Exploring Crypto Dark Matter:
New Simple PRF Candidates and Their Applications

Dan Boneh, Yuval Ishai, Alain Passelègue,
Amit Sahai, and David J. Wu

How Do We Design Cryptographic Primitives?

1. Introduce hardness assumption (e.g., RSA, discrete log , LWE)
2. Reduce security to breaking hardness assumption

Theory-Driven

1. Design primitive (e.g., block ciphers, hash functions) with focus on concrete efficiency
2. Security relies on heuristics, cryptanalysis

Practice-Oriented

How Do We Design Cryptographic Primitives?

1. Introduce hardness assumption (e.g., RSA, discrete log , LWE)
2. Reduce security to breaking hardness assumption

Theory-Driven

Concrete efficiency of these constructions often limited by structure of computational assumptions (e.g., algebraic PRFs vs. AES)

Often exist non-trivial attacks (e.g., sub-exponential attacks, quantum attacks)

How Do We Design Cryptographic Primitives?

Designs often complex and difficult to analyze

Security based on heuristics, experience, cryptanalysis

Typically, designs tailored to one type of application

1. Design primitive (e.g., block ciphers, hash functions) with focus on concrete efficiency
2. Security relies on heuristics, cryptanalysis

Practice-Oriented

The Landscape of Cryptography

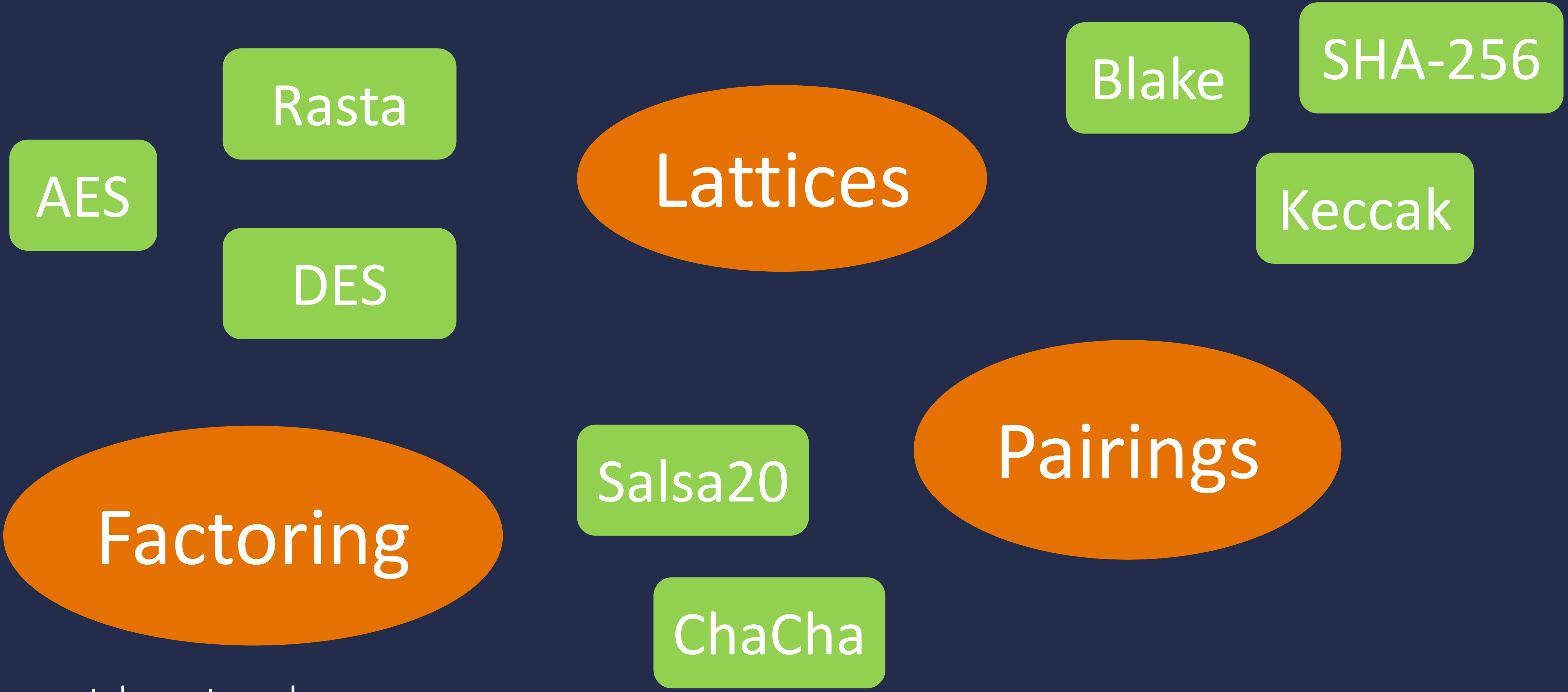


Figure not drawn to scale

The Landscape of Cryptography

Blake

SHA-256

Rasta

Lattices

AES

Keccak

DES

Pairings

Salsa20

Factoring

ChaCha

Crypto Dark Matter

Figure not drawn to scale

Exploring Crypto Dark Matter

Goals: Explore simplest unexplored areas of cryptography and better understand landscape and boundaries of cryptographic hardness

Exploring Crypto Dark Matter

Goals: Explore simplest unexplored areas of cryptography and better understand the scope and boundaries of cryptographic hardne

We seek assumptions that are simple to describe, but breaking them would have positive consequences in other domains (a “win-win” flavor)

Exploring Crypto Dark Matter

Goals: Explore simplest unexplored areas of cryptography and better understand landscape and boundaries of cryptographic hardness

Design Criterion:

- Primitive should be simple to describe and analyze
- Good concrete efficiency
- Well-suited for other cryptographic applications (e.g., MPC)

Examples:

- Goldreich's one-way function based on expander graphs [Gol01]
- Miles and Viola [MV12] and Akavia et al. [ABGKR14] work on constructing low-complexity PRFs

Exploring Crypto Dark Matter

Goals: Explore simplest unexplored areas of cryptography and better understand landscape and boundaries of cryptographic hardness

Our Focus: (weak) pseudorandom functions (PRFs)

PRF: keyed function whose input-output behavior is indistinguishable from a truly random function

Exploring Crypto Dark Matter

Goals: Explore simplest unexplored areas of cryptography and better understand landscape and boundaries of cryptographic

Basic building block for secret-key cryptography (e.g., encryption schemes, message authentication codes, digital signatures, and many more)

PRF: keyed function whose input-output behavior is indistinguishable from a truly random function

