Fast Cryptographic Primitives & Circular-Secure Encryption Based on Hard Learning Problems

Benny Applebaum, David Cash, Chris Peikert, Amit Sahai

Princeton University, Georgia Tech, SRI international, UCLA

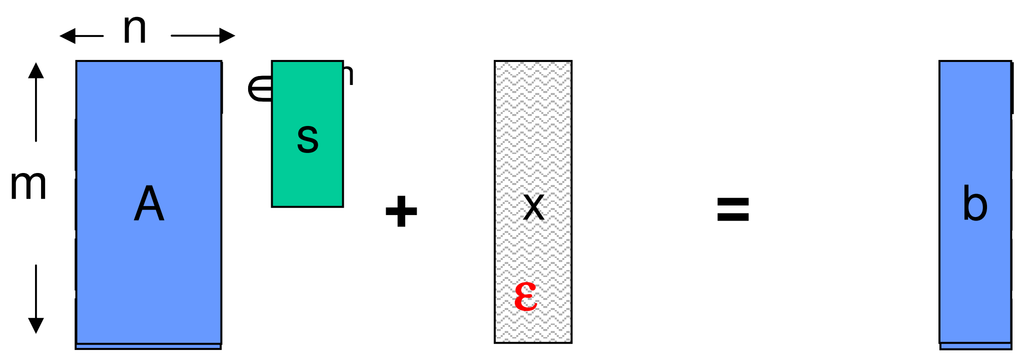
CRYPTO 2009

Learning Noisy Linear Functions

Learning Parity with Noise (LPN)

Problem: find s

$\mathbf{s} \in Z_2^n$

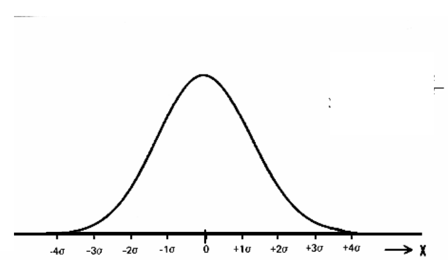


$=\langle a_i, s\rangle + \text{noise}$

iid noise vector of rate ε

e.g., $\varepsilon = 1/4$

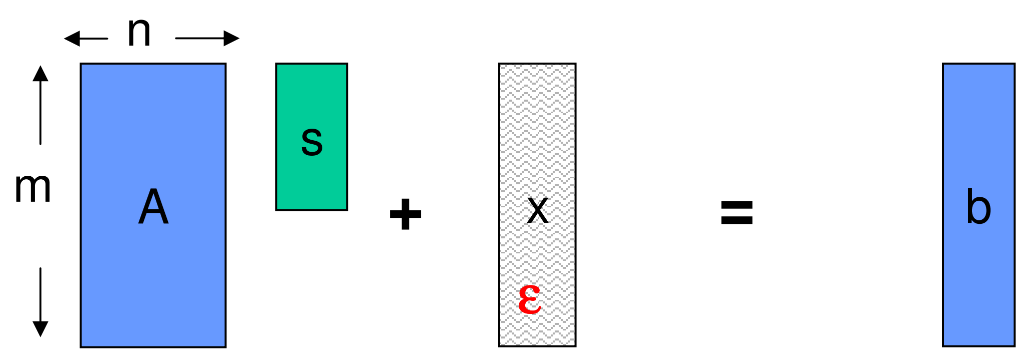
- Extension to larger moduli: Learning-with-Errors (LWE) [Reg05] :
  - Z_q where $q(n)=\text{poly}(n)$ is typically prime
  - Gaussian noise w/mean 0 and std $\approx$ sqrt(q)



