

Computing on Encrypted Data via Secret Dual Codes

Yuval Ishai

Technion and AWS

Fabrice Benhamouda

AWS

Caicai Chen

Bocconi

Shai Halevi

AWS

Hugo Krawczyk

AWS

Tamer Mour

Bocconi

Tal Rabin

AWS

Alon Rosen

Bocconi

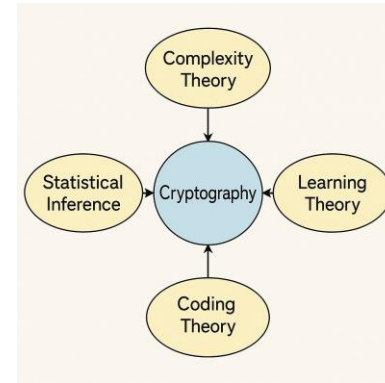
Cryptography 10 Years Later

June 27, 2025

Why New Assumptions?



Diversifying sources of trust



Interactions with other communities



Not previously known from
existing assumptions +
ideal obfuscation

New conclusions!

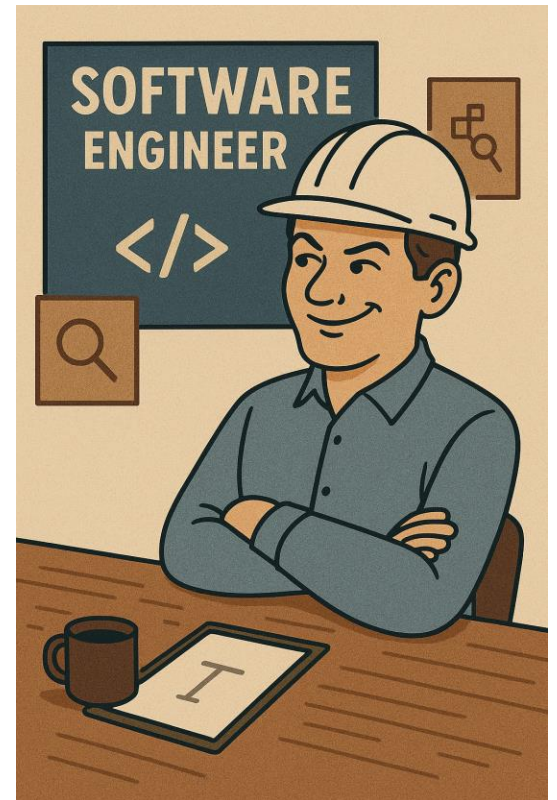
First nontrivial task where
security is “for free”

Motivation: Searching on Encrypted Data



Want a cryptographic solution?

Sure, bring it on!

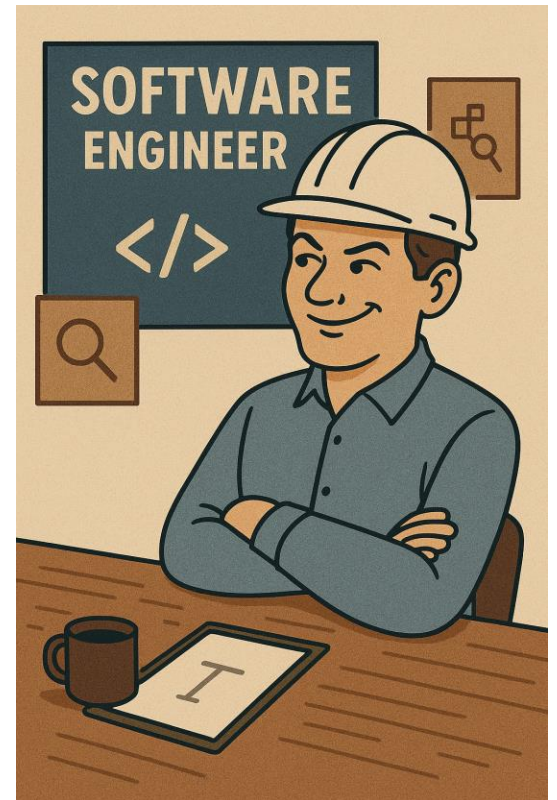


Motivation: Searching on Encrypted Data



How about
Searchable Symmetric Encryption?

Are you kidding me?
It doesn't support fuzzy searches,
and leaks the access pattern.

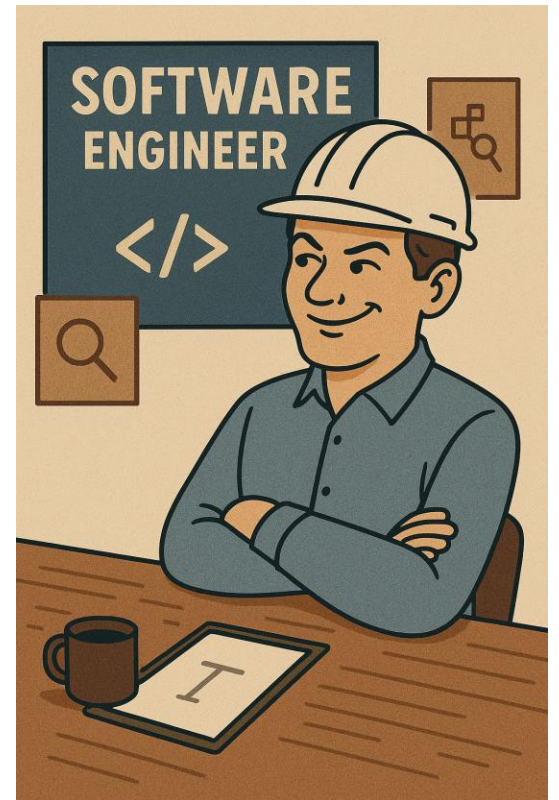


Motivation: Searching on Encrypted Data



Then it looks like you want
Homomorphic Encryption (HE)!

Get real...
My life is too short to wait for this to run.



Motivation: Searching on Encrypted Data



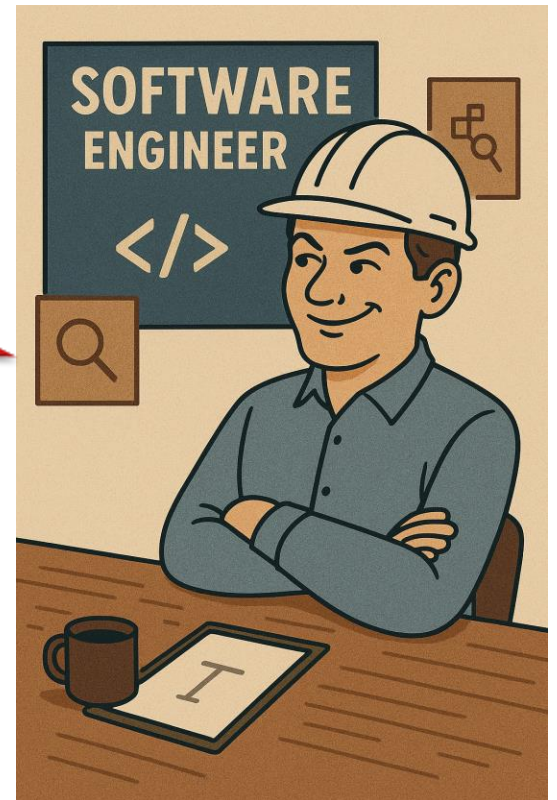
Got it.