Exploring Crypto Dark Matter

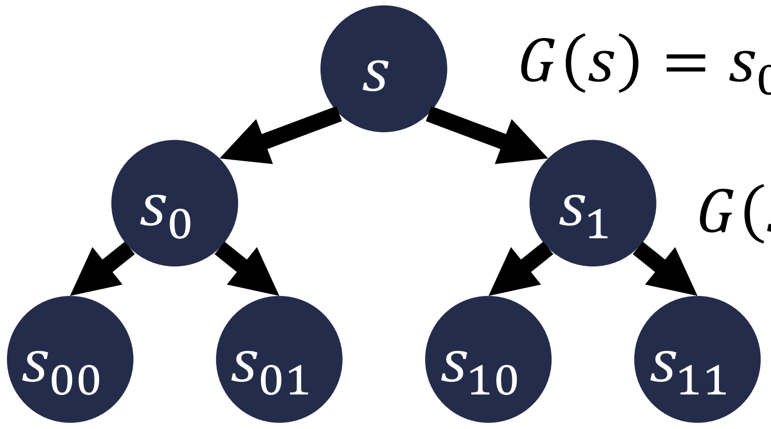
Goals: Explore simplest unexplored areas of cryptography and better u c hardnes

Weak PRF: input-output behavior looks random given PRF evaluations at *random* inputs

Our Focus: (weak) pseudorandom functions (PRFs)

PRF: keyed function whose input-output behavior is indistinguishable from a truly random function

Existing PRF Candidates



$G(s) = s_0 \parallel s_1$

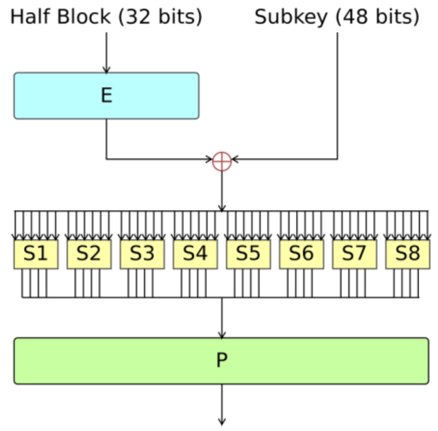
$G(s_1) = s_{10} \parallel s_{11}$

[GGM84]

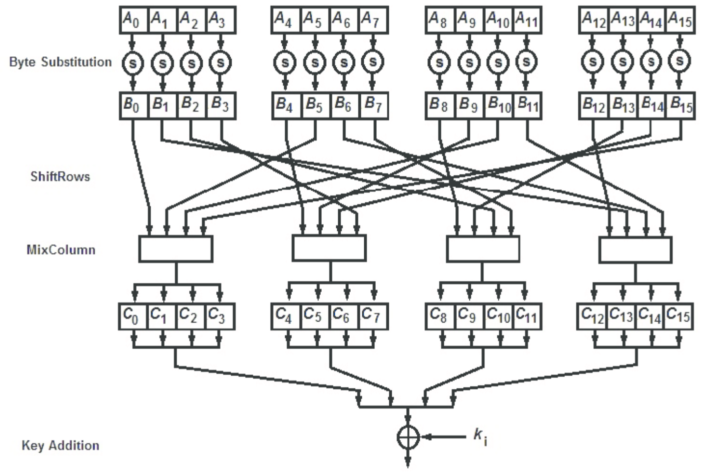
$$F((h, k_1, k_2, \dots, k_n), x) := h^{\prod_{i \in [n]} k_i^{x_i}}$$

[NR97]

Theory-Driven



DES


Byte Substitution
ShiftRows
MixColumn
Key Addition
A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15}
S
B_0 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15}
C_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9 C_{10} C_{11} C_{12} C_{13} C_{14} C_{15}
k_i

AES

Practice-Oriented

Hardness from Modulus Mixing

Define the function map: $\{0,1\}^n \to \mathbb{Z}_3$:

$$\text{map}(x) := \sum_{i \in [n]} x_i \pmod 3$$

"mod-3 sum of binary vector"

Razborov-Smolensky: the map function *cannot* be computed by a low-degree polynomial over $\mathbb{Z}_2$

Hardness from Modulus Mixing

Define the function map: $\{0,1\}^n \rightarrow \mathbb{Z}_3$:

$$\mathrm{map}(x) := \sum_{i}$$

Could this be a source of hardness?

“mod-3 sum of binary ector

<u>Razborov-Smolensky:</u> the map function <u>*cannot*</u> be computed by a low-degree polynomial over $\mathbb{Z}_2$

Our Weak PRF Candidate

$$F_{\boldsymbol{A}}(x) := \text{map}\left(\boldsymbol{A} \times x\right)$$

PRF key

input

$\boldsymbol{A} \in \mathbb{Z}_2^{n \times n}$ $\quad x \in \mathbb{Z}_2^n$

“secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3”

Our Weak PRF Candidate

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"

Conjecture (Informal)**:** The above function family is a weak PRF family.

Basic conjecture: advantage of $\mathrm{poly}(\lambda)$-time adversary is $\mathrm{negl}(\lambda)$ when $n = \mathrm{poly}(\lambda)$

Stronger conjecture: advantage of 2^{λ}-time distinguishers is $2^{-\Omega(\lambda)}$ when $n = O(\lambda)$ – *exponential hardness*

Our Weak PRF Candidate

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"

Conjecture (Informal)**:** The above function family is a <u>weak</u> PRF family.

Candidate is <u>not</u> a strong PRF: can be expressed as a certain sparse polynomial over $\mathbb{Z}_3$ (which can be distinguished from random given *non-adaptive* queries)

Our Weak PRF Candidate

$$F_{\boldsymbol{A}}(x) := \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"

Conjecture (Informal)**:** The above function family is a <u>weak</u> PRF family.

Many extensions and variants:

- Replace mod-2/mod-3 with mod-p/mod-q
- Multiple output bits: replace "sum mod-3" with matrix-vector product mod-3
- Compact keys: take $\boldsymbol{A}$ to be a structured matrix (e.g., Toeplitz matrix)

Focus will be basic candidate above

Why Is This (Plausibly) Secure?

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"

<u>Razborov-Smolensky</u>: the function $F_{\boldsymbol{A}}$ cannot be expressed as a low-degree polynomial over $\mathbb{F}_p$ (due to mixing of different moduli)

Conjecture: For distinct primes p, q, the MOD_p gate cannot be computed by a rational polynomial over $\mathbb{F}_{q^\ell}$ for any $\ell \geq 1$.

Why Is This (Plausibly) Secure?

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n \times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"

Can rule out learning attacks along the lines of Linial et al. [LMN89]

- Can show that above function family is only *negligibly* correlated with any fixed function family of size $2^{n/2}$

BKW-style attacks (for LPN) rely on constructing new samples by taking linear combinations of existing samples – but the map function is highly *non-linear*

We invite further cryptanalysis of our candidates!

Is This Simple?

$$F_{\boldsymbol{A}}(x) := \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"

Conceptual simplicity:
easy to describe; no mention of groups or S-boxes

Complexity-theoretic:
can be computed by a *depth-2* ACC^0 circuit

Complexity-Theoretic Implications

What is the "minimal" complexity class that contains (weak) PRFs (with exponential security)?

Complexity-Theoretic Implications

What is the "minimal" complexity class that contains (weak) PRFs (with exponential security)?

