Worst-Case Hardness for LPN
(and Cryptographic Hashing)
via Code Smoothing

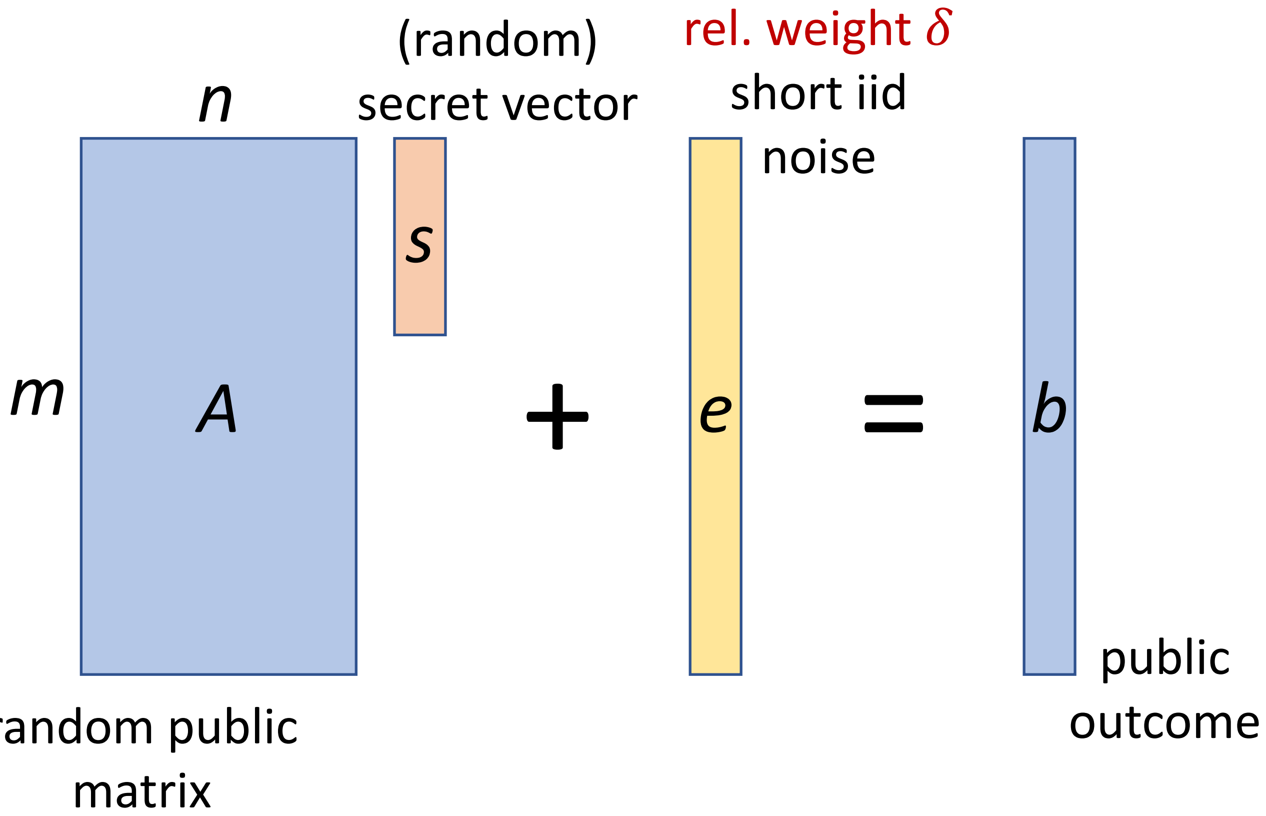
Zvika Brakerski — Weizmann

Vadim Lyubashevsky — IBM

Vinod Vaikuntanathan — MIT

Daniel Wichs — Northeastern

Learning Parity with Noise (LPN) [BFKL93]



Arithmetics modulo 2

Goal: $(A,b) \Rightarrow s$

Problem gets easy as noise gets sparse:

Solvable w.p. $e^{-\delta n}$ => poly. if $\delta = \frac{\log n}{n}$.

