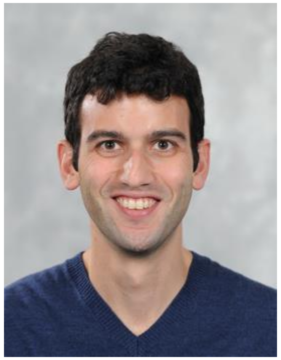
sensitive to the bit-representation of field elements

- Infeasible

if there's no access to the bit-wise representation of field elements

# Motivating Example

**FHE Factory**



Option 3:

Arithmetic FHE ?

**Clients**

FHE that supports operations overs **finite-precision reals**

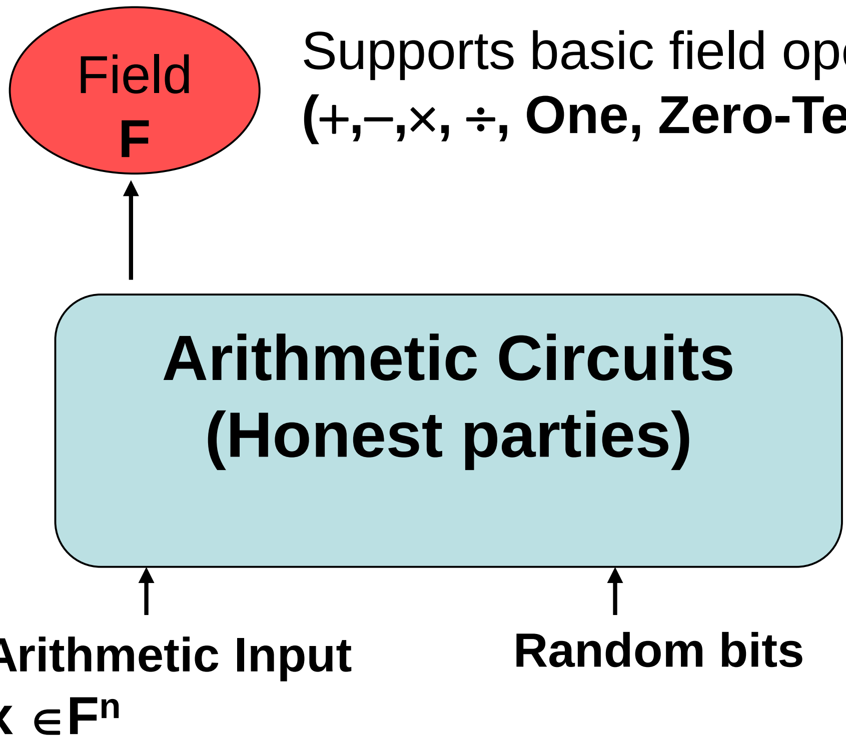
FHE that supports **mod-N** operations

FHE that supports operations over **some field or a ring F**

Ideally:

- Design general scheme with oracle to a field/ring **F**
- Can be later instantiated with any concrete field

# Arithmetic Cryptography

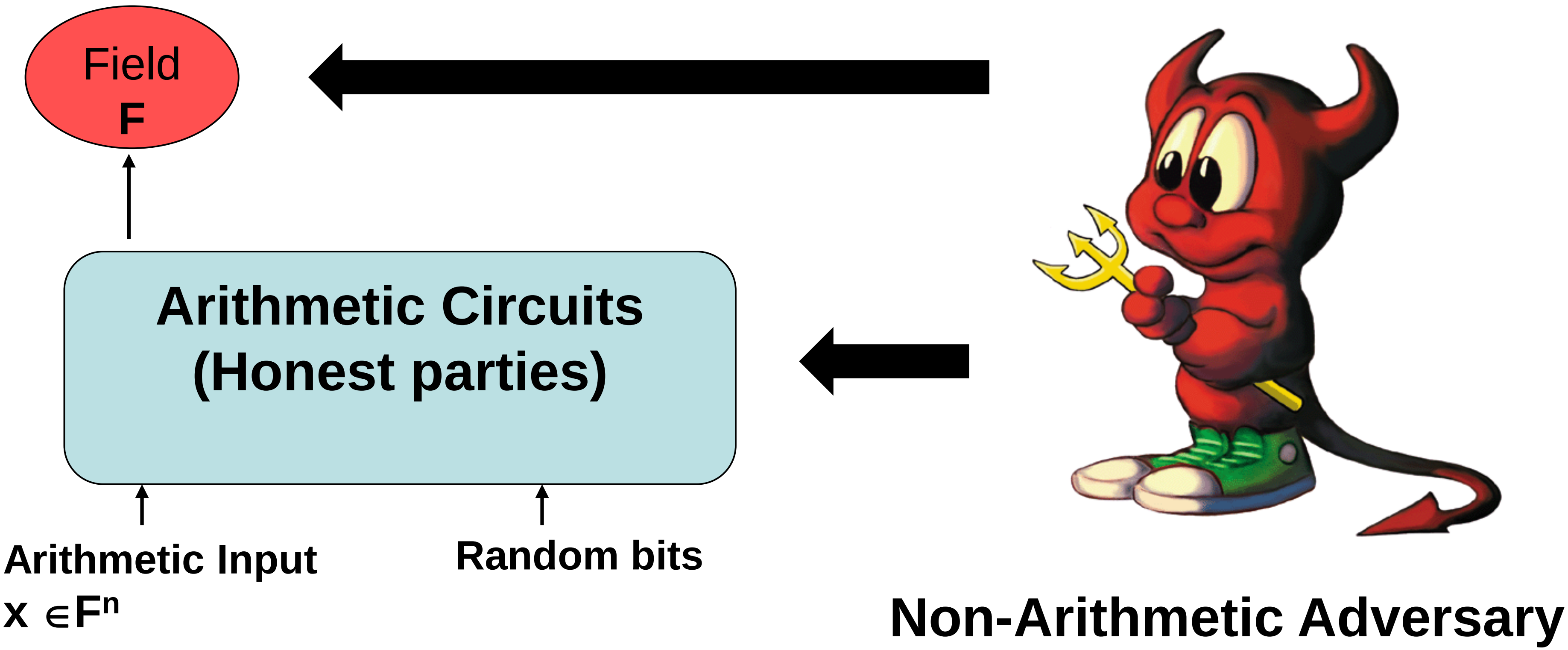



Supports basic field operations
**(+,–,×, ÷, One, Zero-Testing, Get-Random-F)**

**Expressive power:**

- Can solve linear equations
- Cannot sample a Gaussian over F
- Cannot get the i-th bit of **x**

# Arithmetic Cryptography




**Which primitives can be implemented in this model?**

# Previous Works

- Information-theoretic primitives
  - one-time pad, one-time MACs
  - Secret-sharing over fields [Sha79] rings [DF94,CF02]
  - MPC over fields [BGW88,CCD88]
  - Randomized encoding: fields [IK00], rings [CFIK03]

# Previous Works

- So far, no computational primitives in this model

- Some results in weaker models
  - Given (arbitrary) bit-representation of **F**'s elements: secure 2-party computation [NP99, IPS09]

  - Given a special encryption scheme over **F** arithmetic garble circuits [AIK11]

  - Given threshold Add-Hom-Enc over **F**: secure multiparty computation [FH96,CDN01,CDN03]

# Our Results

## Positive*

- Commitments
- Symmetric Encryption
- Public-key Encryption
- Arithmetic OT

$\Rightarrow$ Secure 2-PC (using [IPS])

*Assume pseudorandomness of noisy random linear code over **F** (generalization of LPN)

- Arithmetic model is **non-trivial**
  – The model allows Computational Crypto

# Our Results

## Positive*

- Commitments
- Symmetric Encryption
- Public-key Encryption
- Arithmetic OT

⇒ Secure 2-PC (using [IPS])

## Negative

- Additive-Homomorphic-Enc
- Arithmetic Garbled Circuit
- Secure computation with “low” online complexity

- **Separation**: Arithmetic model ≠ Boolean model
- **Intuition**: Easier to “analyze” arithmetic circuits
  - E.g., can check if f=g (polynomial identity testing)
  - Algorithms for AC’s ⇒ Attacks on Arithmetic Crypto

# What does this mean?

**Arithmetization Barrier**: If your construction "arithmetize" then face the lower-bounds

**Example 1**:

Explains the limitations of LPN-based primitives

as LPN-based constructions typically arithmetize
(e.g., hard to base FHE on LPN see also [Br13])

# What does this mean?

**Arithmetization Barrier**: If your construction "arithmetize" then face the lower-bounds

**Example 2**:

Explains why the gadget needed for [AIK11] does not have an arithmetic implementation

Also explains the communication complexity of [CFIK00, IPS09]