	AC^0	$\mathrm{ACC}^0[p]$	$\mathrm{ACC}^0[m]$
Depth 2			**This Work:** Weak PRF (exponential)
Depth 3	Weak PRF [AR16] (quasi-polynomial)	Weak PRF [ABGKR14] (quasi-polynomial)	**This Work:** Strong PRF (exponential)
Depth ≥ 3	Weak PRF [Kha93] (quasi-polynomial)	Strong PRF [Vio13] (quasi-polynomial)	

No strong PRFs for broad classes of depth-2 circuits [BV96]

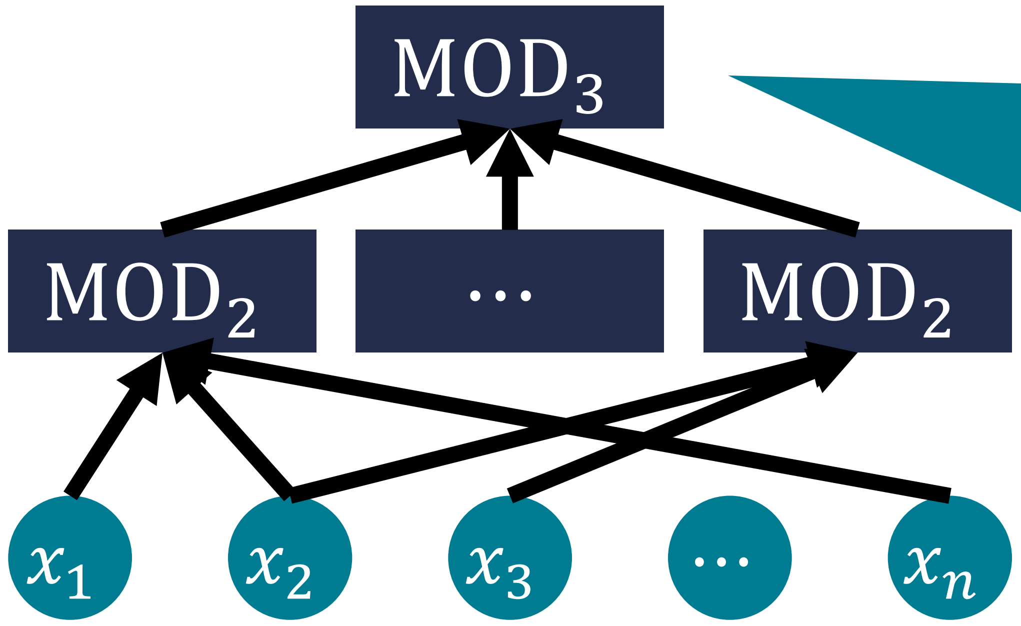
No weak PRFs with better than quasi-polynomial security [LMN89]

No strong PRFs with better than quasi-polynomial security [CIKK16]

Complexity-Theoretic Implications

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

“secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3”

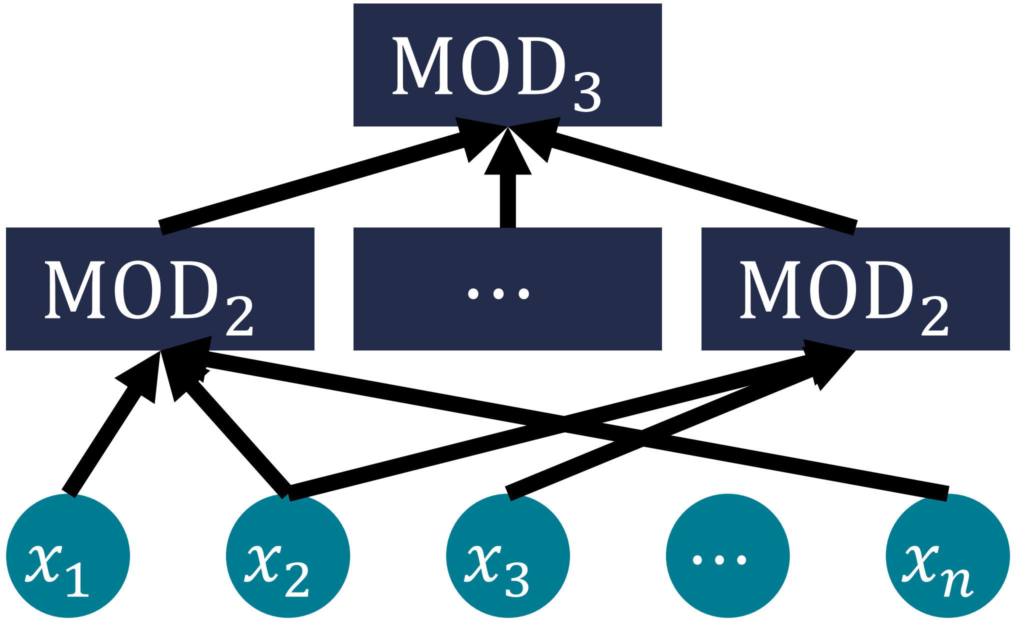


Technically, MOD_3 gate outputs just a single bit (but can use MOD_3 gates to compute binary representation of $\mathbb{Z}_3$ value)

Complexity-Theoretic Implications

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"



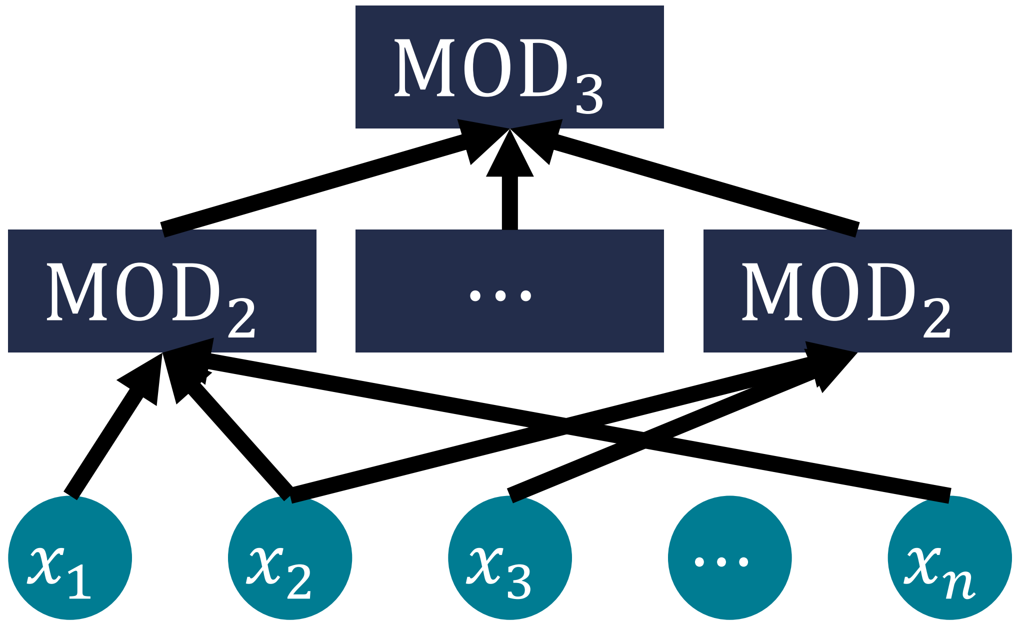
For fixed $\boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$, $F_{\boldsymbol{A}}(\cdot)$ can be computed by a <u>depth-2</u> ACC^0 circuit