-(q-1)/2 0 (q-1)/2

Learning Noisy Linear Functions

Problem: find s



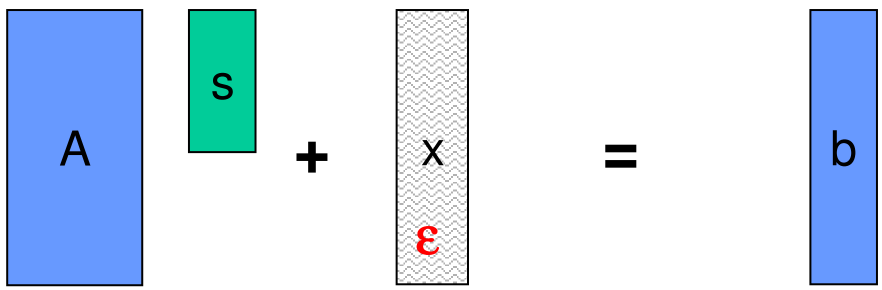
- Assumption: LWE/LPN is computationally hard for all m=poly(n)
- Well studied in Coding Theory/Learning Theory/ Crypto [GKL93,BFKL93, Chab94,Kearns98,BKW00,HB01,JW05,Lyu05,FGKP06,KS06,PW08,GPV08,PVW08…]
- Pros:
  - Reduction from worst-case Lattice problems [Reg05,Peik09]
  - Hardness of search problem
  - So far resists sub-exp & quantum attacks

Why LWE/LPN ?

- Problem has simple algebraic structure: “almost linear” function
  - exploited by [BFKL94, AIK07, D-TK-L09]
- Computable by simple (bit) operations (low hardware complexity)
  - exploited by [HB01,AIK04,JW05]

rare combination

- Message of this talk: Very useful combination



Main Results

This talk:

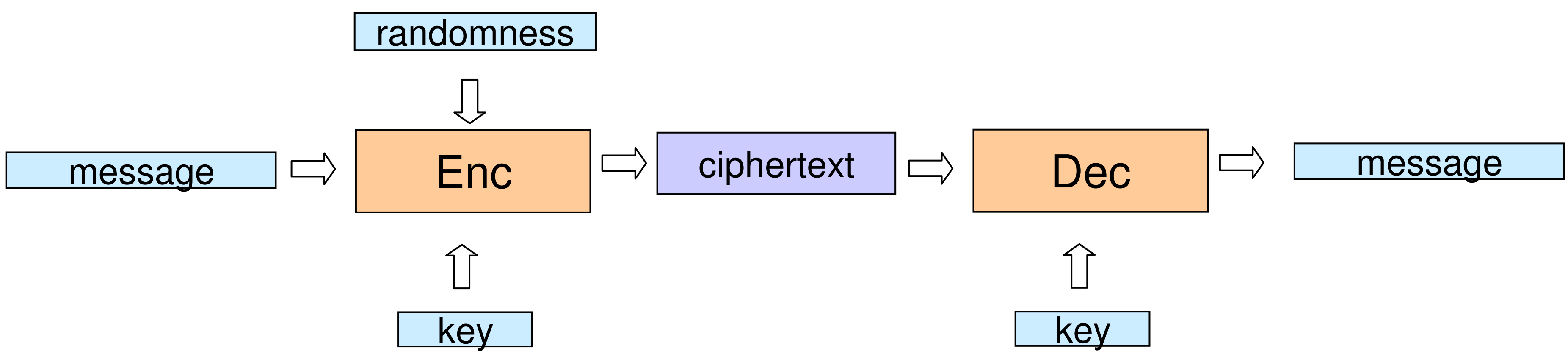
- Fast circular secure encryption schemes
  - Symmetric encryption from LPN
  - Public-key encryption from LWE

- Fast pseudorandom objects from LPN
  - Pseudorandom generator $G:\{0,1\}^n \to \{0,1\}^{2n}$ in quasi-linear time
  - Oblivious weak randomized pseudorandom function

Encryption Scheme

- Security: Even if **Adv** gets information cannot break scheme.
  - CPA [GM82]:given oracle to $E_{key}()$ can't distinguish $E_k(m_1)$ from $E_k(m_2)$

- What if **Adv** sees $E_k(msg)$ where msg depends on the key (KDM attack)?
  - E.g., $E_{key}(key)$ or $E_{key}(f(key))$ or $E_{k1}(k_2)$ and $E_{k2}(k_1)$



KDM / circular security

F-KDM Security [BlackRogawayShrimpton02] : Adv gets $E_k(f(k))$ for $f \in F$

Circular security [CamenischLysyanskaya01] : Adv gets $E_{k1}(k_2)$, $E_{k2}(k_3)\ldots$, $E_{ki}(k_1)$

Can we achieve KDM/circular security?