# What does this mean?

**Arithmetization Barrier**: If your construction "arithmetize" then face the lower-bounds

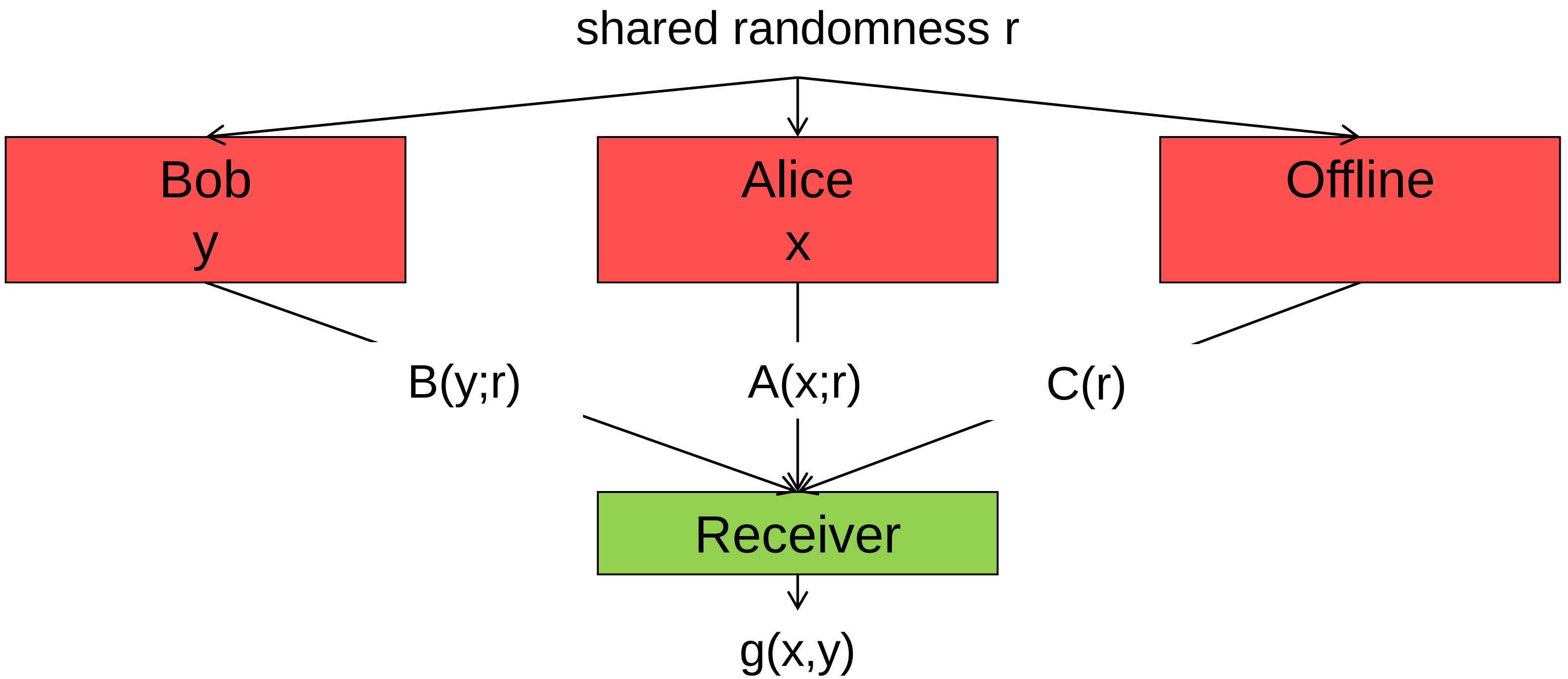
**Example 3**:

Most information-theoretic MPC's arithmetize so they cannot achieve low online complexity

# Proving Lower Bounds

# Private Simultaneous Messages [FKN]

**Privacy**: Receiver learns g(x,y) and nothing else.
**Goal**: Minimize the communication of Alice and Bob.



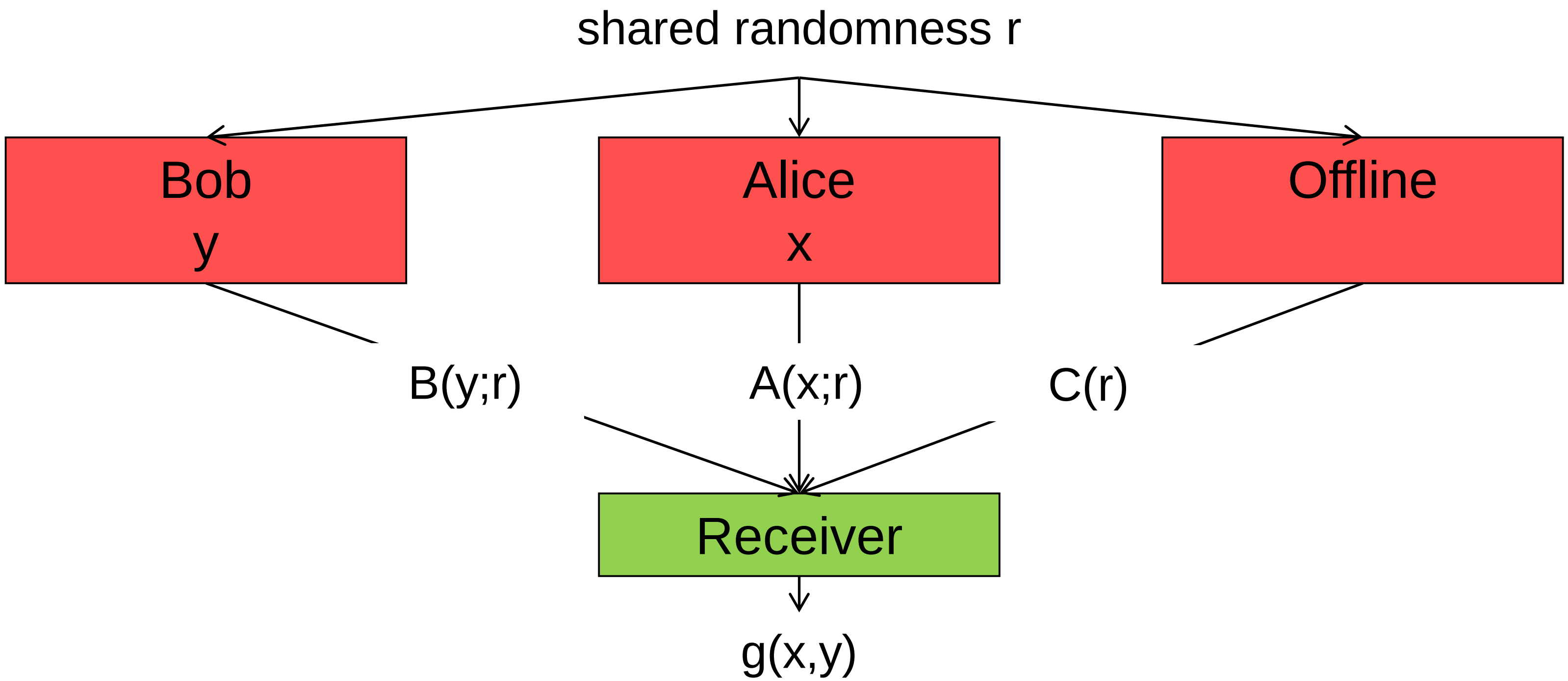
# Private Simultaneous Messages [FKN]

**Boolean case**: Alice's communication ind. of Bob's input and g's complexity.

- |A(x)|=|x|*security-parameter or even |x|+security-parameter [AIKW13]

**Thm:** in the **Arithmetic case** |A(x)|≥|y|

- We will later show that |A(x)| grows with g's complexity
- Both claims generalize to standard MPC setting