First candidate weak PRF computable by depth-2 ACC^0

Complexity-Theoretic Implications

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"



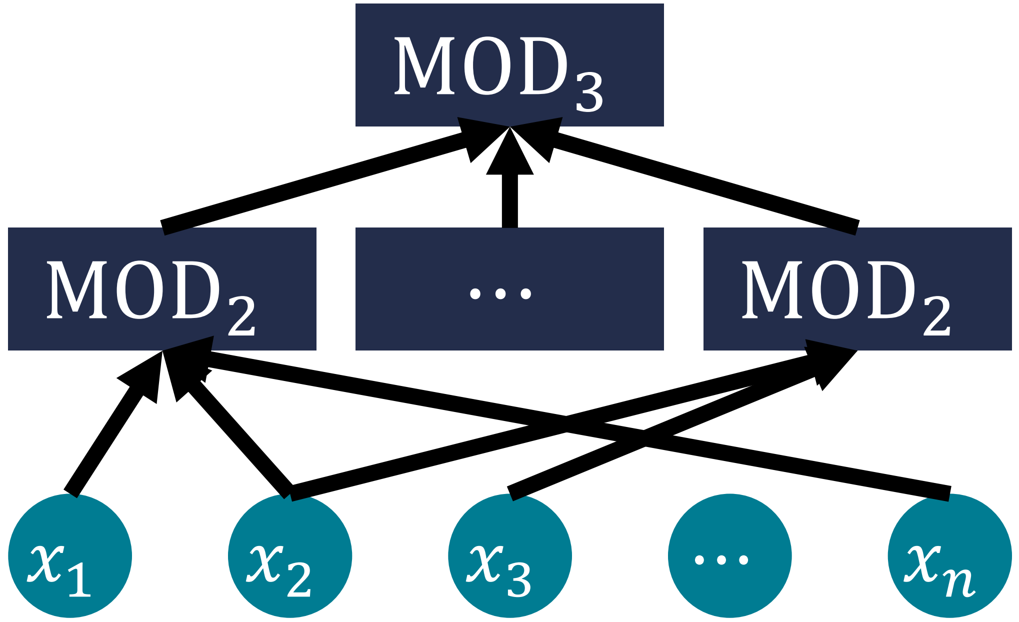
For fixed $\boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$, $F_{\boldsymbol{A}}(\cdot)$ can be computed by a depth-2 ACC^0 circuit

First candidate weak PRF with plausible exponential security from constant-depth ACC^0

Complexity-Theoretic Implications

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"



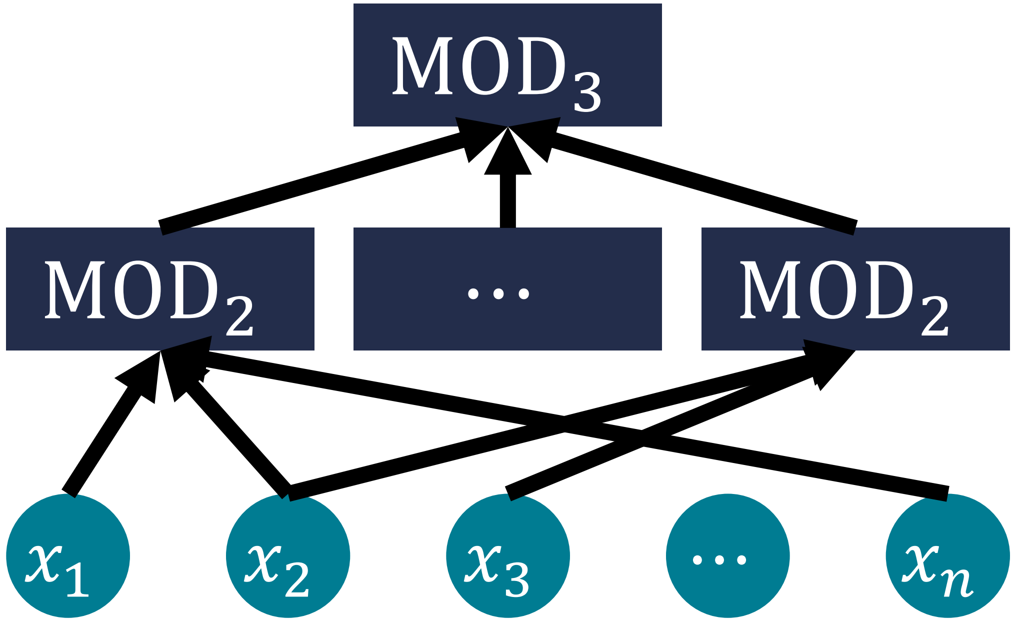
For fixed $\boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$, $F_{\boldsymbol{A}}(\cdot)$ can be computed by a <u>depth-2</u> ACC^0 circuit

Implication: ACC^0 is not PAC-learnable in sub-exponential time under the uniform distribution (in contrast, AC^0 can be learned in quasi-polynomial time with uniform samples)

Complexity-Theoretic Implications

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

“secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3”



Barrington [Bar85] previously showed that circuits of this form can be computed by width-3 branching programs

Implication: Width-3 branching programs are not PAC-learnable under the uniform distribution (learning width-2 branching programs is easy)

Another View: Sparse Polynomial Interpolation

$$F_{\boldsymbol{A}}(x) \coloneqq \text{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"

Consider a change of variables: $y_i \coloneqq 1 + x_i \pmod 3$

$$0 \mapsto 1 \text{ and } 1 \mapsto -1$$

Then, $\langle \boldsymbol{A}_i, x\rangle \pmod 2 \mapsto \prod_{j\in[n]} y_j^{\boldsymbol{A}_{i,j}}$

$$F_{\boldsymbol{A}}(y) \coloneqq \sum_{i\in[n]} \prod_{j\in[n]} y_j^{\boldsymbol{A}_{i,j}} \pmod 3$$

Another View: Sparse Polynomial Interpolation

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}(\boldsymbol{A}x) \text{ where } \boldsymbol{A} \in \mathbb{Z}_2^{n\times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"

Consider a change of variables: $y_i \coloneqq 1 + x_i \pmod 3$

$0 \mapsto 1$ and 1

Then, $\langle \boldsymbol{A}_i, x\rangle \pmod 2 \mapsto \prod_{j\in[n]} y_j^{A_{i,j}}$

Sparse multilinear polynomial of degree n over $\mathbb{Z}_3$ (only n non-zero terms)

$$F_{\boldsymbol{A}}(y) \coloneqq \sum_{i\in[n]} \prod_{j\in[n]} y_j^{\boldsymbol{A}_{i,j}} \pmod 3$$