So what you probably want is honest-majority 3PC. It's very fast.

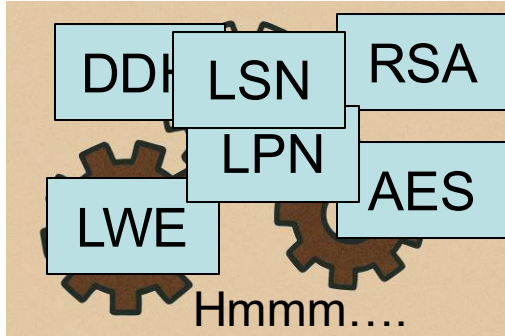
I don't like non-collusion assumptions.

... but you make them all the time, whether you like it or not.

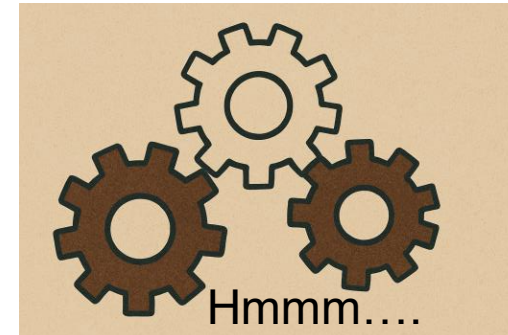
I don't care. Can't make them here.



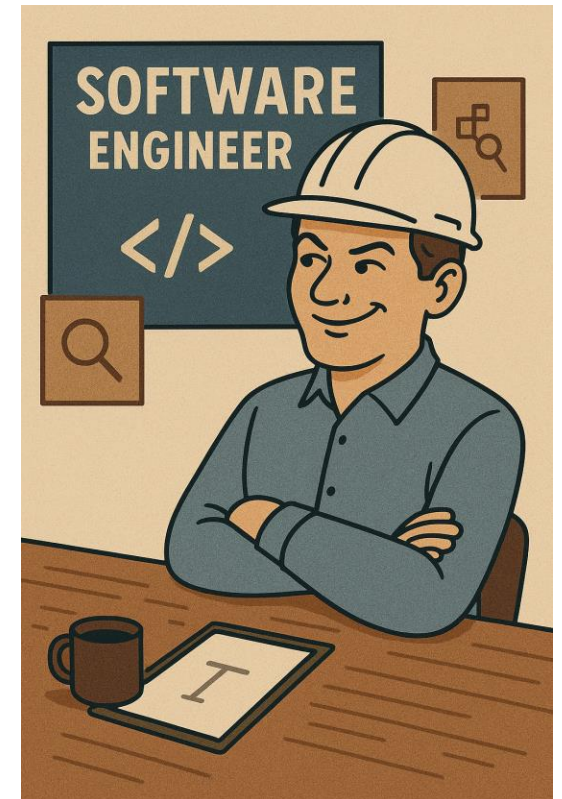
Motivation: Searching on Encrypted Data



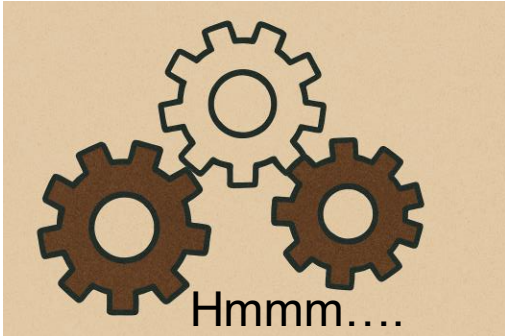
[Excited] **Wait!!!**
I have a solution which is as secure as HE
and even faster than 3PC!



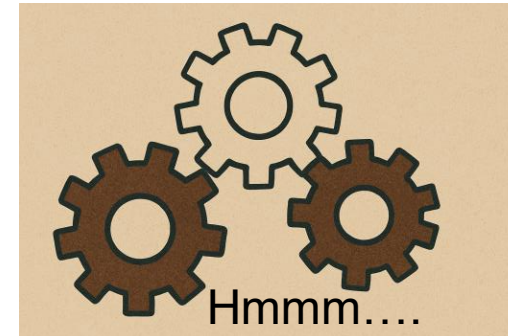
I don't like the color of your shirt.



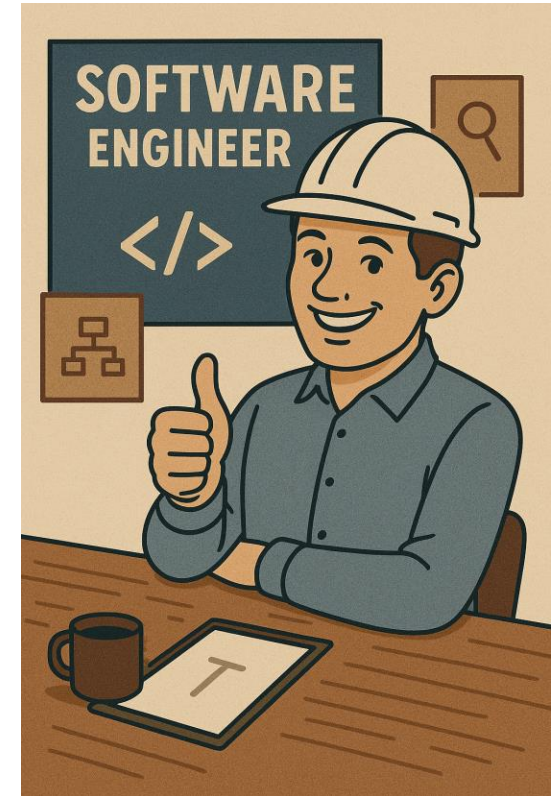
Motivation: Searching on Encrypted Data



[Excited] **Wait!!!**
I have a solution which is as secure as HE
and even faster than 3PC!