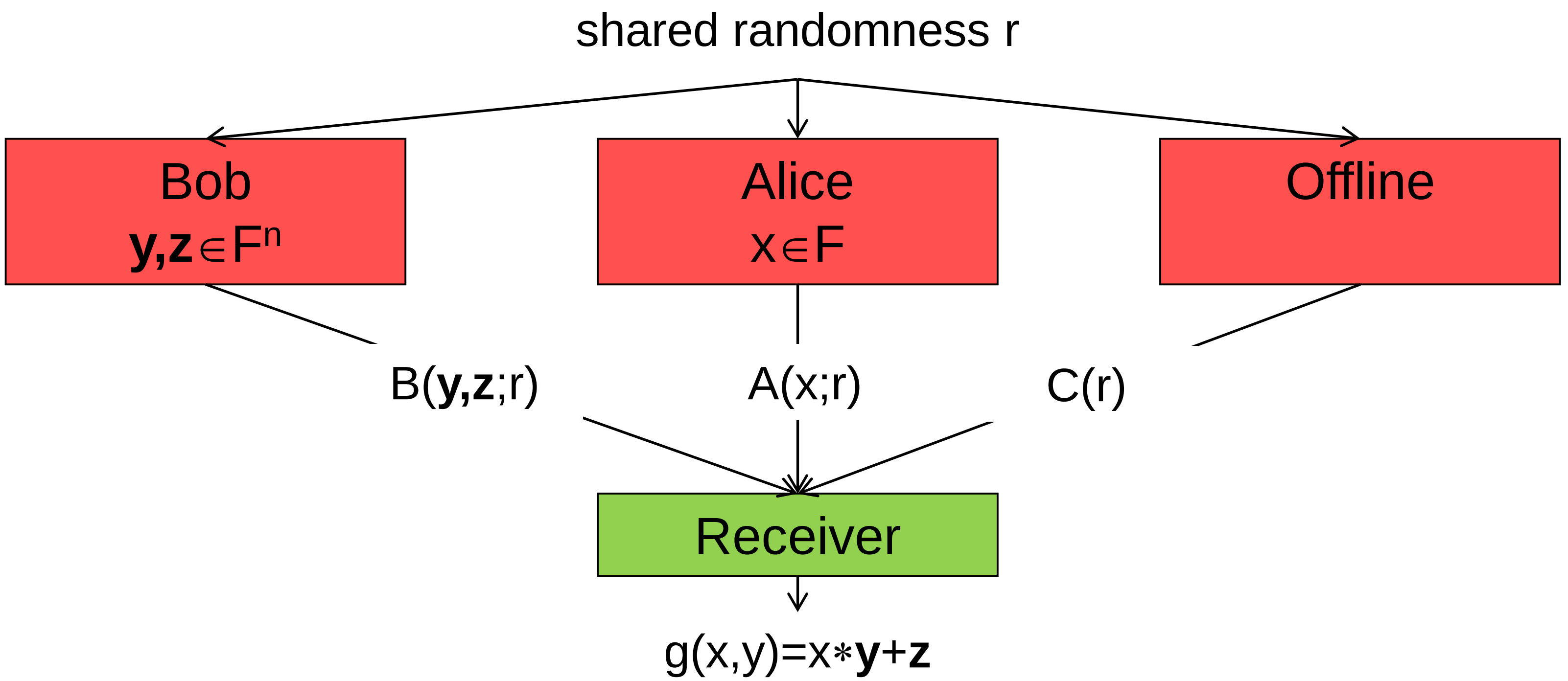
# Lower Bound for Affine Functions

**Goal**: Assuming |A(x;r)|<n, the Receiver learns information about **y**.

- The receiver will output **y*** such that **y*≠y**.

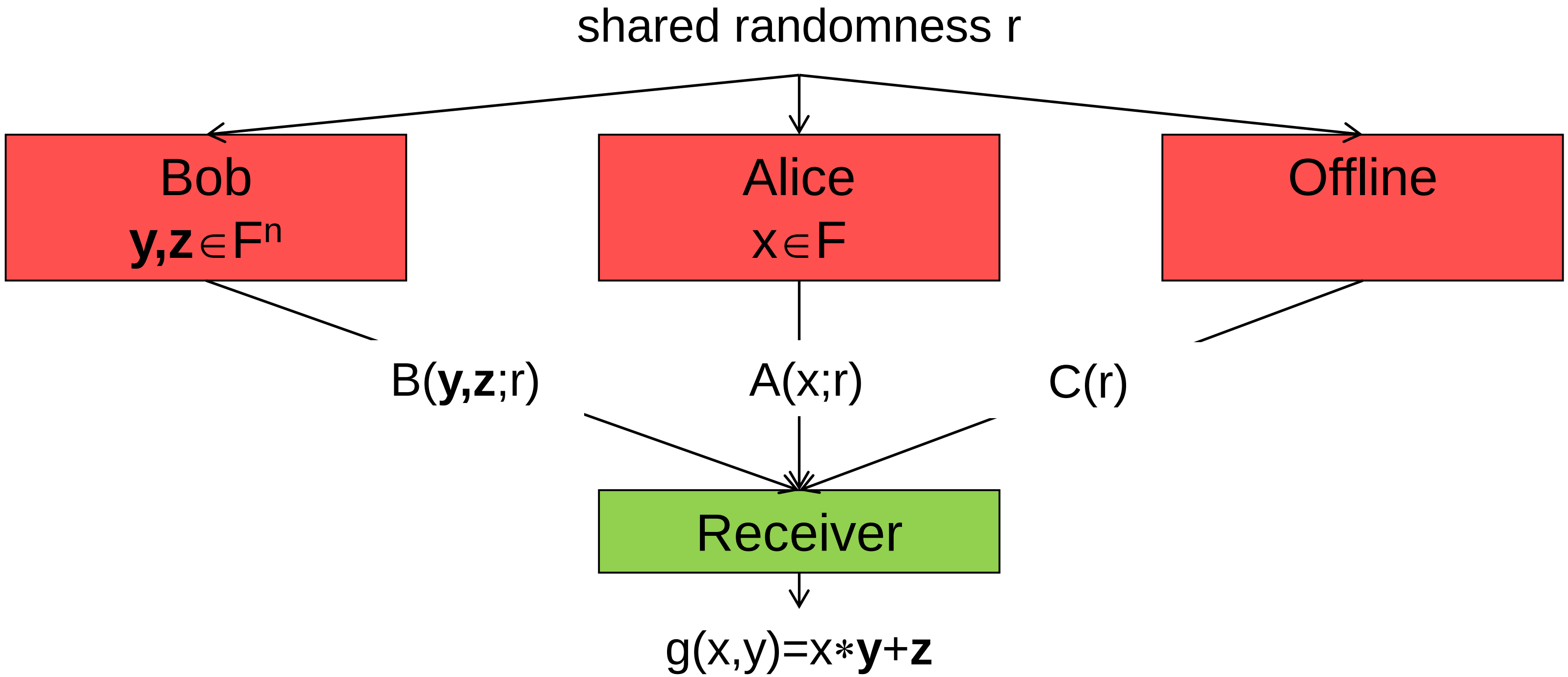
**Simplification**: For now we disallow division gates and zero-testing

- So all parties are polynomials over F



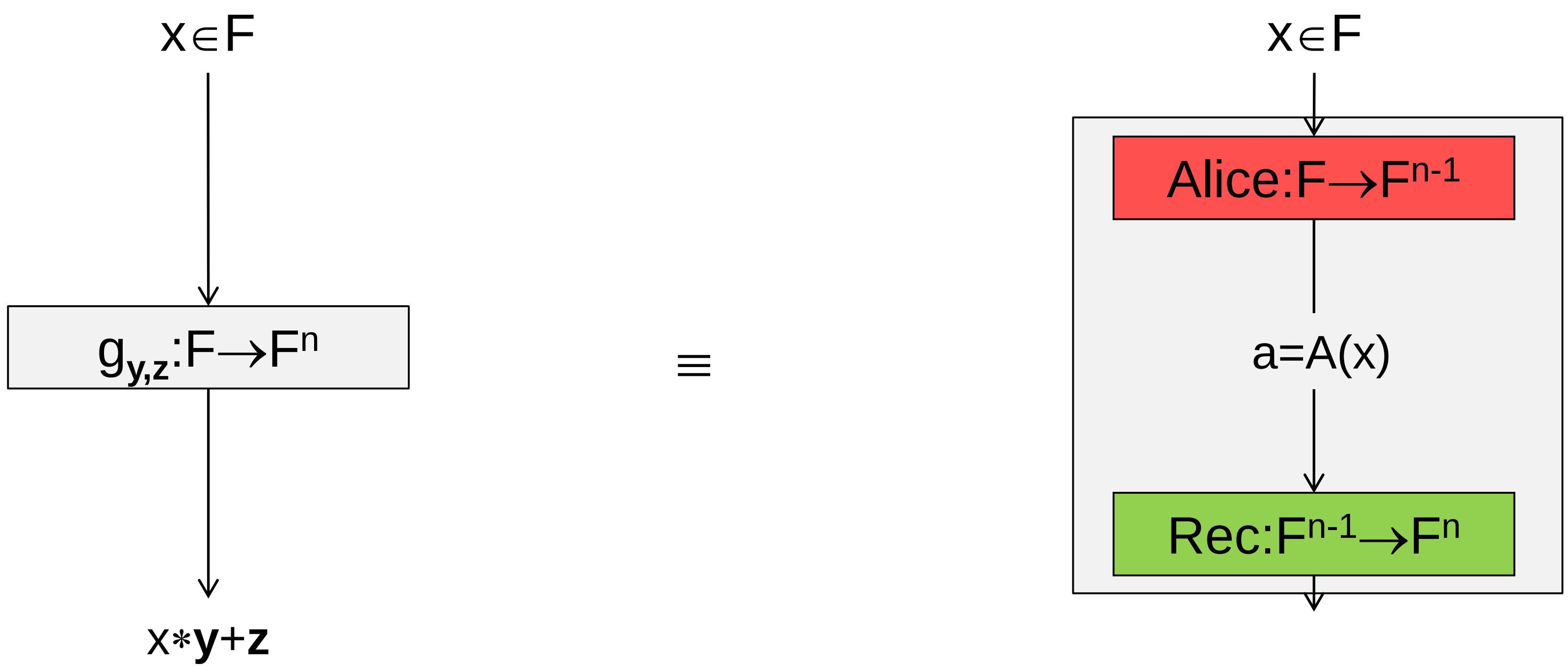
# Observations

Fix r,y,z, C(r).
Consider Alice’s polynomial and the Receiver's polynomial.