Another View: Sparse Polynomial Interpolation

Natural direction for cryptanalysis: Can we interpolate sparse (multilinear) polynomials (over $\mathbb{Z}_3$) given *random* evaluations drawn from $\{-1,1\}^n$

Under our conjectures, both interpolation (and even property testing) for such polynomials is difficult

$$F_{\boldsymbol{A}}(y) := \sum_{i \in [n]} \prod_{j \in [n]} y_j^{\boldsymbol{A}_{i,j}} \pmod 3$$

Another View: Sparse Polynomial Interpolation

Natural direction for cryptanalysis: Can we interpolate sparse (multilinear) polynomials (over $\mathbb{Z}_3$) given *random* evaluations drawn from $\{-1,1\}^n$

Existing interpolation algorithms require making queries over the full domain (not much known about random queries over a subset of the domain)

$$F_A(y) := \sum_{i\in[n]} \prod_{j\in[n]} y_j^{A_{i,j}} \pmod 3$$

Another View: Sparse Polynomial Interpolation

Natural direction for cryptanalysis: Can we interpolate sparse (multilinear) polynomials (over $\mathbb{Z}_3$) given *random* evaluations drawn from $\{-1,1\}^n$

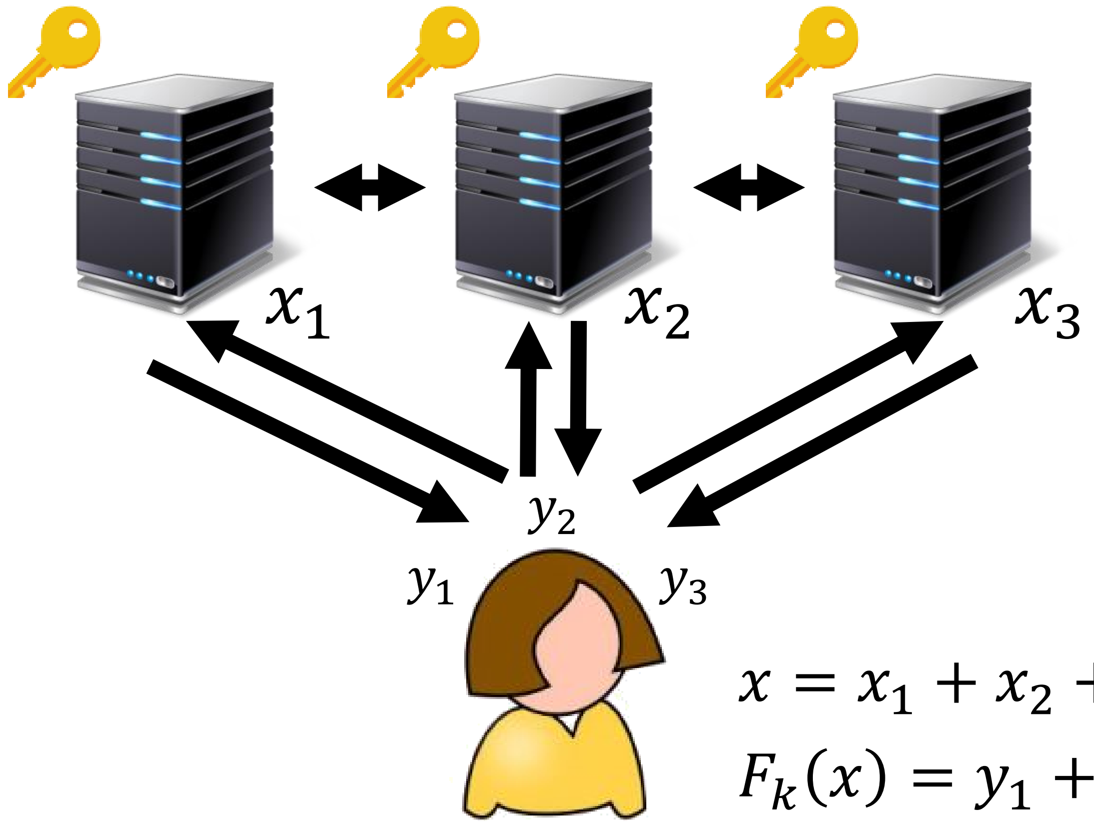
Representation as a sparse polynomial admits a *non-adaptive* distinguisher: fix all but $O(\log n)$ variables and test whether restricted polynomial can be represented as a sparse polynomial

$$F_{\boldsymbol{A}}(y) := \sum_{i\in[n]} \prod_{j\in[n]} y_j^{\boldsymbol{A}_{i,j}} \pmod 3$$

Distributed PRF Evaluation

secret key is secret-shared across multiple parties

$$k = k_1 + k_2 + k_3 \pmod{m}$$



$$x = x_1 + x_2 + x_3 \pmod{m}$$
$$F_k(x) = y_1 + y_2 + y_3 \pmod{m}$$

In typical MPC protocols, costs (e.g., communication, number of rounds, etc.) scale with the number of non-linear operations

Distributed PRF Evaluation

"Pairwise linear" structure very suitable for "MPC with preprocessing" model

	Round Complexity	Online Communication (bits)	Preprocessing Size (bits)
Yao + AES	2	$64.8 \cdot 10^3$	$1491.2 \cdot 10^3$
Yao + LowMC	2	$64.8 \cdot 10^3$	$292.1 \cdot 10^3$
Our Candidate	4	$2.6 \cdot 10^3$	$3.5 \cdot 10^3$

Comparison for two-party distributed PRF evaluation

From Weak PRFs to Strong PRFs

$$F_A(x) \coloneqq \mathrm{map}(Ax) \text{ where } A \in \mathbb{Z}_2^{n\times n}$$

Not a strong PRF:

- Sparse polynomial representation gives a non-adaptive attack on the PRF candidate
- Can also represent PRF as an “automata with multiplicity” which can be learned in polynomial time with membership queries [BV94]