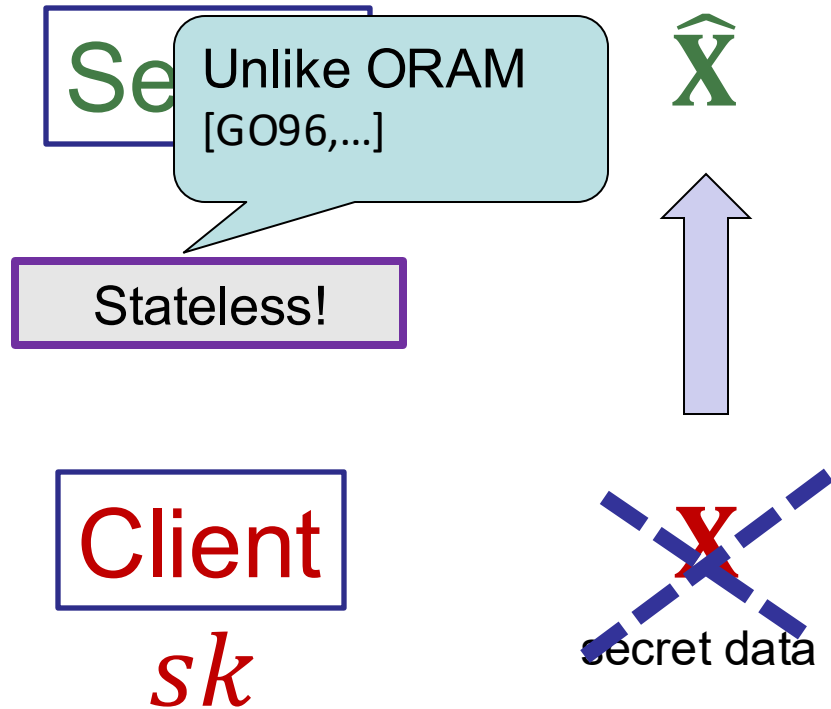


Awesome!
When do we start?



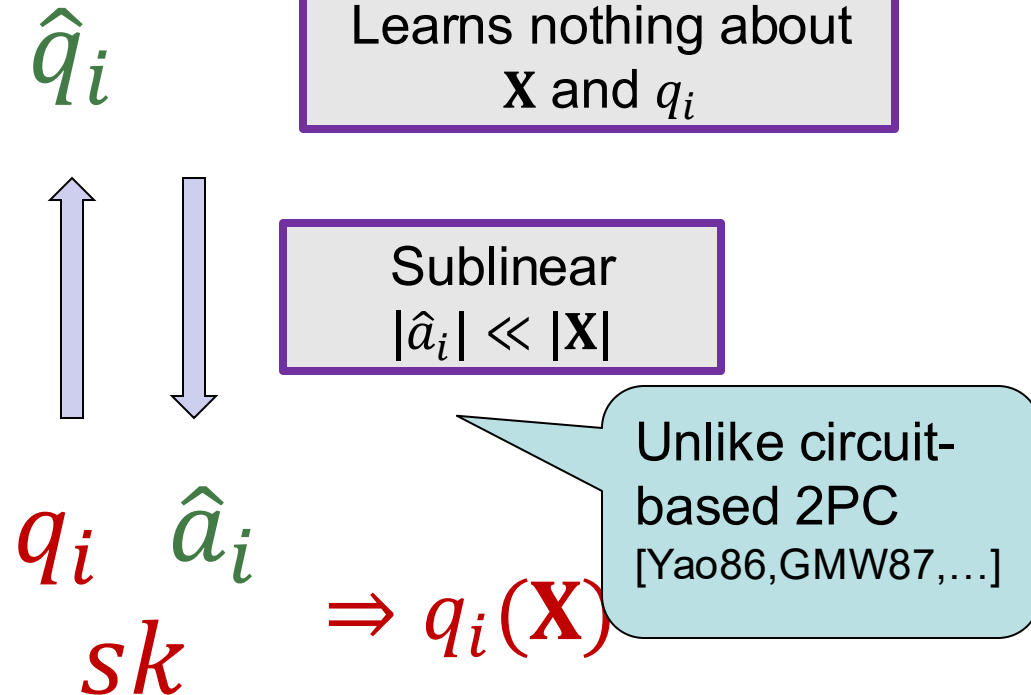
Computing on Encrypted Data

Offline



Online

repeat many times



Functionalities

Weaker than public preprocessing
[Beimel-I-Malkin 00, Lin-Mook-Wichs 23]

- Private Information Retrieval

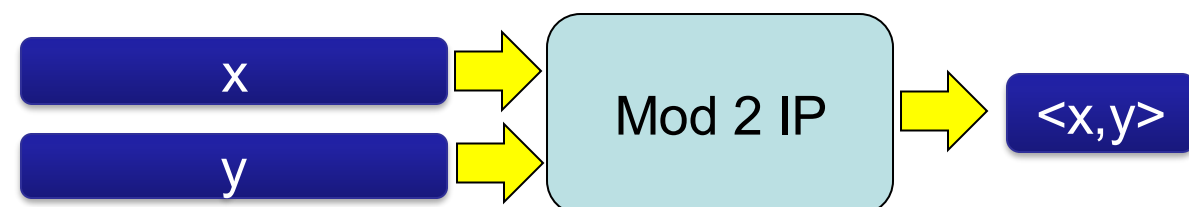
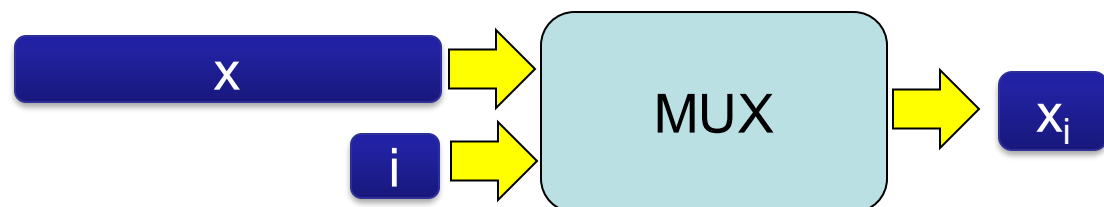
[Chor-Goldreich 98]

- $\mathbf{X} \in \{0,1\}^n$ is a database, $q_i(x) \in [n]$... but similar when client is distributed, e.g., in MPC applications
- “Secret-key PIR with preprocessing”
[Boyle-I-Pass-Wootters 17, Canetti-Holmgren-Richelson 17]
- Typical goal: “doubly efficient” PIR in the RAM model
 - We aim to minimize assumptions and/or efficiency in Boolean circuit model

Is this

so much cheaper than

this?