- many recent works [BRS02, HK07, BPS07, BHHO08, CCS08, BDU08, HU08,HH08]
- natural questi
  - disk encrypti
  - anonymous 01]
  - axiomatic se
  - Gentry's fully homomorphic scheme [Gen09]

[BHHO08]: **Yes, we can !**

- non-trivial to achieve:
  - some ciphers become insecure under KDM attacks (e.g.,AES in LRW mode)
  - random oracle constructions are problematic [HofheintzUnruh08,HaleviKrawczyk07]
  - can't get KDM from trapdoor permutation in a black-box way [HaitnerHolenstein08]

BHHO Scheme vs. Our Scheme

- [BonehHaleviHamburgOstrovsky08] First circular public-key scheme from DDH
  - Get "clique" security + KDM for affine functions
  - But large computational/communication overhead
  - t-bit message: **Time**: t exponentiations (compare to El-Gamal)

    Communication: t group elements

- Our schemes: circular encryption under LPN/LWE
  - Get "clique" security + KDM for affine functions
  - Proofs of security follow the [BHHO08] approach
  - Circular security comes "for free" from standard schemes
  - Efficiency comparable to standard LWE/LPN schemes
  - t-bit message: **Time**: symmetric case: $t \cdot \text{polylog}(t)$;

    public-key: $t^2 \cdot \text{polylog}(t)$

    Communication: $O(t)$ bits.

Symmetric Scheme from LPN

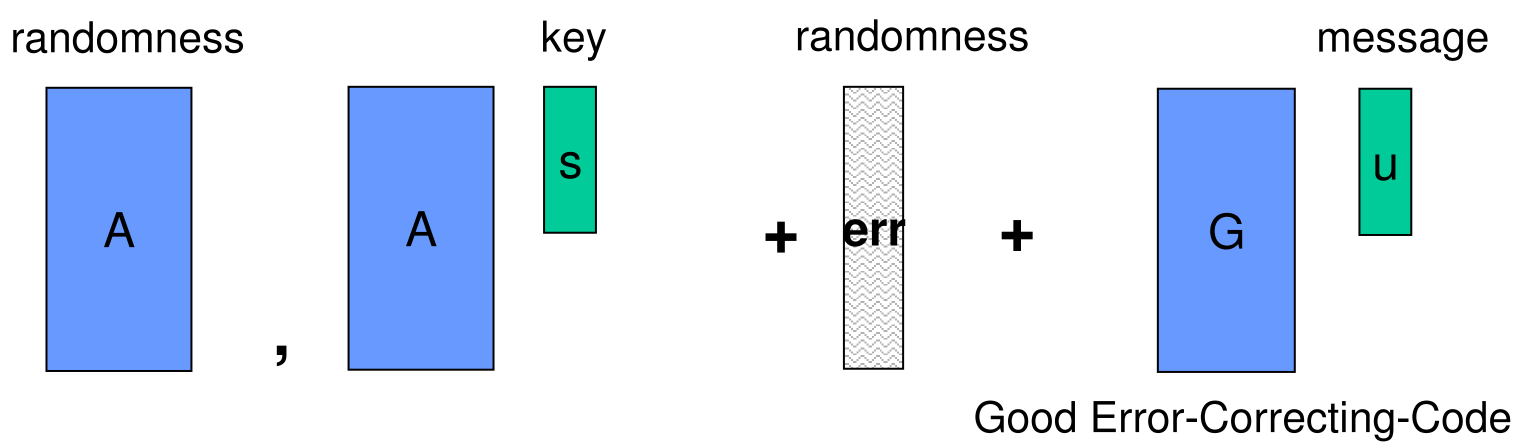
Symmetric Scheme

- Let G be a good linear error-correcting code with decoder for noise ε+0.1

$\mathrm{Enc}_s(\mathrm{mes};\, A,\, \mathrm{err}) = (A,\, As + \mathrm{err} + G \cdot \mathrm{mes})$