$$F_A(y) \coloneqq \sum_{i\in[n]} \prod_{j\in[n]} y_j^{A_{i,j}} \pmod 3$$

From Weak PRFs to Strong PRFs

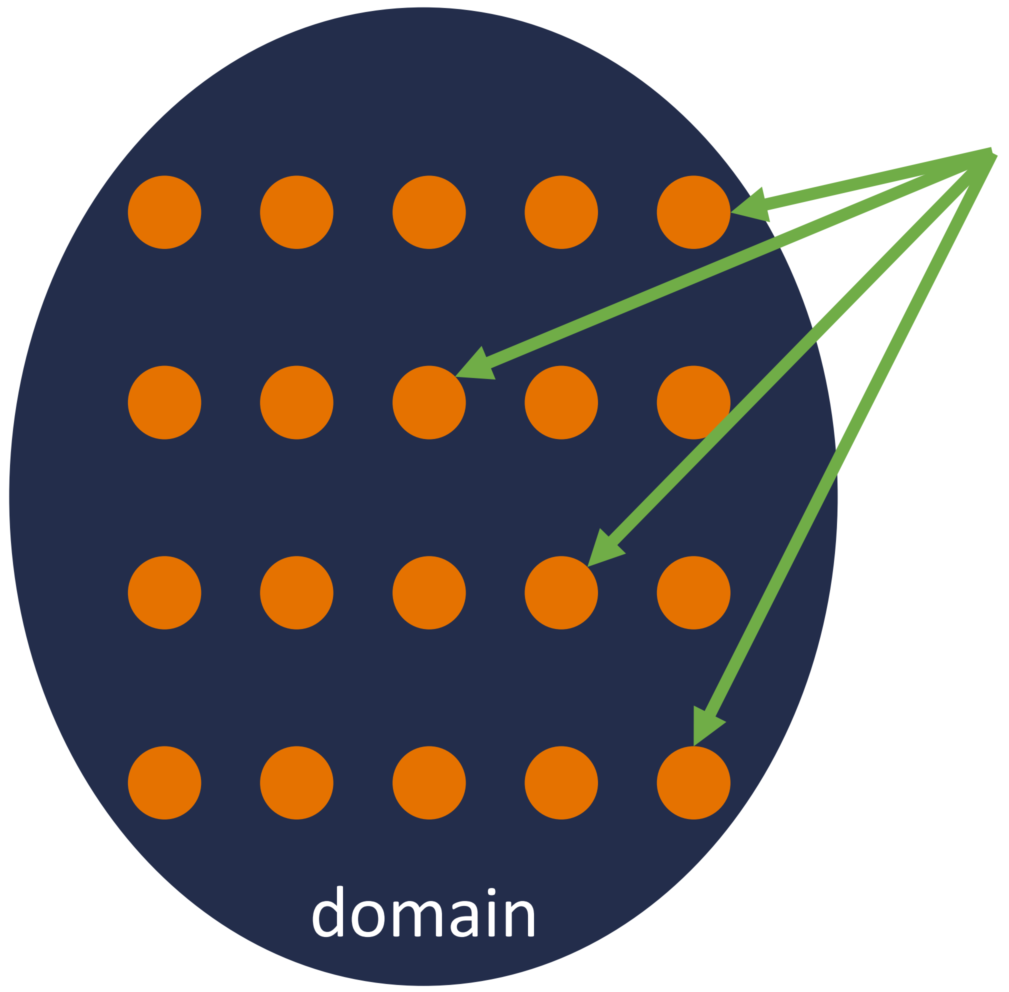
$$F_A(x) \coloneqq \mathrm{map}(Ax) \text{ where } A \in \mathbb{Z}_2^{n \times n}$$

Not a strong PRF:

- Sparse polynomial representation gives a non-adaptive attack on the PRF candidate
- Can also represent PRF as an “automata with multiplicity” which can be learned in polynomial time with membership queries [BV94]

Idea: Avoid attacks by requiring that valid PRF inputs are “far” away

Encoded-Input PRFs



Encoded-input PRF: function whose behavior is pseudorandom on a <u>sparse</u> subset of the domain

(F, E) is an encoded-input PRF if $F'(k, x) \coloneqq F\big(k, E(x)\big)$ is a <u>strong</u> PRF

Advantage: <u>checking</u> that an input is properly encoded is simple (depth-2 circuit); this is useful for many applications

Encoded-Input PRFs

Implication: If F can be computed by a low-depth circuit, then the combination of checking than an input is properly-encoded + computing F is also low-depth (even if E is complex!)

Given EI-PRF with low-depth F:

- Symmetric encryption with low-depth decryption
- MACs with low-depth verification
- CCA-secure symmetric encryption with low-depth decryption

Encoded-input PRF: function whose behavior is pseudorandom on a <u>sparse</u> subset of the domain

(F, E) is an encoded-input PRF if $F'(k, x) \coloneqq F\big(k, E(x)\big)$ is a <u>strong</u> PRF

Advantage: <u>checking</u> that an input is properly encoded is simple (depth-2 circuit); this is useful for many applications

Encoded-Input PRFs

Implication: If F can be computed by a low-depth circuit, then the combination of checking than an input is properly-encoded + computing F is also low-depth (even if E is complex!)

Given EI-PRF with low-depth F:

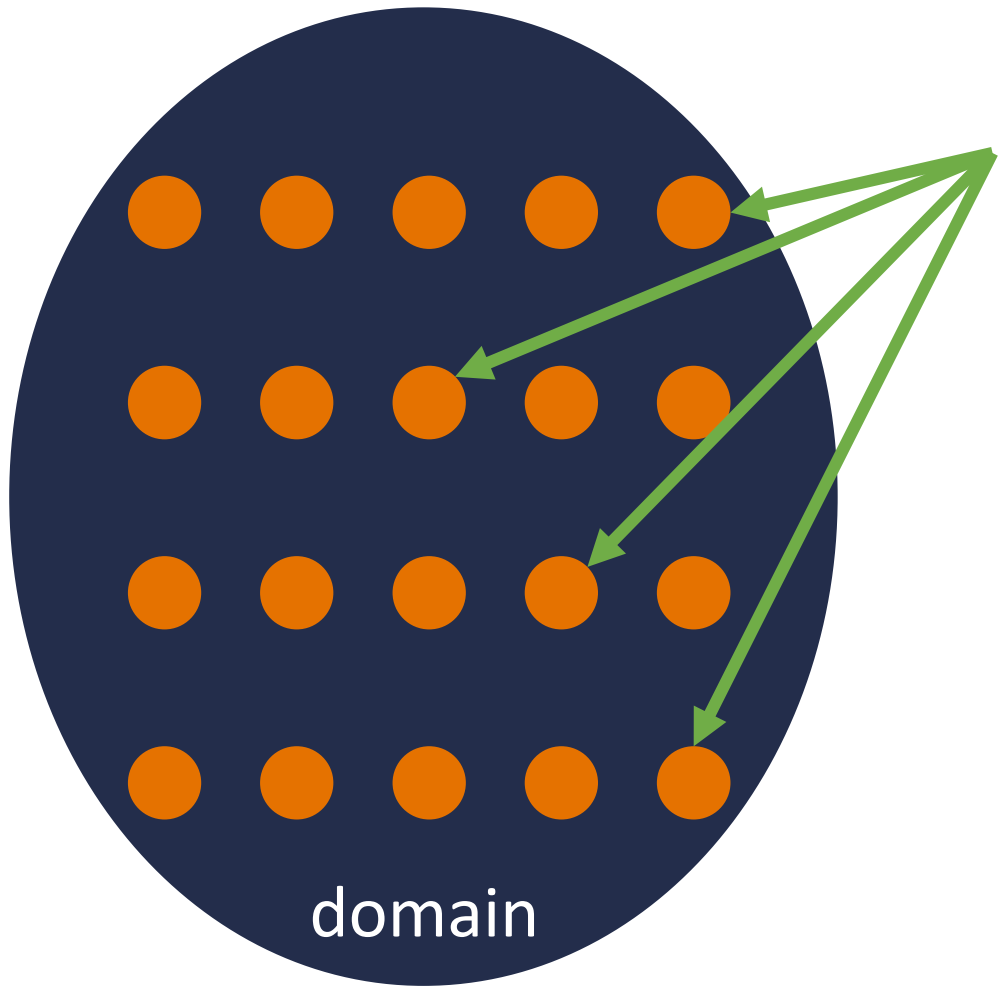
- Symmetric encryption with low-depth decryption
- MACs with low-depth verification
- CCA-secure symmetric encryption with low-depth decryption