Functionalities

- Private Information Retrieval (PIR)

[Chor-Goldreich-Kushilevitz-Sudan 95, Kushilevitz-Ostrovsky 97]

- $\mathbf{X} \in \{0,1\}^N$ is a database, $q_i(\mathbf{X}) = \mathbf{X}[q_i]$
- “Secret-key PIR with preprocessing”
[Boyle-I-Pass-Wootters 17, Canetti-Holmgren-Richelson 17]
- Typical goal: “doubly efficient” PIR in the RAM model
 - We aim to minimize assumptions and/or efficiency in Boolean circuit model

- Encrypted Matrix-Vector Product (EMVP)

- $\mathbf{X} \in \mathbb{F}^{m \times w}$ is a matrix, $q_i \in \mathbb{F}^w$ a vector, $q_i(\mathbf{X}) = \mathbf{X}q_i$
- Output is typically short, can be securely post-processed

EMVP for Encrypted Fuzzy Search

Janus: Safe Biometric Deduplication for Humanitarian Aid Distribution

Kasra EdalatNejad*, Wouter Lueks[†], Justinas Sukaitis[‡], Vincent Graf Narbel[‡]

Massimo Marelli[‡], Carmela Troncoso*

**SPRING Lab, EPFL, Lausanne, Switzerland*

{kasra.edalat, carmela.troncoso}@epfl.ch

[†]*CISPA Helmholtz Center for Information Security, Saarbrücken, Germany*

lueks@cispa.de

[‡]*International Committee of the Red Cross, Geneva, Switzerland*

dpo@icrc.org

HE / GC

Large-Scale MPC: Scaling Private Iris Code Uniqueness Checks to Millions of Users

Remco Bloemen², Bryan Gillespie³, Daniel Kales¹, Philipp Sippl² and Roman Walch¹

¹ TACEO, Graz, Austria

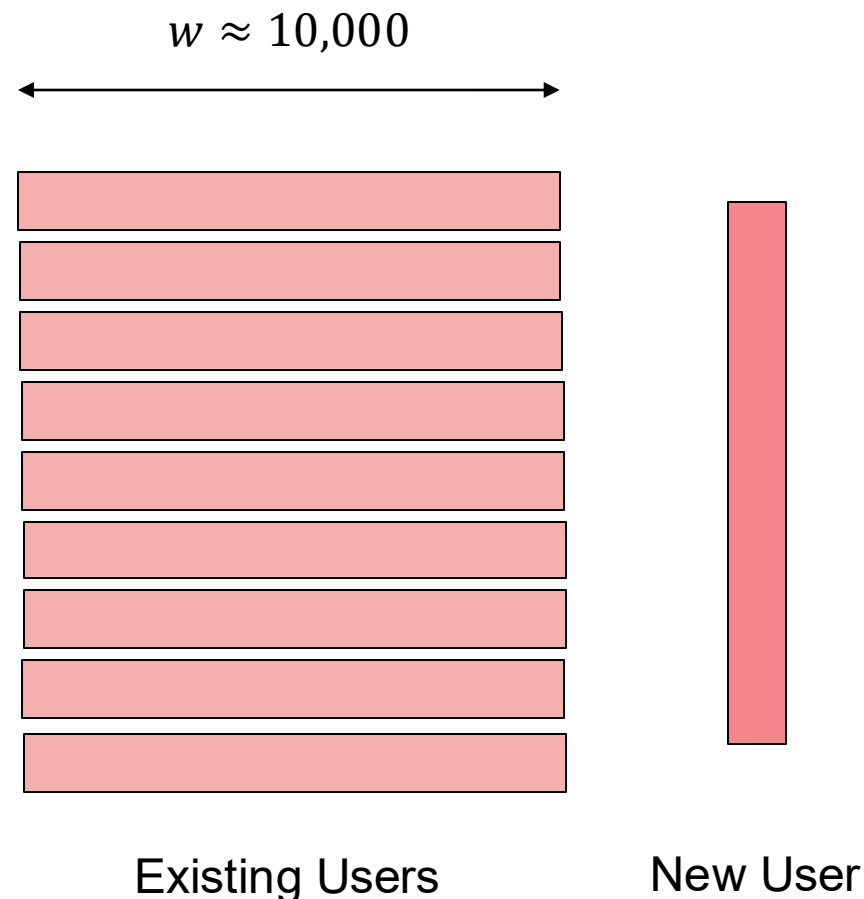
lastname@taceo.io

² Worldcoin Foundation

³ Inversed Tech

bryan@inversed.tech

Much faster via
3PC



Public-Data EMVP for Encrypted Fuzzy Search

Private Web Search with Tiptoe

Alexandra Henzinger
MIT

Emma Dauterman
UC Berkeley

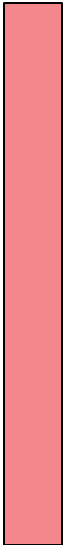
Henry Corrigan-Gibbs
MIT

Nickolai Zeldovich
MIT

Linearly Homomorphic Encryption



Document embeddings



Web search
query

EMVP from Homomorphic Encryption

- Easy from HE for degree-2 polynomials [BGN05, Gen09, ...]
- Possible from linearly homomorphic encryption?
 - Simple for public-data EMVP hiding q_i but not \mathbf{X}
 - Upgrade to EMVP via masking
 - Run public EMVP with public matrix $\mathbf{X}' = \mathbf{X} + \mathbf{R}$
 - Client recovers $\mathbf{X}q = \mathbf{X}'q - \mathbf{R}q$
 - Use trapdoored matrix \mathbf{R} to improve client computation
[Braverman-Newman 25, Vaikuntanathan-Zamir 25]
We propose new TDM constructions with better concrete efficiency

Field-Agnostic EMVP?

- Field-agnostic protocols only make black-box use of \mathbb{F}
 - Typical feature of honest-majority MPC protocols
 - Possible also with no honest majority

[Naor-Pinkas 99, I-Prabhakaran-Sahai 09, Applebaum-Avron-Brzuska 15]
- None of the existing EMVP protocols is field agnostic
 - Lattice-based: sampling noise, rounding, $q \gg p$
 - Inherent to linearly homomorphic encryption [AAB15]
- Advantages of field-agnostic protocols
 - Concrete efficiency
 - Client can be easily distributed

Our Results: EMVP

[BCHIKMRR25]

- Feasibility result for field-agnostic EMVP
 - Assuming LPN over general \mathbb{F}
 - In fact, even in parameter regimes not known to imply PKE or CRH
 - Everything optimal up to polylog factors
 - ... but not concretely efficient
- Concretely efficient EMVP under new assumptions
 - Variants of “Learning Subspace with Noise” [Dodis-Kalai-Lovett 09]
 - Can get $< 1.1x$ cleartext costs for realistic matrix sizes
 - Based on our cryptanalysis

Our Results: Secret-Key PIR

[CIMR25]