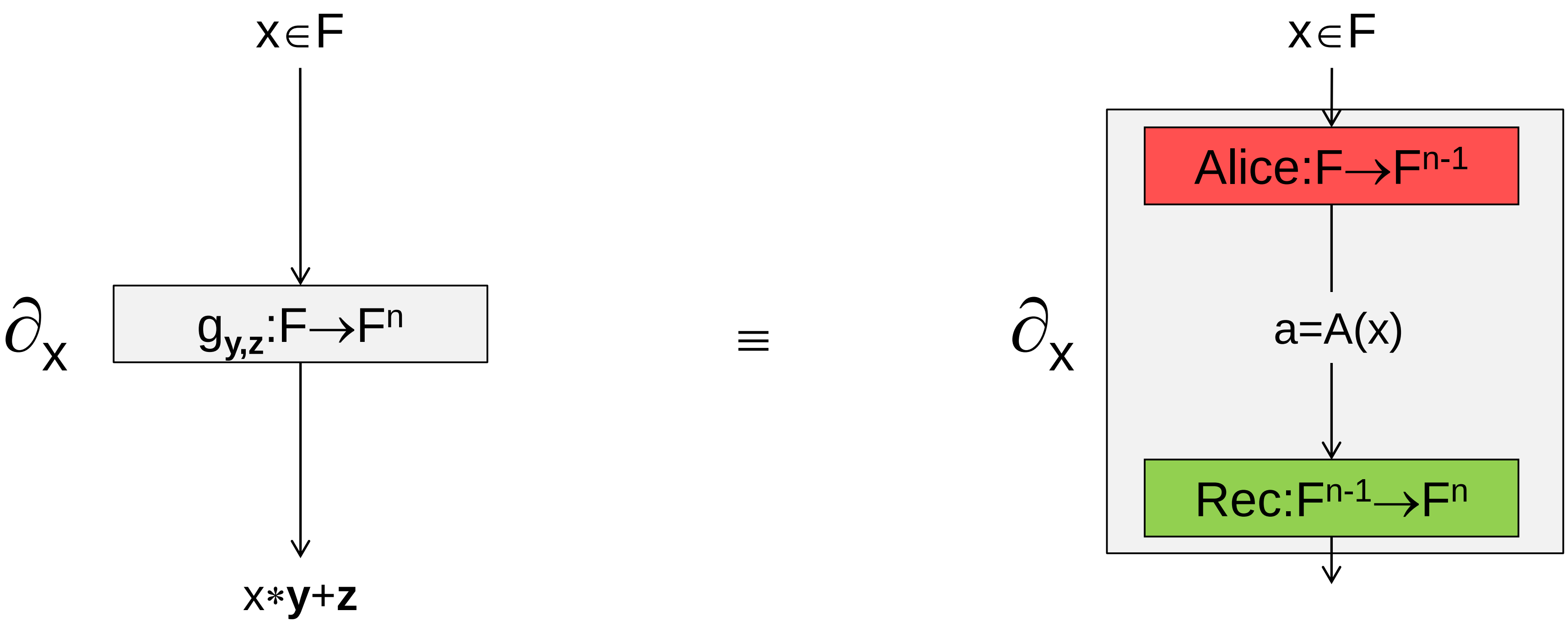
# Observations

Fix a sufficiently large F such that |F|>>exp(circuit-depth)
The formal (univariate) polynomials are equivalent (since the field is large)



# Observations

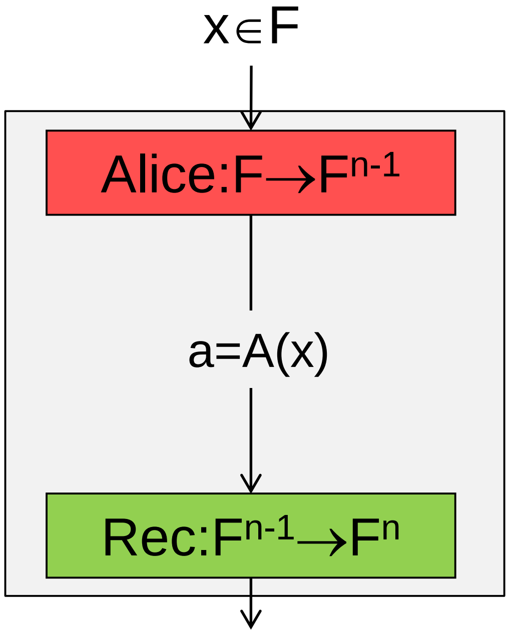
The formal derivatives are also equivalent



# Observations

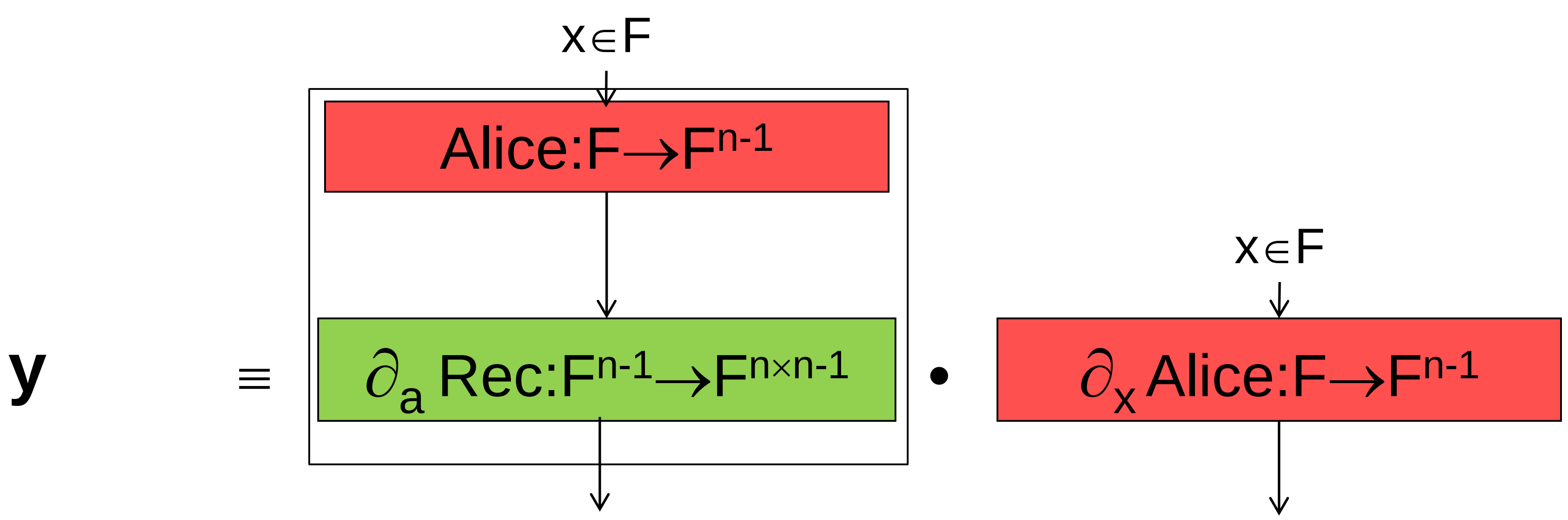
The formal derivatives are also equivalent

$$\mathbf{y} \equiv \partial_x$$



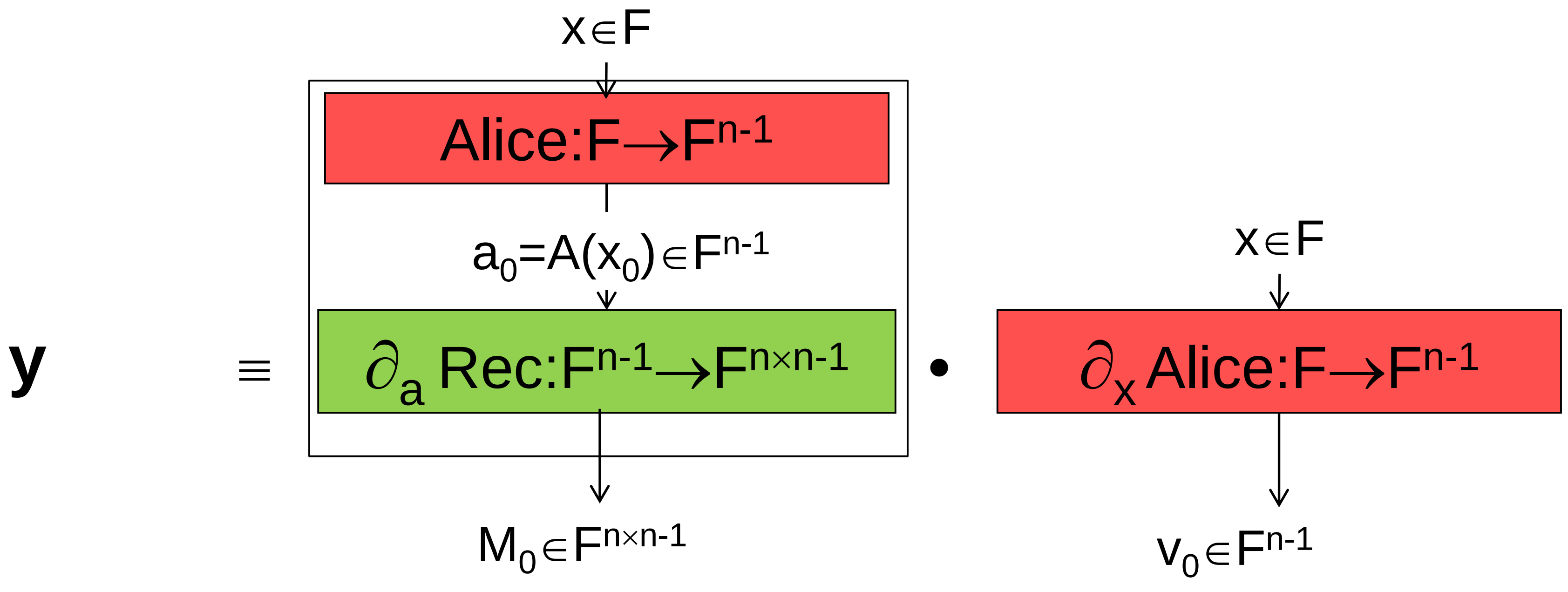
# Observations

By the chain rule $\partial_x P(Q(x))=\partial_Q P(Q(x)) * \partial_x Q(x)$



# Key Observation

- The attacker (Rec) doesn't have Alice's polynomial.
- But has a point $a_0=A(x_0)$ for some $x_0$!
- There must exist a vector $v_0$ such that $M_0*v_0=y$
- So $y \in$ column_Span($M_0$)