Encoded-input PRF: function whose behavior is pseudorandom on a <u>sparse</u> subset of the domain

(F, E) is an encoded-input PRF if $F'(k, x) \coloneqq F\big(k, E(x)\big)$ is a <u>strong</u> PRF

A way to bypass impossibility results for weak/strong PRFs (e.g., can have EI-PRF in complexity class where weak/strong PRFs do not exist)

Encoded-Input PRFs

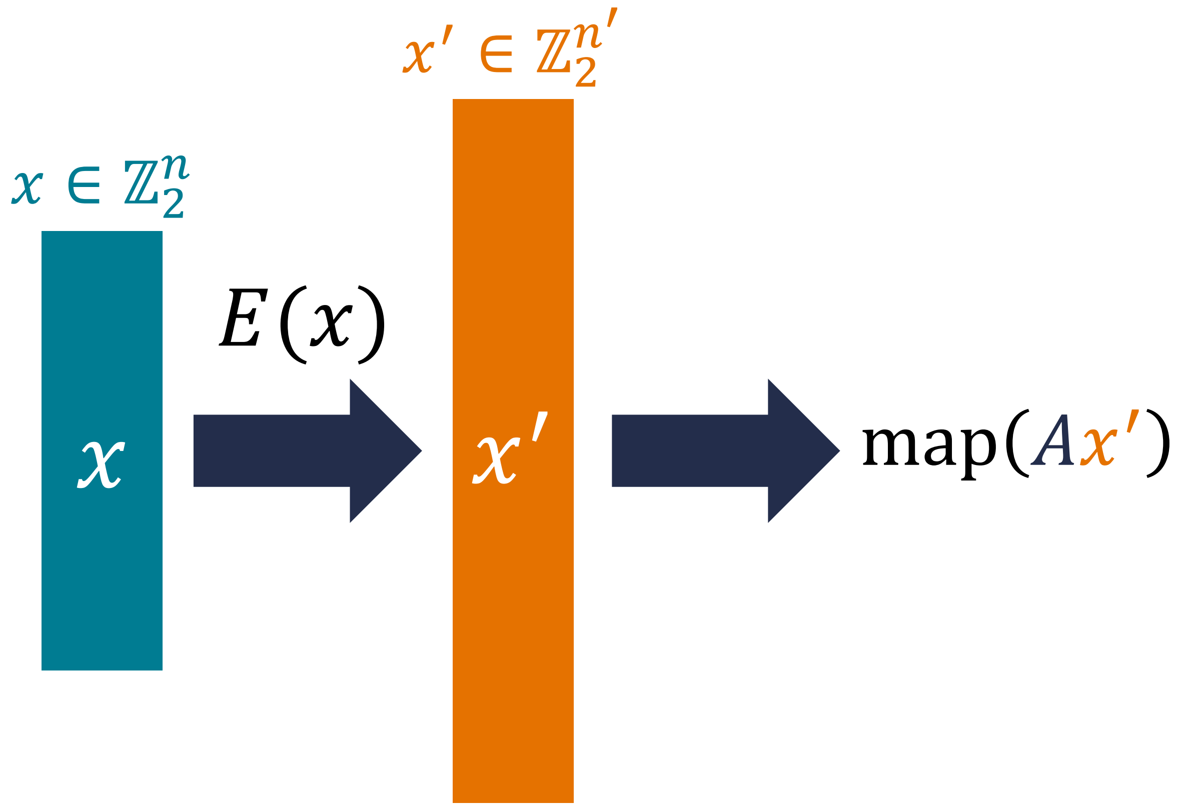


Encoded-input PRF: function whose behavior is pseudorandom on a sparse subset of the domain

(F, E) is an encoded-input PRF if $F'(k, x) \coloneqq F\big(k, E(x)\big)$ is a strong PRF

Concrete proposal: take encoding function to be encoding algorithm of a linear error-correcting code

Encoded-Input PRFs



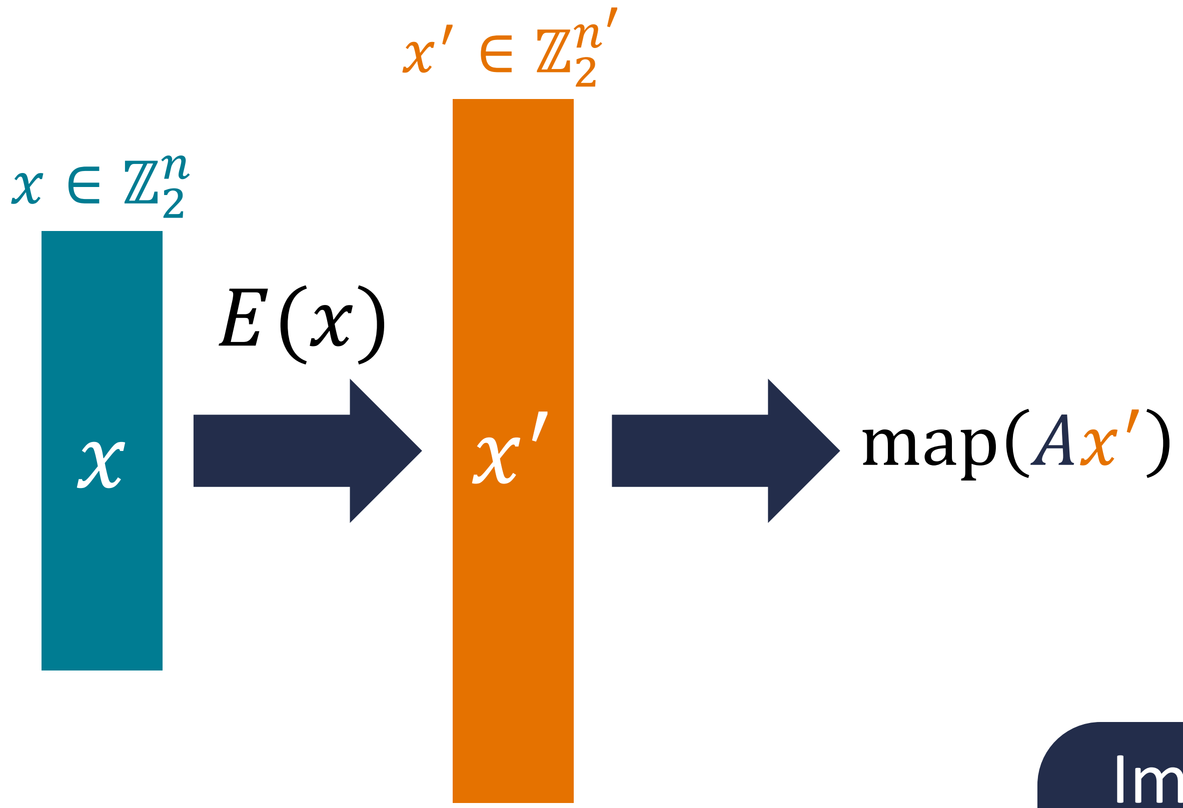
Encoding is done using a linear ECC over $\mathbb{Z}_3$ and taking the binary decomposition