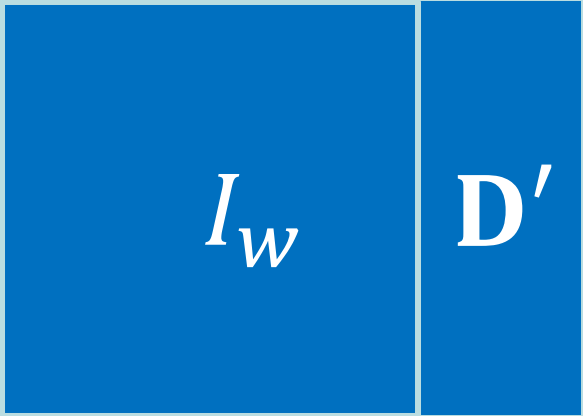
- Feasibility of low-communication sk-PIR from LPN
 - $\approx N^\epsilon$ communication
 - Strong evidence that public-key cryptography is **not** needed
 - Doubly efficient PIR impossible from black-box OWF [Lin-Mook-Wichs 25]
- Minimizing computation in Boolean circuit model
 - Server implemented by circuit of size $\approx 4N$
 - Much better than previous approaches, including heuristics
 - Downside: only slightly sublinear in the RAM model
 - Based on a new variant of LSN

Technical Approach: Secret Dual Codes

- Pseudorandom codes over \mathbb{F} determined by secret key sk
- Optionally: structured codes for better efficiency


Data Code

$$D = \text{span } \mathbf{D}$$

$$n = w + k$$


I_w \mathbf{D}' w

Query Code $\mathcal{C} = D^\perp = \text{span } \mathbf{C}$

$$n$$


\mathbf{C} k

EMVP Protocol

Offline:

$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{D} + \mathbf{R}$$

Pseudorandom matrix determined by sk

$$\mathbf{D} =$$

$$n = w + k$$

$$\begin{bmatrix} I_w & \mathbf{D}' \end{bmatrix} \quad w$$

$$n$$

$$\mathbf{C} \quad k$$

\mathbf{X}

\mathbf{X}'

EMVP Protocol

Offline:

$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{D} + \mathbf{R}$$

$$\mathbf{D} = \begin{array}{|c|c|} \hline & \\ \hline I_w & \mathbf{D}' \\ \hline \end{array} \begin{array}{l} n = w + k \\ w \end{array}$$

Online:

$$\tilde{q}_i = \begin{pmatrix} q_i \\ 0^k \end{pmatrix} + c_i$$

Random codeword in \mathcal{C}

$$\uparrow \tilde{q}_i \quad \downarrow \hat{\mathbf{X}}\tilde{q}_i$$

$$\Rightarrow \mathbf{X}q_i = \hat{\mathbf{X}}\tilde{q}_i - \mathbf{R}\tilde{q}_i$$

$$\begin{array}{|c|} \hline \mathbf{C} \\ \hline \end{array} \begin{array}{l} n \\ k \end{array}$$

Efficiency: Storage and Communication

Offline:

Storage overhead: $1 + k/w$

$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{D} + \mathbf{R}$$

$$\mathbf{D} = \begin{bmatrix} I_w & \mathbf{D}' \end{bmatrix} \quad w$$

Online:

Upload overhead: $1 + k/w$

$$\tilde{q}_i = \begin{pmatrix} q_i \\ 0^k \end{pmatrix} + c_i$$

$$\uparrow \tilde{q}_i \quad \downarrow \hat{\mathbf{X}}\tilde{q}_i$$

No download overhead...

$$\Rightarrow \mathbf{X}q_i = \hat{\mathbf{X}}\tilde{q}_i - \mathbf{R}\tilde{q}_i$$

$$\begin{matrix} n = w + k \\ \begin{bmatrix} I_w & \mathbf{D}' \end{bmatrix} & w \\ \begin{bmatrix} \mathbf{C} \end{bmatrix} & k \end{matrix}$$

Efficiency: Online Computation

Offline:

$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{D} + \mathbf{R}$$

$$\mathbf{D} = \begin{array}{c|c} & \\ \hline & \end{array} \begin{array}{c} n = w + k \\ I_w \quad \mathbf{D}' \end{array} \begin{array}{c} w \\ \end{array}$$

Online:

$$\tilde{q}_i = \begin{pmatrix} q_i \\ 0^k \end{pmatrix} + c_i$$

$$\uparrow \tilde{q}_i \quad \downarrow \hat{\mathbf{X}}\tilde{q}_i \quad \text{1 + } k/w$$

$$\Rightarrow \mathbf{X}q_i = \hat{\mathbf{X}}\tilde{q}_i - \mathbf{R}\tilde{q}_i \quad \text{Use TDM!}$$

$$\begin{array}{c} n \\ \mathbf{C} \end{array} \quad k$$

Security?

Offline:

$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{D} + \mathbf{R}$$

$$\mathbf{D} =$$

$$\begin{array}{|c|c|} \hline I_w & \mathbf{D}' \\ \hline \end{array} \quad \begin{array}{l} n = w + k \\ w \end{array}$$

Online:

q_i are the same $\Rightarrow \text{rank}(\tilde{q}_i) \leq k + 1$
 q_i are random $\Rightarrow \text{rank}(\tilde{q}_i) \cong n$

$$\tilde{q}_i = \begin{pmatrix} q_i \\ 0^k \end{pmatrix} + \dots + c_i$$

$$\Uparrow \tilde{q}_i \Downarrow \hat{\mathbf{X}}\tilde{q}_i$$

$$\Rightarrow \mathbf{X}q_i = \hat{\mathbf{X}}\tilde{q}_i - \mathbf{R}\tilde{q}_i$$

Want:
pseudorandom \tilde{q}_i

$$\begin{array}{|c|} \hline \mathbf{C} \\ \hline \end{array} \quad \begin{array}{l} n \\ k \end{array}$$

EMVP Protocol: For Real

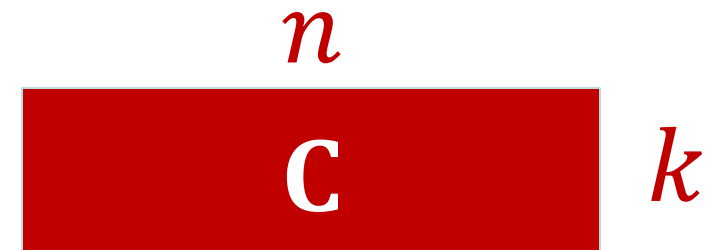
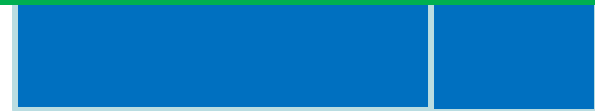
- **Idea:** Make \tilde{q}_i pseudorandom by adding “noise”
- **Standard LPN or LWE noise hurts correctness**
- **Instead:**
 - **Mixture noise:** w/prob $\mu < 1$, replace \tilde{q}_i by a sample from a different distribution (e.g., uniform)
Repeat/encode to correct failures
 - **Planting noise:** reveal a random low-dimensional subspace $\hat{\mathbf{Q}}_i$ containing \tilde{q}_i