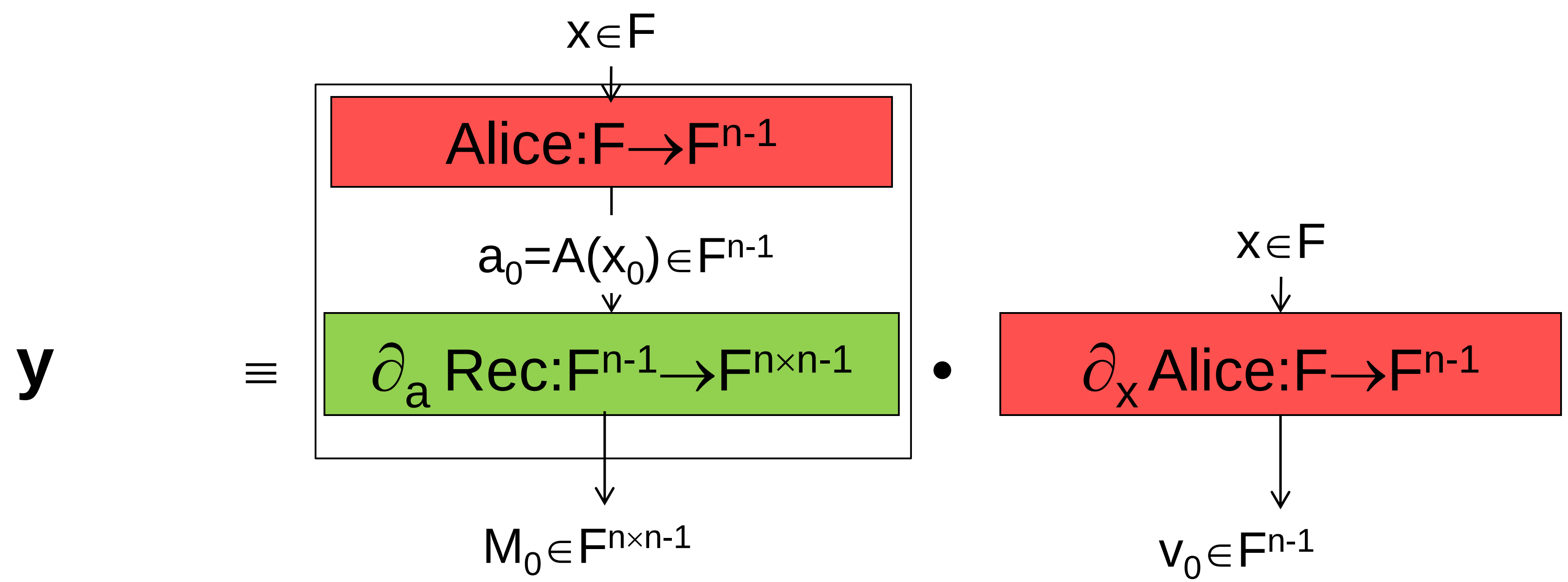


# Key Observation

**Attack**:

- Compute ($n \times n-1$) matrix $M_0$
- Bob's input **y** must be spanned by this matrix
- Find a vector $\mathbf{y^*} \notin span(M_0)$ which is not held by Bob.

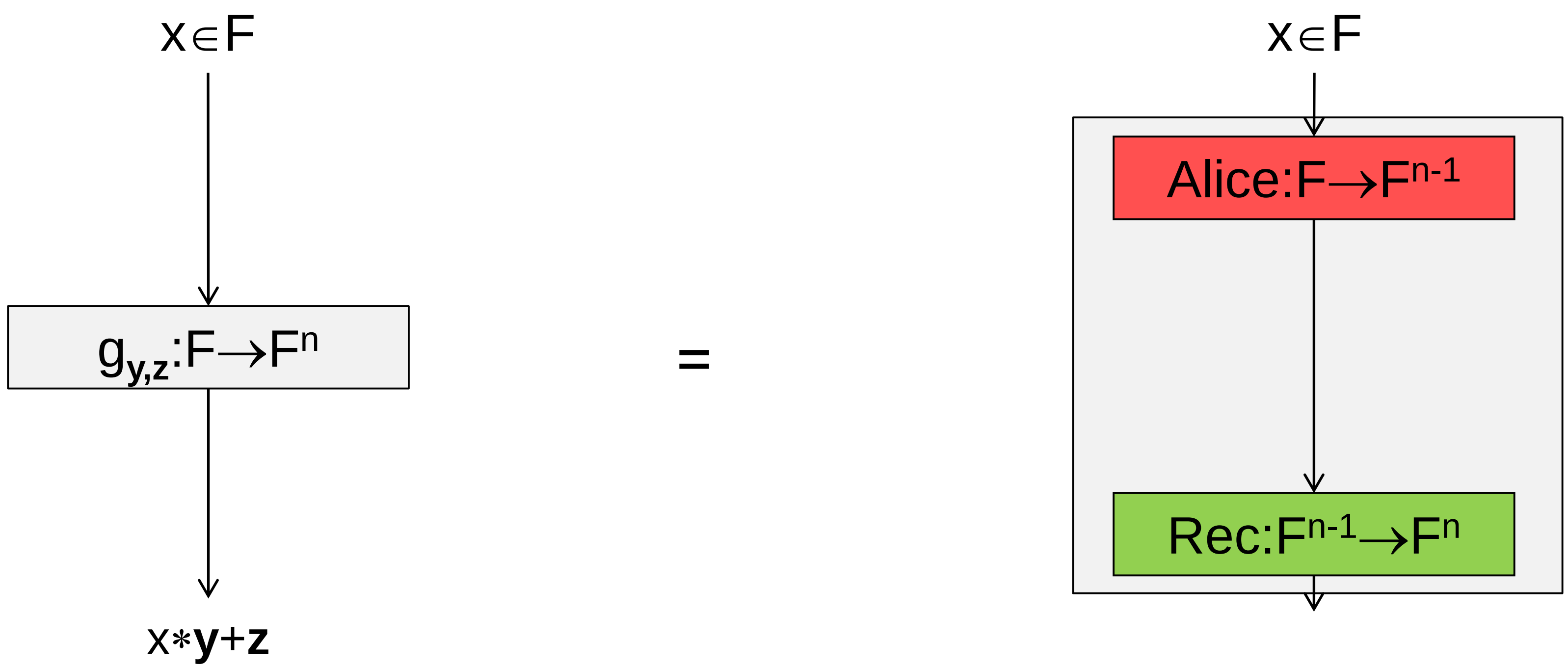
$\Rightarrow$Violates privacy



# Coping with **Is-Zero** gates

Problem: If there are **Is-Zero** gates then the computation of Alice and Receiver is not a polynomial