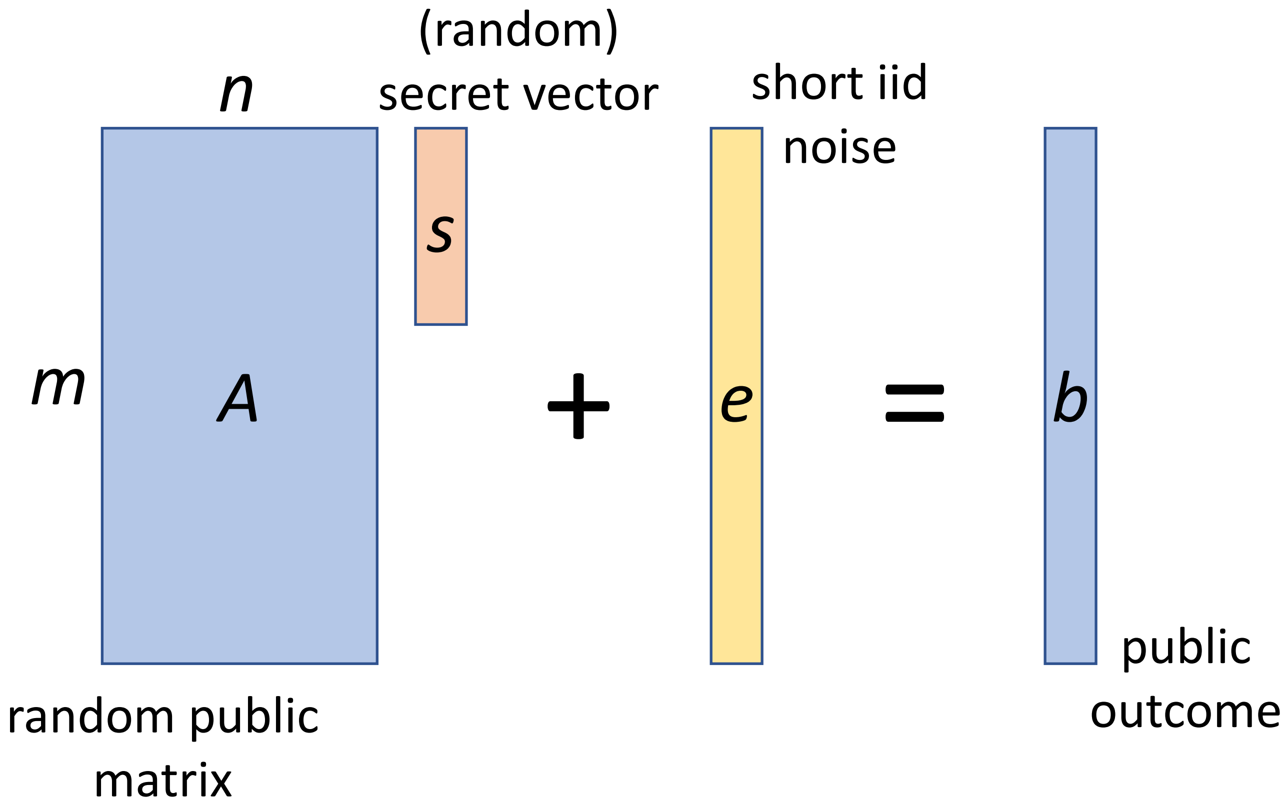
"Barely hard" LPN: $\delta = \frac{\log^2 n}{n}$.

Impossible w.p. $> (1 - 2\delta) \cdot m$

=> negl. if $\delta = \frac{1}{2} - n^{-\omega(1)}$.

"Super hard" LPN: $\delta = \frac{1}{2} - \frac{1}{\text{poly}(n)}$.

Learning with Errors (LWE) [R05]