$\mathrm{Dec}_s(A, y) = \mathrm{decoder}(y - As)$

- Natural scheme originally from [GilbertRobshawSeurin08]
  - independently discovered by [A08,DodisTauman-KalaiLovet09]
- Also obtain amortized version with quasilinear implementation (See paper)



Clique Security

$Enc_s(mes; A, err) = (A, As+err + G\cdot mes)$

$Dec_s(A,y) = decoder(y-As)$

Thm. Scheme is circular (clique) secure and KDM w/r to affine functions

Proof:

- Useful properties:
  - Plaintext homomorphic: Given $E_s(u)$ and v can compute $E_s(u+v)$

$$(A, As+err + G\cdot(u+v) + G\cdot v \quad)$$

Clique Security

$Enc_s(mes; A, err) = (A, As + err + G \cdot mes)$

$Dec_s(A, y) = decoder(y - As)$

Thm. Scheme is circular (clique) secure and KDM w/r to affine functions

Proof:

- Useful properties:
  - Plaintext homomorphic: Given $E_s(u)$ and v can compute $E_s(v+u)$
  - Key homomorphic: Given $E_s(u)$ and r can compute $E_{s+r}(u)$

$$(A, A\cdot(s+r)+err+Gu \; +A\cdot r \;)$$

Clique Security

$Enc_s(mes; A, err) = (A, As + err + G \cdot mes)$

$Dec_s(A, y) = decoder(y - As)$

Thm. Scheme is circular (clique) secure and KDM w/r to affine functions

Proof:

- Useful properties:
  - Plaintext homomorphic: Given $E_s(u)$ and v can compute $E_s(v+u)$
  - Key homomorphic: Given $E_s(u)$ and r can compute $E_{s+r}(u)$
  - Self referential: Given $E_s(0)$ can compute $E_s(s)$

$$(A - G, \; As + err)$$

$$= (A', \; (A'+G)s + err)$$

$$= (A', \; A's + err + Gs)$$

$$= E_s(s)$$

Clique Security

$Enc_s(mes; A, err) = (A, As+err + G\cdot mes)$

$Dec_s(A,y) = decoder(y-As)$

Thm. Scheme is circular (clique) secure and KDM w/r to affine functions

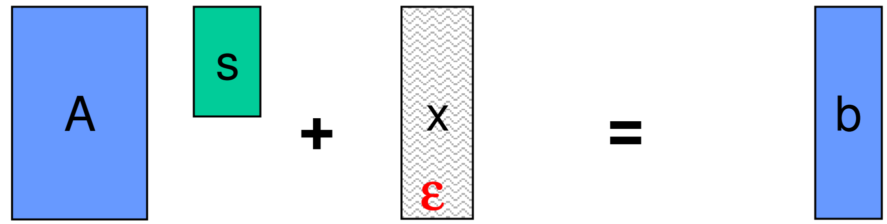
Proof:

- Useful properties:
  - Plaintext homomorphic: Given $E_s(u)$ and v can compute $E_s(v+u)$
  - Key homomorphic: Given $E_s(u)$ and r can compute $E_{s+r}(u)$
  - Self referential: Given $E_s(0)$ can compute $E_s(s)$
- Suppose that Adv break clique security (can ask for $E_{Si}(S_k)$ for all $1 \leq i,k \leq t$)
- Construct B that breaks standard CPA security (w/r to single key S).
- B simulates Adv: choose t offsets $\Delta_1,\ldots, \Delta_t$ and pretend that $S_i=S+\Delta_i$
  - Simulate $E_{si}(S_k)$: get $E_s(0) \rightarrow E_s(S) \rightarrow E_{s+\Delta i}(S) \rightarrow E_{s+\Delta i}(S+\Delta_k)$