Online: // with planting noise

$$\tilde{q}_i = \begin{pmatrix} q_i \\ 0^k \end{pmatrix} + c_i$$

$\uparrow \hat{\mathbf{Q}}_i \downarrow \hat{\mathbf{X}}\hat{\mathbf{Q}}_i$

dim($\hat{\mathbf{Q}}_i$) overhead
... mitigated later



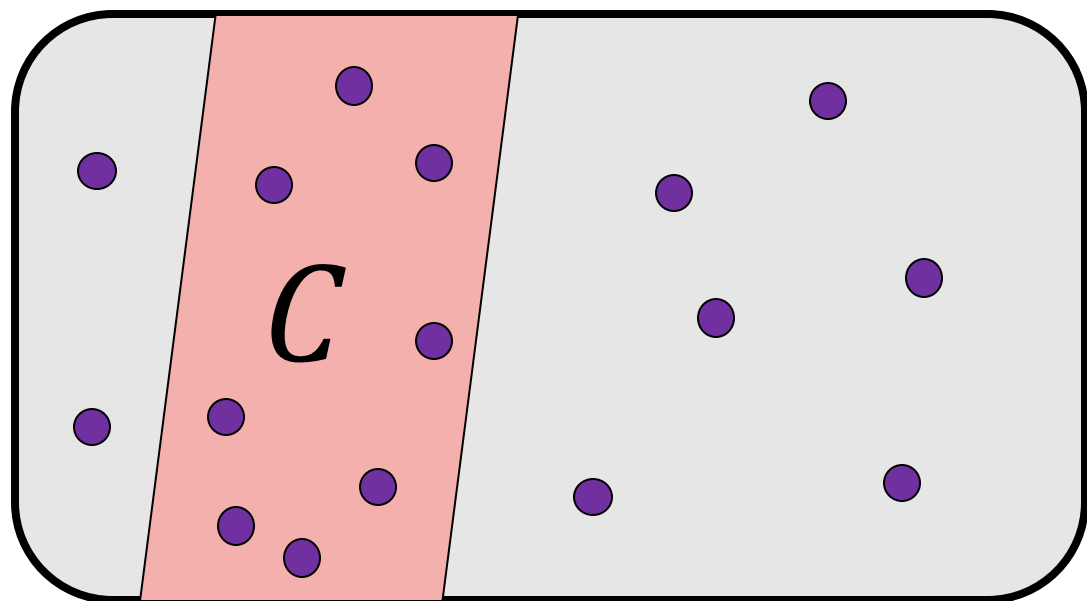
$\Rightarrow \hat{\mathbf{X}}\tilde{q}_i$ spanned by columns of $\hat{\mathbf{X}}\hat{\mathbf{Q}}_i$

Learning Subspace with Noise

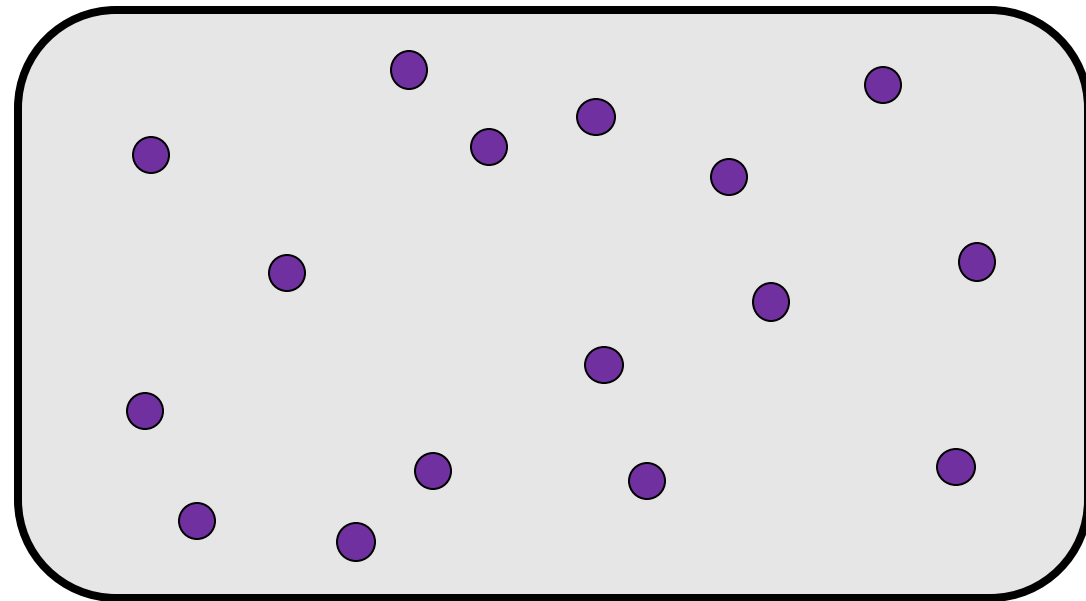
[Dodis-Kalai-Lovett 09]

secret k -dimensional
subspace

$U(C)$ μ -mixed with $U(\mathbb{F}^n)$



\approx



\mathbb{F}^n

\mathbb{F}^n

LSN: History

- First applied for leakage-resilient cryptography [DKL09]
 - Similar applications from standard LPN [Yu-Zhang 16]
- Search-LSN studied by learning theory community [Chen-De-Vijayaraghavan 21]
 - Instance of learning mixtures of linear subspaces
- Structured variants implicit in doubly-efficient secret-key PIR [Boyle-I-Pass-Wootters 17, Canetti-Holmgren-Richelson 17]
 - \mathcal{C} = secretly permuted Reed-Muller code
 - Samples contain low-weight codewords
 - Noise hides nonzero values \approx split LSN noise
 - So far withstood analysis [BHW19, BW21, BHMW21]

LSN: Known Attacks

- LSN is equivalent to LPN when \mathbb{F} is small and $k = n - 1$ [DKL09,CDV21]
- LSN in constant-rate regime:
 - Implies LPN [DKL09,CIMR25]
 - Idea: use dual-LPN search oracle to find sparse linear dependence of samples
 - Better attacks on LSN are known [Raz09,DKL09,CDV21,CIMR25]
 - Poly-time attack with constant noise rate $\mu < 1$
 - Quasipoly-time attack when $\mu = 1 - 1/k^\varepsilon$
- Our LSN conjecture: security when rate is constant and $\mu = 1 - o(1)$
 - Similar conjecture in [DKL09] for $\mu = 1 - 1/k^\varepsilon$