Arithmetics modulo q>n, Gaussian noise

Goal: $(A,b) \Rightarrow s$

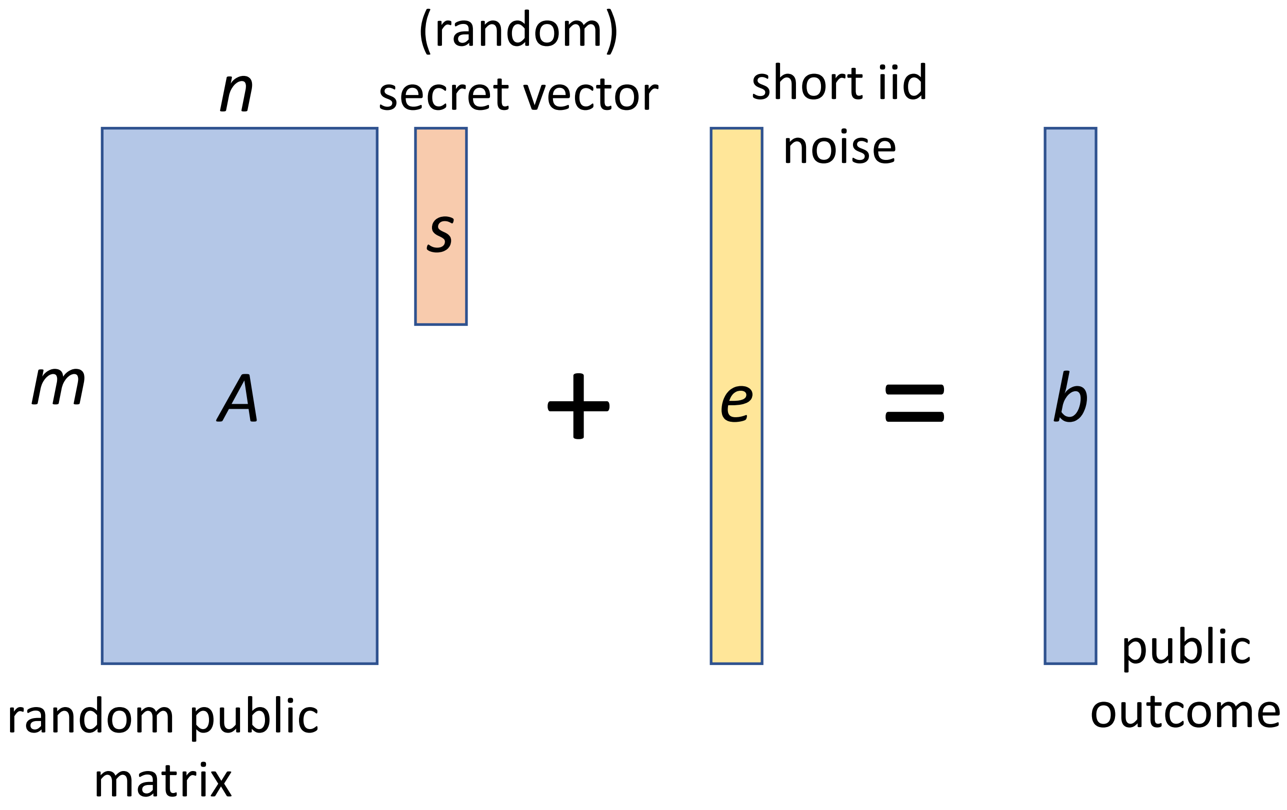
Known properties:

- Worst-case to average-case reduction.
- Contained in SZK (for useful params).

Known applications:

- Symmetric / Public-Key Encryption
- Collision resistant hash (CRH).
- Homomorphic Encryption.
- Attribute-Based Encryption.
- Non-Interactive Zero-Knowledge.

Learning Parity with Noise (LPN) [BFKL93]



Arithmetics modulo 2, Bernoulli noise

Goal: $(A,b) \Rightarrow s$

Known properties:

Known applications:

- Symmetric / Public-Key Encryption

Why so different?

Our Results

New properties:

- Worst-case to average-case reduction.

  LPN ≡ Average-case “Nearest Codeword Problem” (NCP).

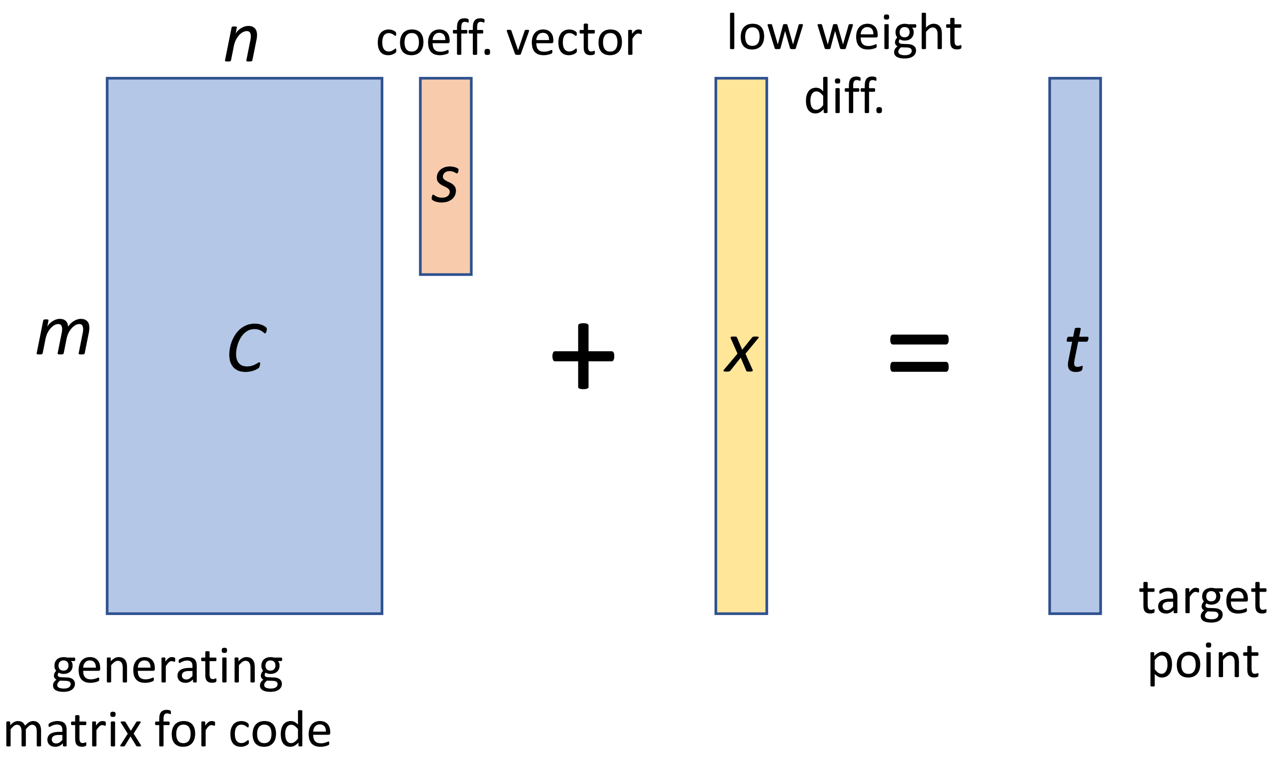
  We show: ”super hard” LPN is harder than “barely hard” (worst-case) NCP.