Sol: Eliminate zero gates

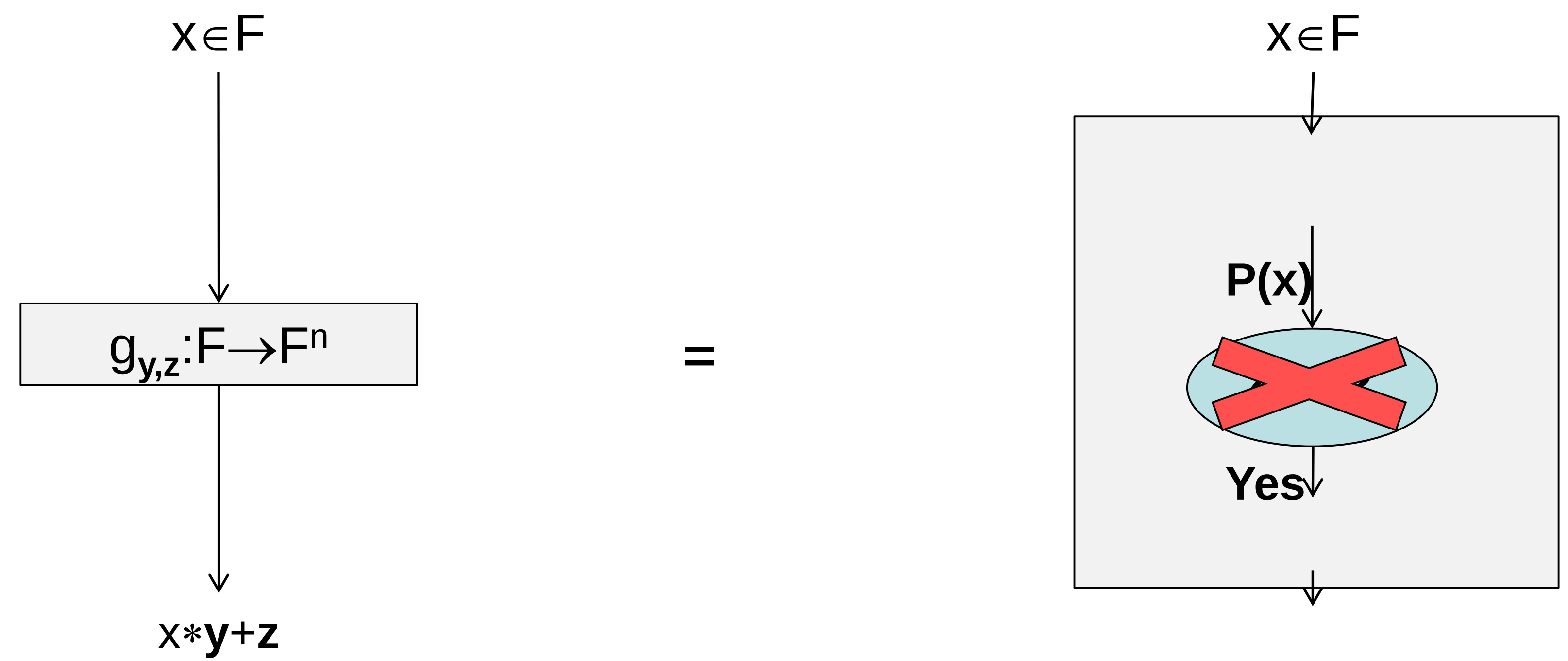


# Coping with **Is-Zero** gates

Consider a single **Is-Zero** gate.

**Case 1**: P is the zero polynomial

$\Rightarrow$ can eliminate the gate



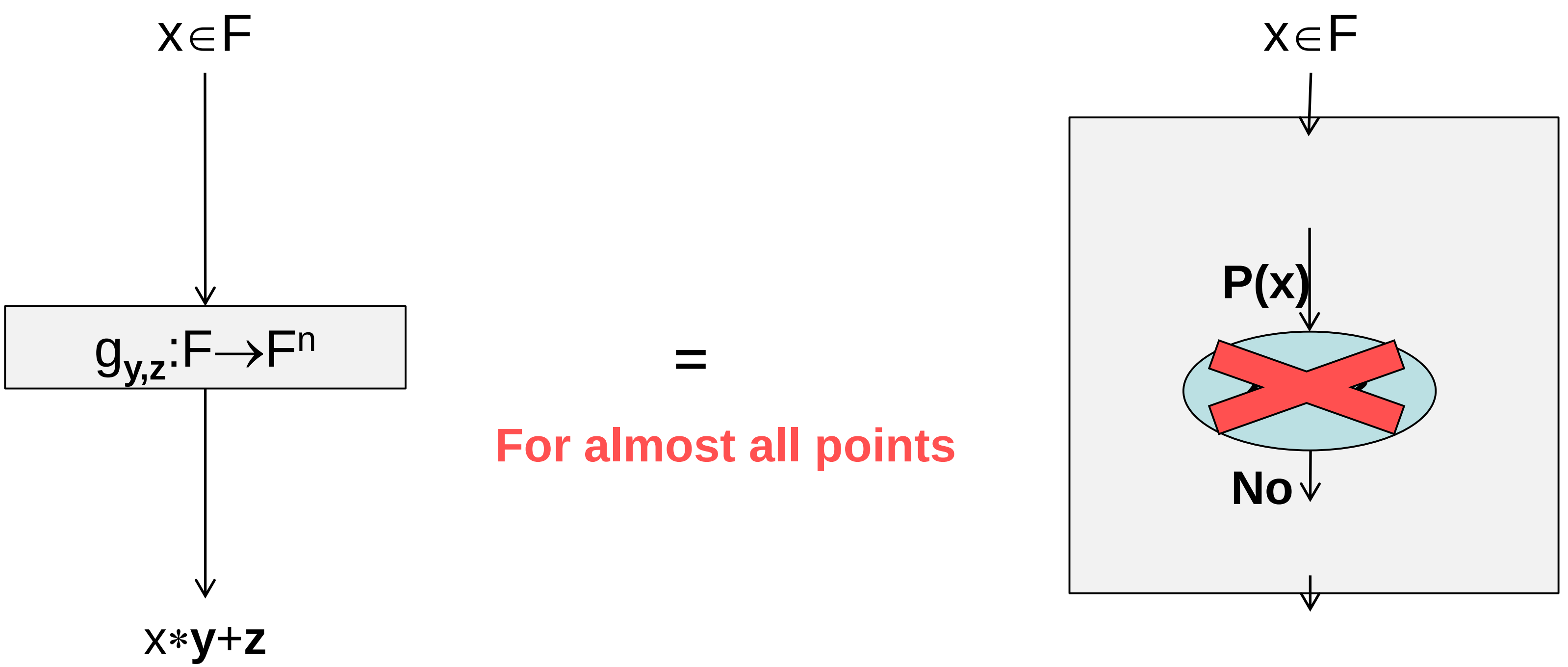
# Coping with **Is-Zero** gates

Consider a single **Is-Zero** gate.

**Case 2**: P is non-zero polynomial of degree< exp(depth) << |**F**|

$\Rightarrow$ For almost all points $P(x) \neq 0$

$\Rightarrow$ Eliminate the gate and get an approximation of g



# Coping with **Is-Zero** gates

Consider a single **Is-Zero** gate.

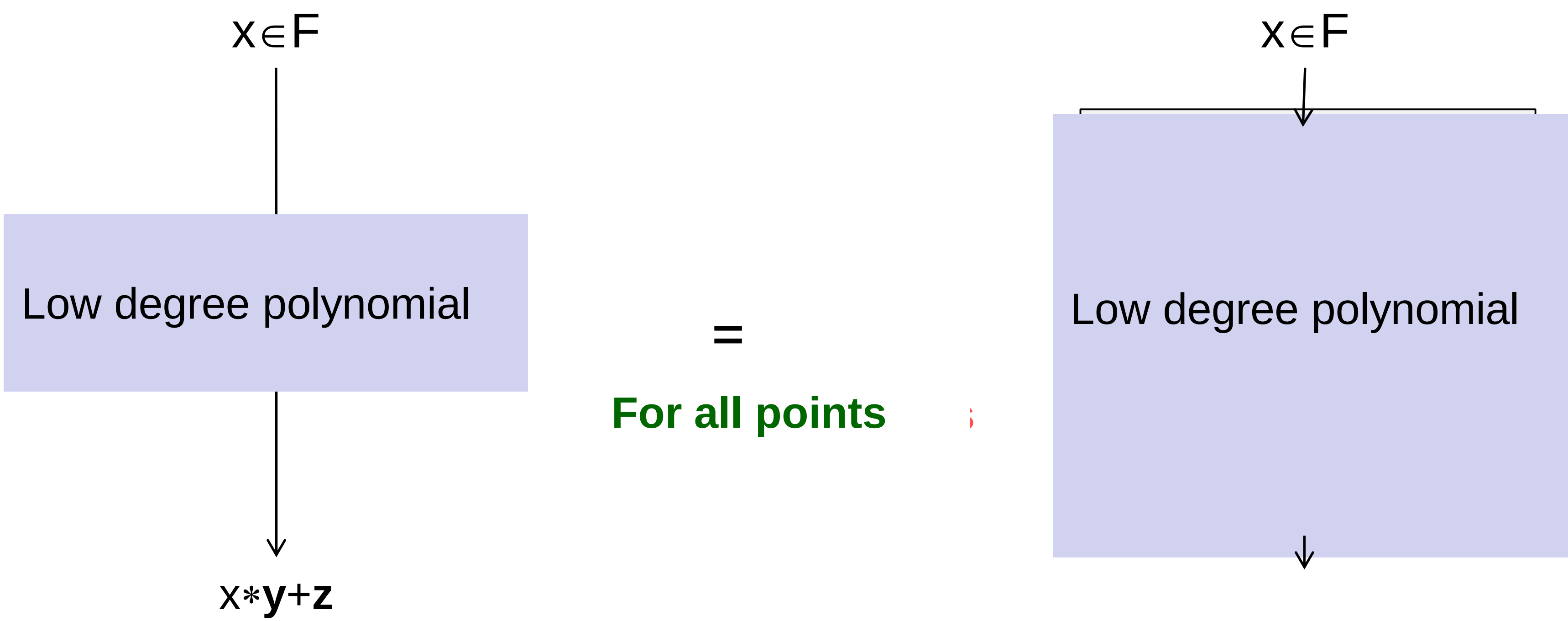
**Case 2**: P is non-zero polynomial of degree< circuit-size << |**F**|