Encoded-input PRF: function whose behavior is pseudorandom on a sparse subset of the domain

(F, E) is an encoded-input PRF if $F'(k, x) \coloneqq F\big(k, E(x)\big)$ is a strong PRF

Concrete proposal: take encoding function to be encoding algorithm of a linear error-correcting code

Encoded-Input PRFs



Encoded-input PRF: function whose behavior is pseudorandom on a sparse subset of the domain

(F, E) is an encoded-input PRF if $F'(k, x) := F\big(k, E(x)\big)$ is a strong PRF

Encoding is done using a linear ECC over $\mathbb{Z}_3$ and taking the binary decomposition

Important to consider ECC over $\mathbb{Z}_3$ and not $\mathbb{Z}_2$ since otherwise, encoding and multiplication by secret key A can be combined (again relies on modulus mixing!)

Encoded-Input PRFs and Strong PRFs

$\boldsymbol{A} \in \mathbb{Z}_2^{m\times m}$ $\boldsymbol{G} \in \mathbb{Z}_3^{m\times n}$ $x \in \{0,1\}^n$

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}\left(\boxed{\boldsymbol{A}} \times \mathrm{BinaryDec}\left(\boxed{\boldsymbol{G}}\ \boxed{x} \right) \right)$$

Secret linear mapping

Public encoding procedure

Conjecture: F_A is a strong PRF (when considering the composition of encoding with weak PRF)

Encoded-Input PRFs and Strong PRFs

$\boldsymbol{A} \in \mathbb{Z}_2^{m\times m}$ $\boldsymbol{G} \in \mathbb{Z}_3^{m\times n}$ $x \in \{0,1\}^n$

$$F_{\boldsymbol{A}}(x) \coloneqq \mathrm{map}\Big(\boldsymbol{A} \times \mathrm{BinaryDec}\big(\boldsymbol{G}\, x\big)\Big)$$

First candidate strong PRF in depth-3 ACC^0 (and even has plausible exponential security)

Conjecture: F_A is a strong PRF (when considering the composition of encoding with weak PRF)

Asymptotically-Optimal Strong PRFs

Does there exist strong PRFs with <u>exponential security</u> that can be computed by <u>linear-size</u> circuits?

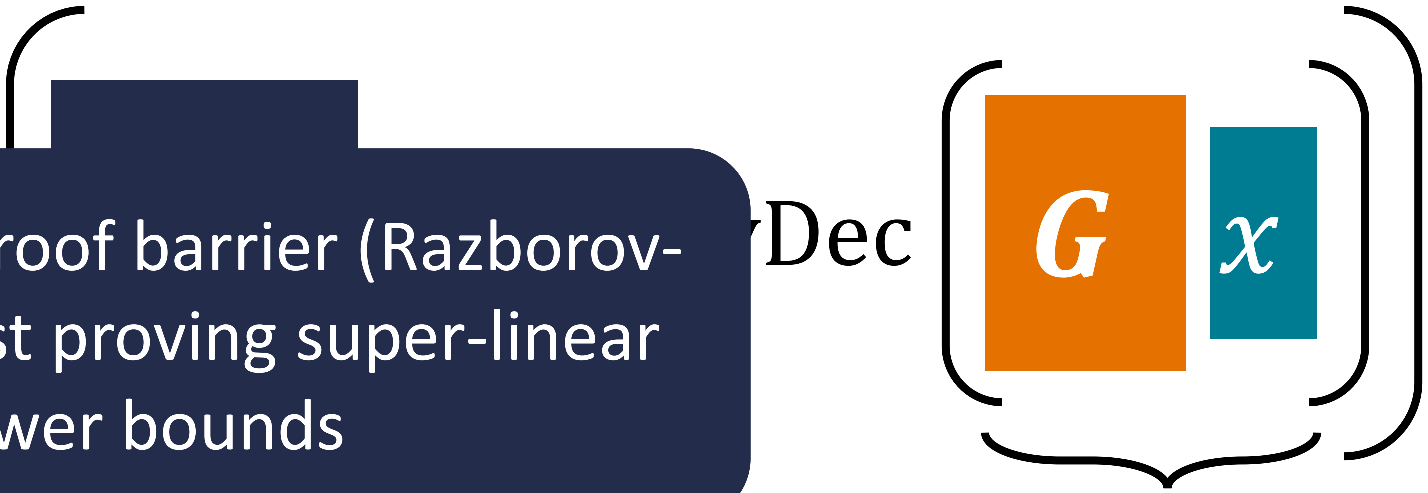
$$F_{\boldsymbol{A}}(x) \coloneqq \text{map}\left(\boxed{\boldsymbol{A}} \times \text{BinaryDec}\left(\underbrace{\boxed{\boldsymbol{G}} \; \boxed{x}} \right) \right)$$

Resulting construction can be implemented by a *<u>linear-size</u>* ACC^0 circuit

Can instantiate with linear-time encodable codes (e.g., IKOS / Druk-Ishai family)

Asymptotically-Optimal Strong PRFs

Does there exist strong PRFs with <u>exponential security</u> that can be computed by <u>linear-size</u> circuits?



Gives new natural proof barrier (Razborov-Rudich style) against proving super-linear circuit lower bounds

Resulting construction can be implemented by a *<u>linear-size</u>* ACC^0 circuit

Can instantiate with linear-time encodable codes (e.g., IKOS / Druk-Ishai family)

Conclusions

$$F_{A}(x) \coloneqq \mathrm{map}(Ax) \text{ where } A \in \mathbb{Z}_2^{n\times n}$$

"secret matrix-vector product over $\mathbb{Z}_2$, sum resulting values mod 3"

Modulus mixing is a relatively unexplored source of hardness:

- Enables new and <u>simple</u> cryptographic primitives (e.g., weak PRF candidate in depth-2 ACC^0, strong PRF candidate in depth-3 ACC^0)
- Assumptions have numerous connections to problems in complexity theory, learning theory, mathematics

Open Questions and Future Directions

Building other cryptographic primitives (e.g., hash functions, signatures, etc.) from modulus mixing assumptions

- MPC-friendly primitives give natural candidate for *post-quantum* signatures [IKOS07]

Further cryptanalysis of new PRF candidates

More crypto dark matter out there to be explored!

Thank you!