- “Barely hard” LPN/NCP contained in SZK.

New applications:

- Collision resistant hashing based on “barely hard” LPN (concurrently with [YZWGL17]).
- Follow-up works [BLSV18] extend to IBE, leakage resilience, KDM security (via laconic OT).

Nearest Codeword Problem (NCP)



Arithmetics modulo 2

Goal: (C,*t*) => *s*

Same as LPN except (*C*, *x*) are arbitrary.

NP-Hard in the worst case [ABSS93,DMS99].

We require *C* is *balanced*.

“Barely hard” and “super hard” regimes as in LPN.

Our Technique: Smoothing

[L05]: Solver for LPN with $m = n^{100}$, rel. weight δ'

=> Solver for LPN with $m = n^2$, rel. weight $\delta \ll \delta'$

Idea: Random matrix = *extractor.* Use to *rerandomize*.

Q: What is the min-weight to get entropy *n*? **A:** $\frac{n}{\log n}$ suffices!

The reduction: Generate n^{100} vectors *r*

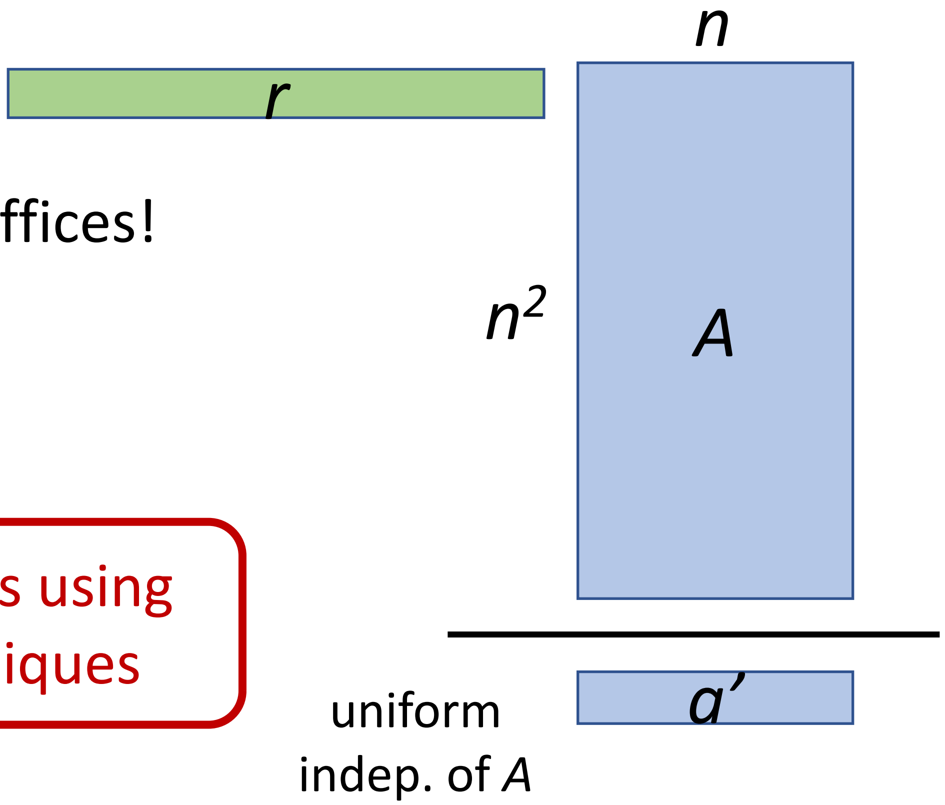
For each compute (*a'*,*b'*)=(*rA*, *rb*).