$\Rightarrow$ For almost all points $P(x)\neq 0$

$\Rightarrow$ Eliminate the gate and get an approximation of g

Handle many **Is-Zero** gates iteratively

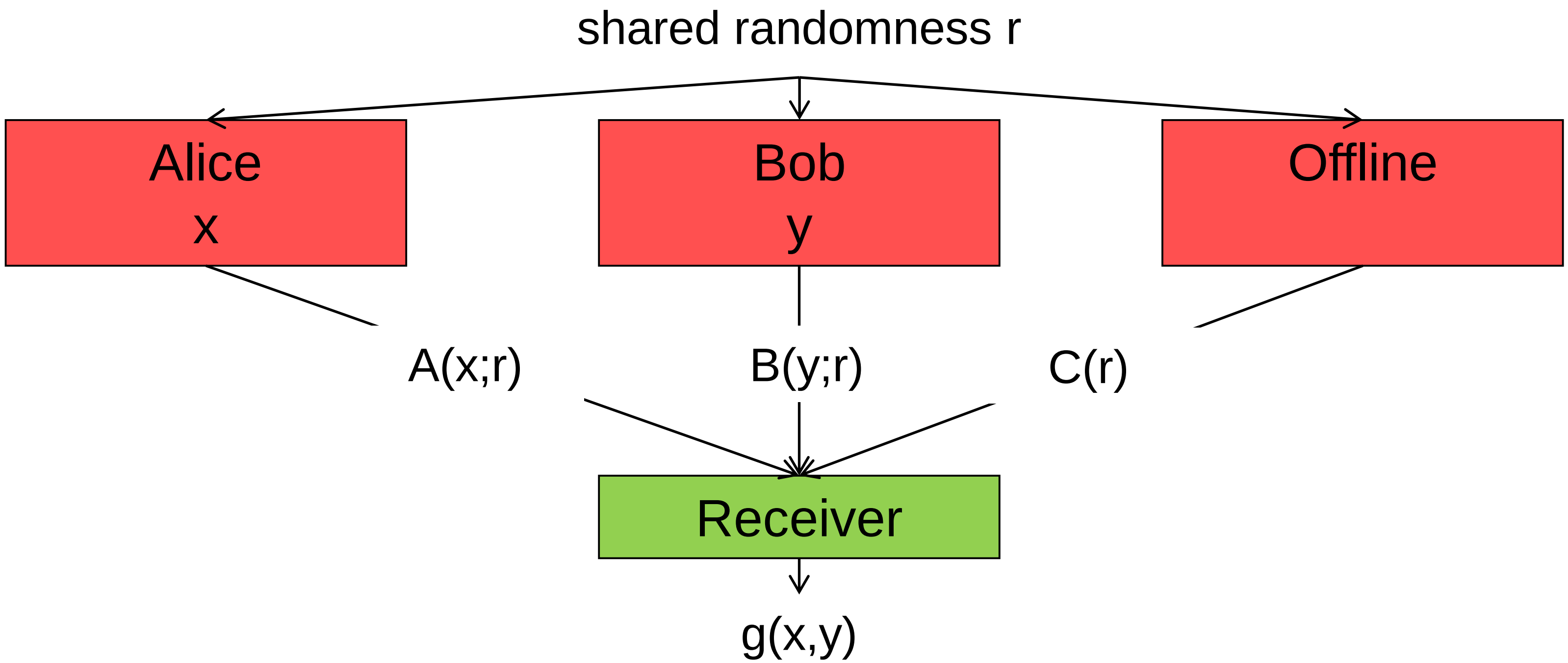
Attack easily generalizes to **Division** gates



# Extension I: Shortening Bob's Input

We showed**:** in the **Arithmetic case** |A(x)|≥|y|

What if both **x** and **y** are short?



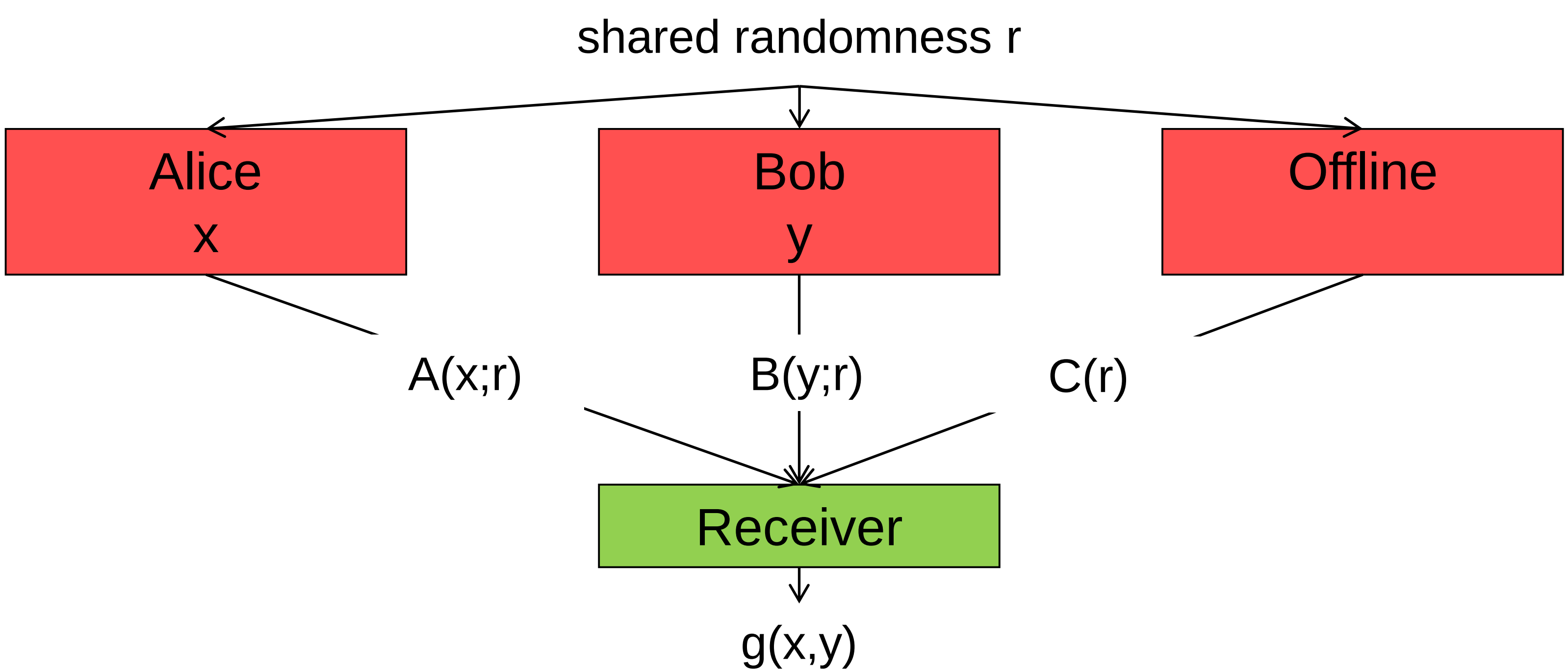
# Extension I: Shortening Bob's Input

**Thm**: Assume the existence of a (standard) pseudorandom generator.
Then, $\forall$ **c>0** there exists a function **g** such that:

- Alice and Bob inputs are of length **n**
- Alice's communication > $\mathbf{n^c}$

**Proof Idea:** Let g(x,seed)=x*Y+Z where (Y,Z)=PRG'(seed)
Low communication $\Rightarrow$ can break the PRG

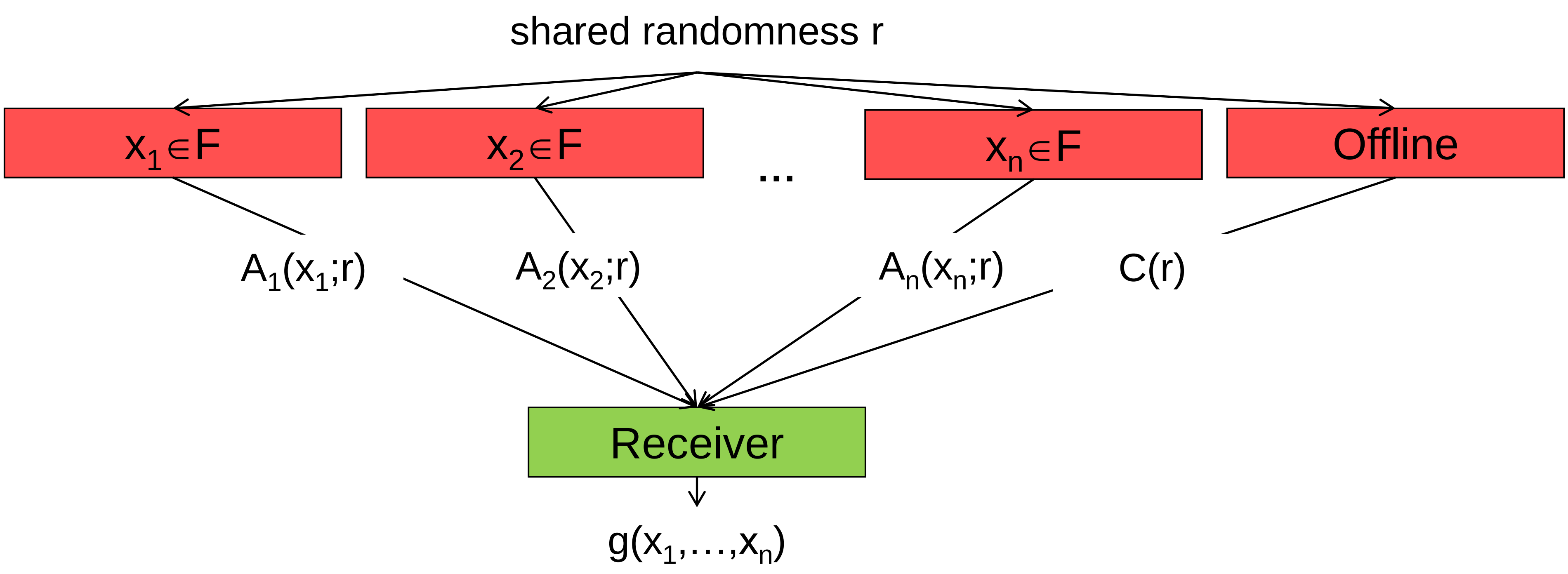
**Open:** Improve to a single-output function



# Extension II: Multiple Players

Each player holds a single input [IK97]
Equivalent to **Decomposable Randomized Encoding**
**(**aka **Projective Garbling Scheme** [BHR]**)**

**Thm**: Assume the existence of a (standard) PRG.
Then, $\forall$ polynomial **m()** there exists a function $\mathbf{g}:\mathbf{F}^{\mathbf{n}}\rightarrow\mathbf{F}^{\mathbf{m}}$ s.t.
each player has to send **m** field elements, total communication: **m*n**.