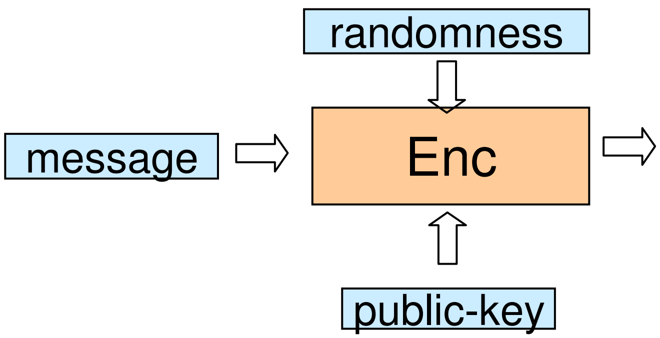
Public-key Scheme from LWE

Regev's Scheme - [GPV-PVW08] variant

- Public-key: $A \in Z_q^{n\times m}$, $b \in Z_q^m$
- Secret-key: $s \in Z_q^n$



- Encrypt $z \in Z_p \subset Z_q$ by $(u \in Z_q^n, c \in Z_q)$



random vector

fixed linear ECC

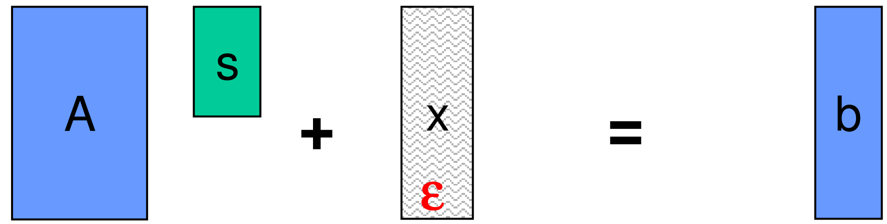
$(u, \langle s,u\rangle + err + g\cdot(message))$

distribution over low-weight elements

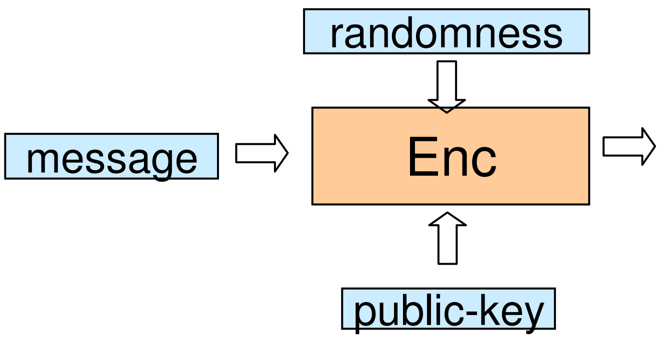
- To Decrypt (u,c): compute $c - \langle s,u\rangle = g\cdot mes + err$ and decode
- CPA Security in [Regev05, GentryPeikertVaikuntanathan08]
- Want: Plaintext homomorphic, Self referential, Key homomorphic

Regev's Scheme - [GPV-PVW08] variant

- Public-key: $A \in Z_q^{n\times m}$, $b \in Z_q^m$
- Secret-key: $s \in Z_q^n$



- Encrypt $z \in Z_p \subset Z_q$ by $(u \in Z_q^n, c \in Z_q)$



random vector

fixed linear ECC

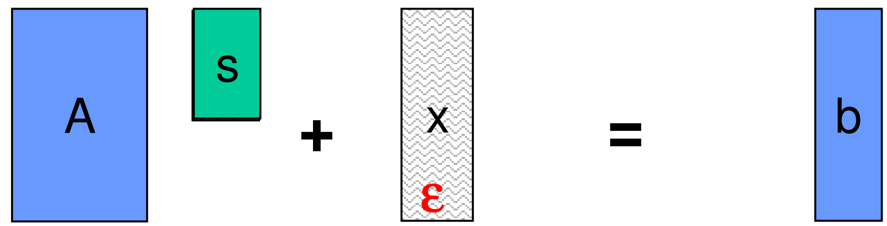
$(u, \langle s,u\rangle + \text{err} + g\cdot(\text{message}))$