$\delta = \frac{\log^2 n}{n} \rightarrow \delta' = \frac{1}{2} - \frac{1}{poly(n)}$

CRH follows using std. techniques

Weight of … **gh**.

$1 - 2\delta' = (1 - 2\delta)^{n/\log n}$

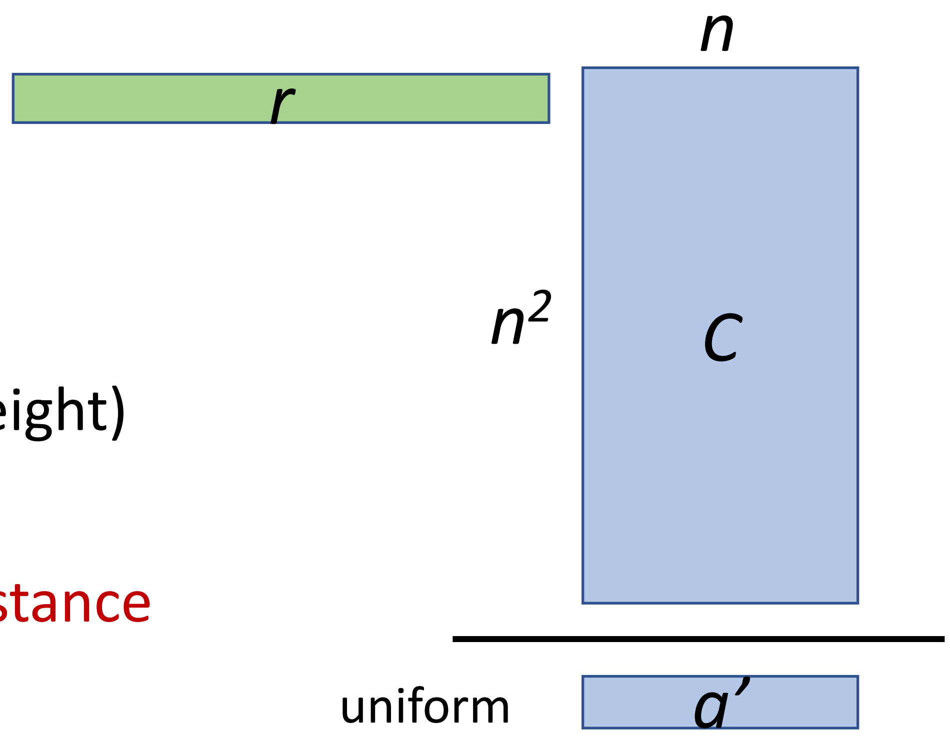


Our Technique: Smoothing

Apply technique to arbitrary (balanced) C ? Arbitrary C cannot be extractor.

Observation: Entropy extraction not needed, only extract from specific **smoothing** dist.

Can show this using harmonic analysis / linear distinguishers / Vazirani XOR lemma.



Note: We get $b' = rt = rCs+rx = a's+e'$

Need to argue that e' is indep. of a' (not just low weight)

Plug in barely hard NCP instance to get super hard LPN instance

=> Worst-case to average case reduction.

Connection to LWE / Lattices

Our technique is analogous to the concept of smoothing in the lattice world.

A distribution is smoothing for a lattice, if modulo the lattice it is uniform

<=> if its product with the dual bases is uniform (over cosets)

Usually in lattice literature: Smoothing using Discrete Gaussians,

in this work we extend the notion of smoothing beyond Gaussians.

Open Problems

Extend the params of our reduction.

Lower bound on smoothing? Non-trivial smoothing for unbalanced codes?

Is “barely hard” LPN/NCP actually not solvable in poly-time? Does balance help?

Construct more cryptography from LPN.

Vinod’s Question: Is there a CRH candidate that is provably not in SZK?

Thank you

European Research Council
Established by the European Commission

Research is supported by EU Horizon 2020 Program (ERC Project REACT, Secure Societies Project PROMETHEUS).