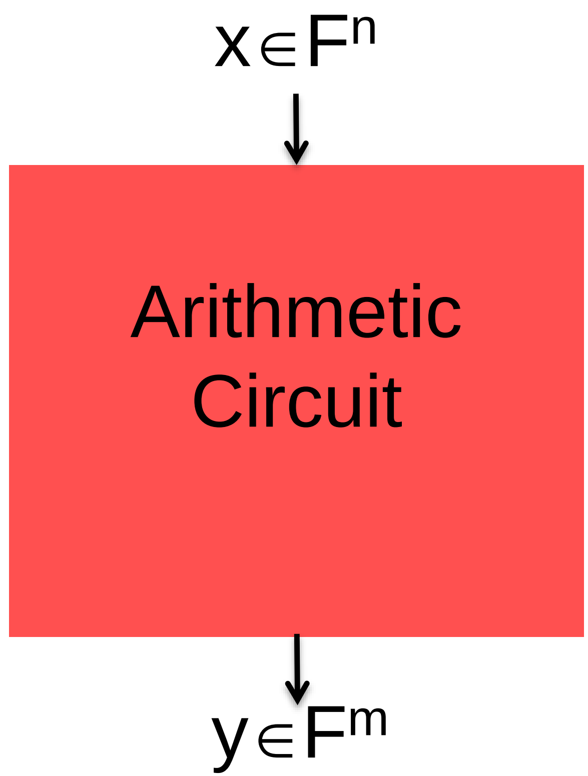


# Impossibility of Homomorphic Encryption

**Thm** [DGW09]: Let $g:\mathbf{F}^n \rightarrow \mathbf{F}^m$ be an arithmetic circuit.
The entropy of the distribution $g(U_n)$ can be approximated

In the binary setting this is hard

- complete for Statistical Zero Knowledge [GV99]



# Impossibility of Homomorphic Encryption

- Assumption: Enc supports scalar multiplication

$$\mathbf{a} \otimes \mathrm{Enc}(b) \equiv \mathrm{Enc}(\mathbf{a} * b)$$

- Given a challenge $c \in \{\mathrm{Enc}(0), \mathrm{Enc}(1)\}$ define:

$$g_c : x \to x \otimes c$$

- If $c=\mathrm{Enc}(1) \Rightarrow g_c(U_n)=E(\mathbf{U_n})$ has high entropy
- If $c=\mathrm{Enc}(0) \Rightarrow g_c(U_n)=E(\mathbf{0})$ has low entropy

$\Rightarrow$ Can break the encryption!

The argument can be extended to other primitives

# A word about Positive Results

# Arithmetic Public-Key based on Alekhnnovich

**Public-key:** (A,b)

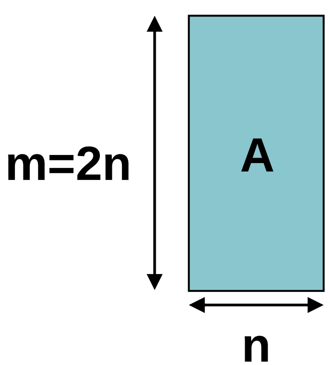
**Private-key:** low-weight vector $e \in \mathrm{ColSpan}(A,b)$

**Encrypt(x)**: $r \leftarrow \mathrm{Ker}(A,b)$, $e' \leftarrow \mathrm{Weight}(\sqrt{n})$

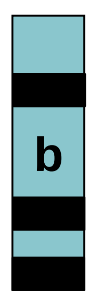
output $c = r + e' + x \cdot \mathbf{1}$

**Decryption:** $\langle c,e \rangle / |e|$

$= (\langle r,e \rangle + \langle e',e \rangle + \langle x \cdot \mathbf{1}, e \rangle)/|e| =_{\mathrm{whp}} x$




Random Code




$\sqrt{n}$-noisy codeword

**RLC assumption(m,ε):**
(A,b) is pseudorandom

# Arithmetic Public-Key based on Alekhnnovich

**Public-key:** (A,b)

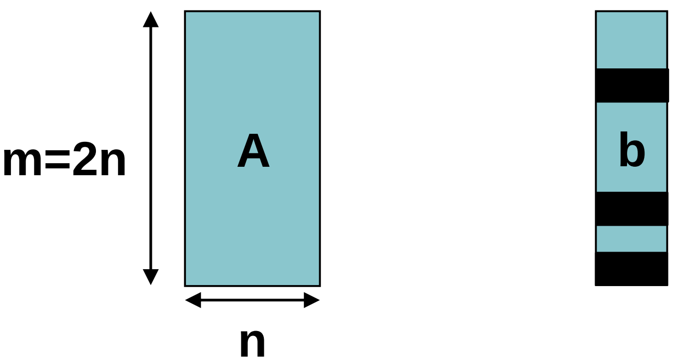
**Private-key:** low-weight vector e $\in$ColSpan(A,b)

**Observation:** The scheme has a “lossy mode”

If b is replaced with a random vector decryption is computationally infeasible

$\Rightarrow$(1:2)-Arithmetic OT

$\Rightarrow_{\textbf{RLC}}$ Oblivious Linear Function Evaluation [NP,IPS]




Random Code

$\sqrt{}$n-noisy codeword

**RLC assumption(m,$\varepsilon$):**
(A,b) is pseudorandom

# Conclusion

- New (stronger) notion of Arithmetic Cryptography
  - Captures classical information-theoretic results
- Feasibility results for computational crypto
- Non-trivial lower-bounds
  - Communication complexity of MPC
  - Different technique to rule out Homomorphic Encryption

# Future Works: Negative

**Hope**: Establish stronger lower-bounds on efficient information-theoretic cryptography

– Several old (and hard) open problem

Arithmetic setting is a new promising starting point

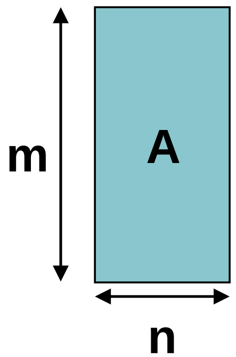
- Easier for lower-bounds
- Meaningful as it captures natural IT-MPC

# Future Works: Positive

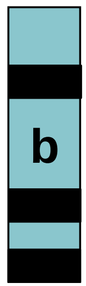
Construct more primitives in the Arithmetic model

- Hash functions, Signatures, PRFs?

Understand the **Random Linear Code** assumption




Random Code




ε-noisy codeword

**RLC assumption(m,ε):**
(A,b) is pseudorandom

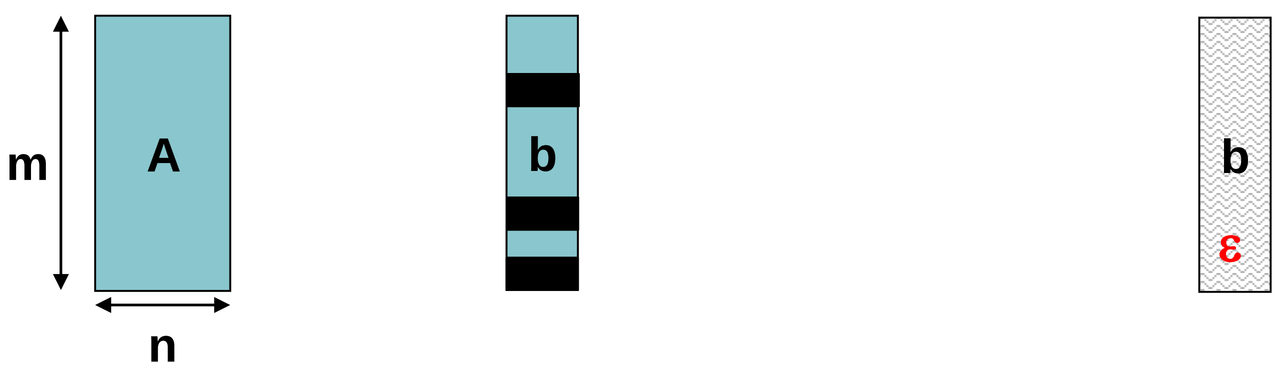
# Future Works: Positive

Construct more primitives in the Arithmetic model

- Hash functions, Signatures, PRFs?

Understand the **Random Linear Code** assumption

- Harder or easier than LWE?




Random Code   ε-noisy codeword   Gaussian noise of width ε