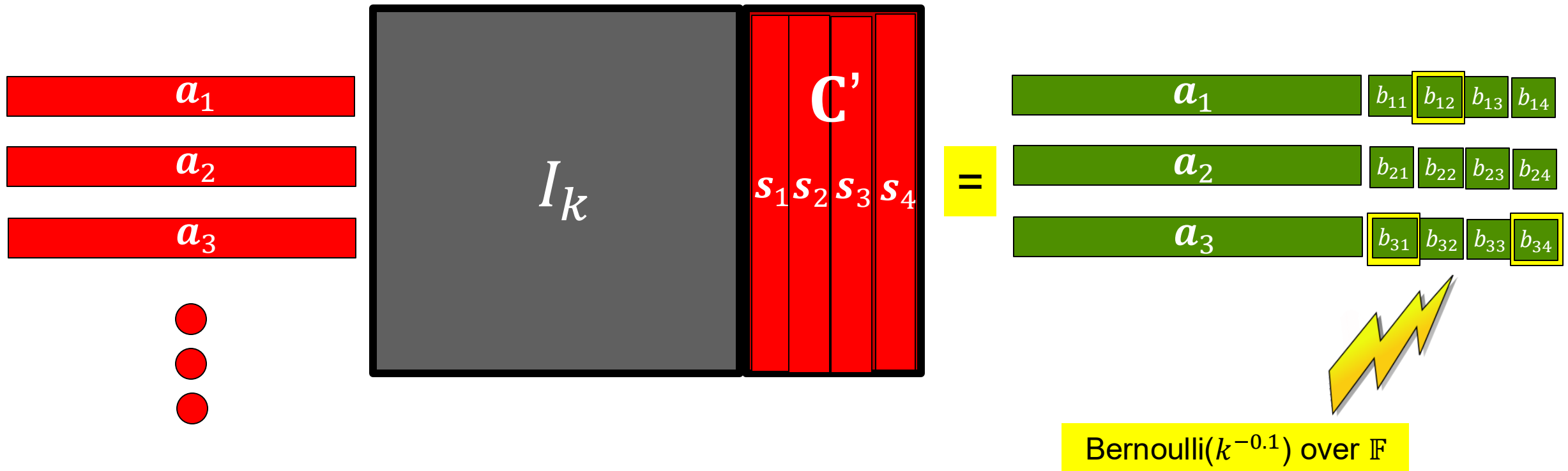
Algebraic Attack on LSN

[Raz 09]

- Simple rank attack
 - Fix code rate $\rho = k/n = 1/2$ and noise rate $\mu = 1/3$
 - In random experiment, expect first linear dependence after $\approx n$ samples
 - In pseudorandom experiment, after at most $\approx (3/4)n$ samples
 - Avoided by using noise rate $\mu > 1/2$
- Reducing code rate via tensoring
 - The code $\mathcal{C}^d \subseteq \mathbb{F}^{n^d}$ spanned by the d -tensors of $c \in \mathcal{C}$ has rate ρ^d
 - For any constant $\mu < 1$ choose constant d for which $\rho^d < 1 - \mu$
 - Apply rank distinguisher to tensored samples
- Equivalently: find degree- d polynomial $p \neq 0$ vanishing on all samples

LPN \Rightarrow high-rate LSN with mixture noise

$$n = k + k^{0.1}$$



- Each row is noiseless w/prob $1/e$
- Secure under LPN with noise rate $k^{-0.1}$

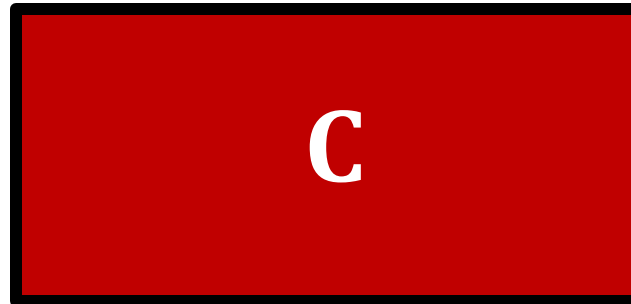
Regular LSN

$$n = 3k$$

r_1

r_2

r_3



=

c_1

c_2

c_3



Plant in d -dimensional (affine) space

Algebraic attacks require time $\approx k^d$

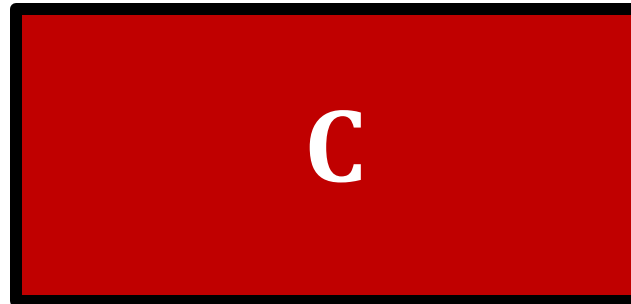
Split LSN

$$n = 3k$$

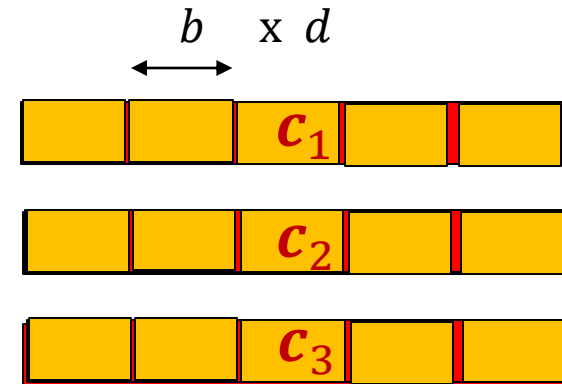
r_1

r_2

r_3



=



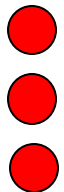
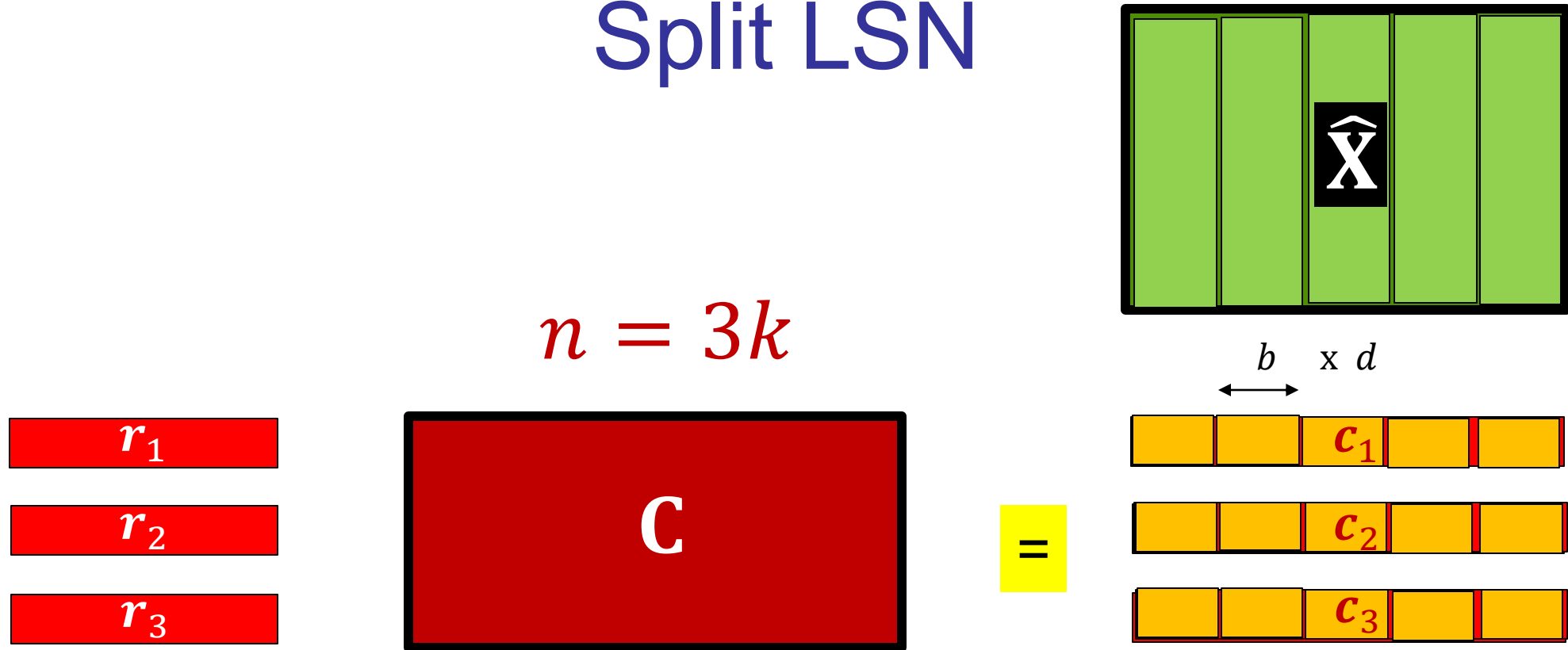
- Planting in a d -dimensional product space
- More structured assumption



Multiply each size- b block by random $\alpha_i \in \mathbb{F}^*$

Algebraic attacks require time $\approx b^{k/b}$

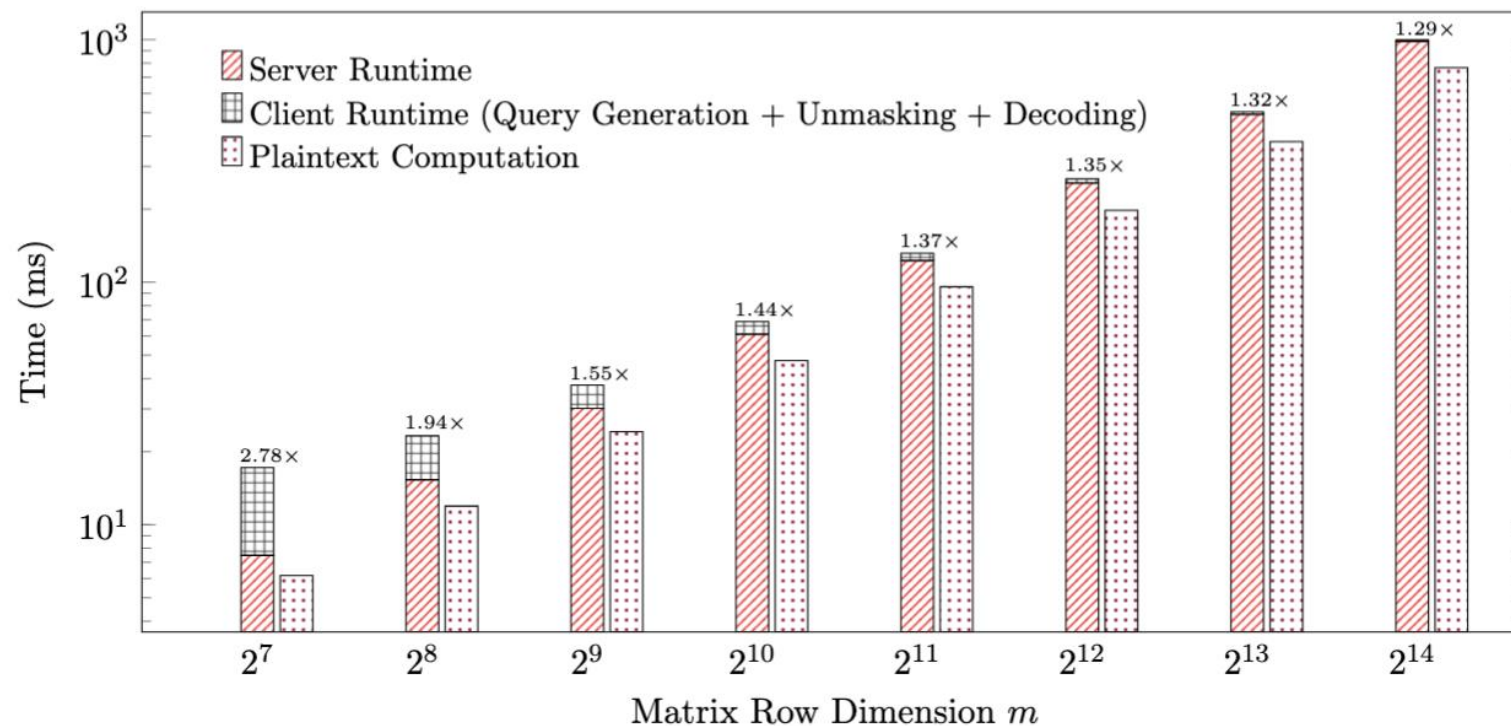
Split LSN



- Avoids factor- d overhead in upload and server computation
- Still factor- d overhead in download
- ... but can be mitigated by composing with LHE

Concrete Analysis and Benchmarks

32-bit prime field, $(w, k, b) = (10000, 2600, 140)$



LSN-Based Secret-Key PIR

- EMVP implies PIR with $\approx N^{0.5}$ communication
 - Improve to $\approx N^\epsilon$ via folding
- New features
 - **Feasibility**: Based on LPN with parameters not known to imply PKE/CRH
 - **Efficiency**: Server Boolean circuit size $\approx 4N$, under binary variant of split-LSN
- Client's computation is sublinear, but far from optimal
 - Can be improved by composing with any standard (single-server) PIR

Conclusion

- New technique for computing on encrypted data
 - Use a secret code to encrypt the data and its dual to encrypt queries
 - Combines advantages of HE and 3PC
 - Lightweight clients that are easy to distribute
- First-of-their-kind feasibility results from standard LPN
 - Field-agnostic sublinear secure computation
 - Secret-key PIR without public-key cryptography
- Security (essentially) for free!
 - Based on new LSN-style assumptions

Further Directions

- Minimal assumption for sk-PIR / EMVP
 - Are one-way functions sufficient?
- LSN assumptions
 - Search-to-decision reductions?
 - More relations between LSN variants and LPN
 - Relation with permuted RM codes assumption
 - More concrete analysis, also over non-field rings
- Optimize for applications
 - Post-processing, distributed clients, distributed setup, ...