distribution over low-weight elements

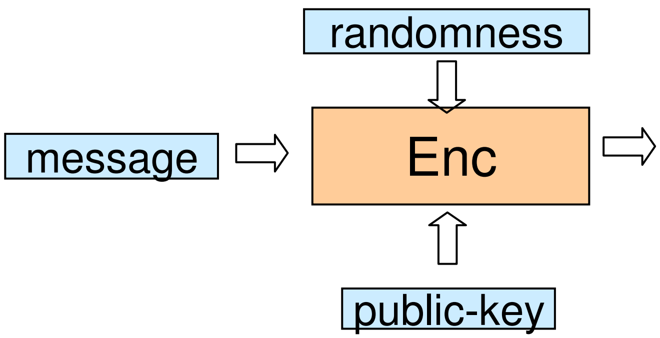
- To Decrypt (u,c): compute $c - \langle s,u\rangle = g\cdot \text{mes} + \text{err}$ and decode
- CPA Security in [Regev05, GentryPeikertVaikuntanathan08]
- Want: Plaintext homomorphic, **Self referential**, Key homomorphic

Self Reference

- Public-key: $A \in Z_q^{n \times m}$, $b \in Z_q^m$
- Secret-key: $s \in Z_q^n$



- Encrypt $z \in Z_p \subset Z_q$ by $(u \in Z_q^n, c \in Z_q)$



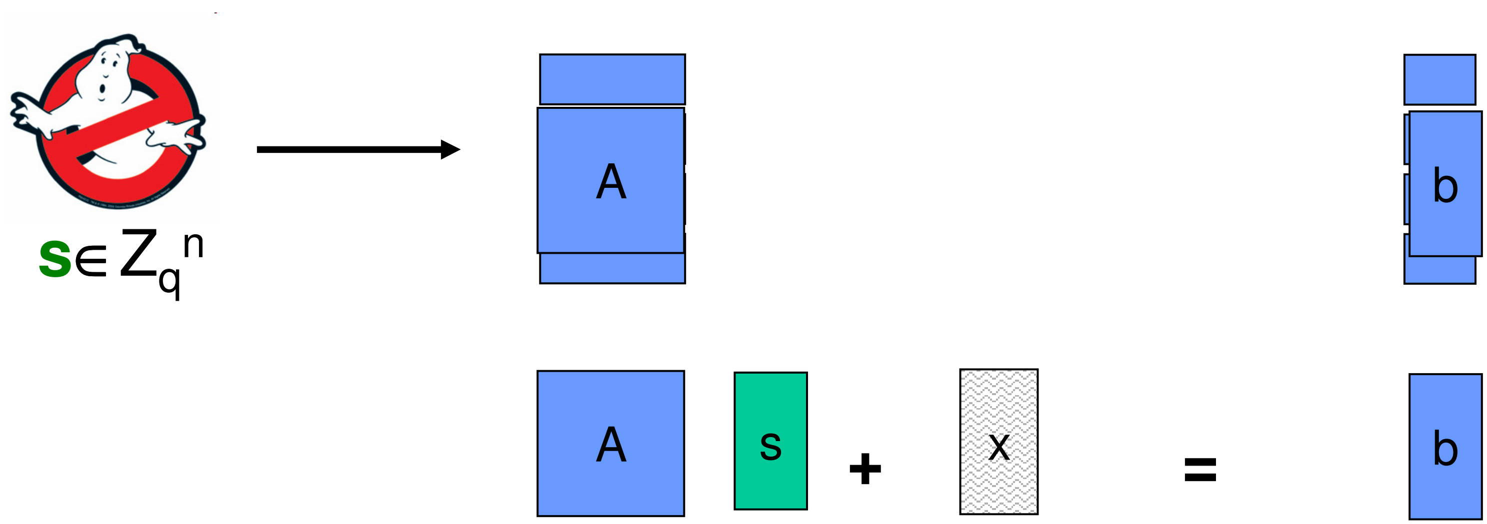
$$(u, \langle s,u \rangle + err + g \cdot (\text{message}))$$

- Can we convert E(0) to E(s_1) ?
- Can use prev ideas (up to some technicalities) but…
- Problem: s_1 may not be in Z_p
- Sol: Choose s with entries in Z_p by sampling from Gaussian around $(0 \pm p/2)$
- Security: we show how to convert standard LWE to LWE with $s \leftarrow$ Noise

Hardness of LWE with s←Noise

Convert standard LWE to LWE with s←Noise

1. Get (A,b) s.t A is invertible



Hardness of LWE with s←Noise

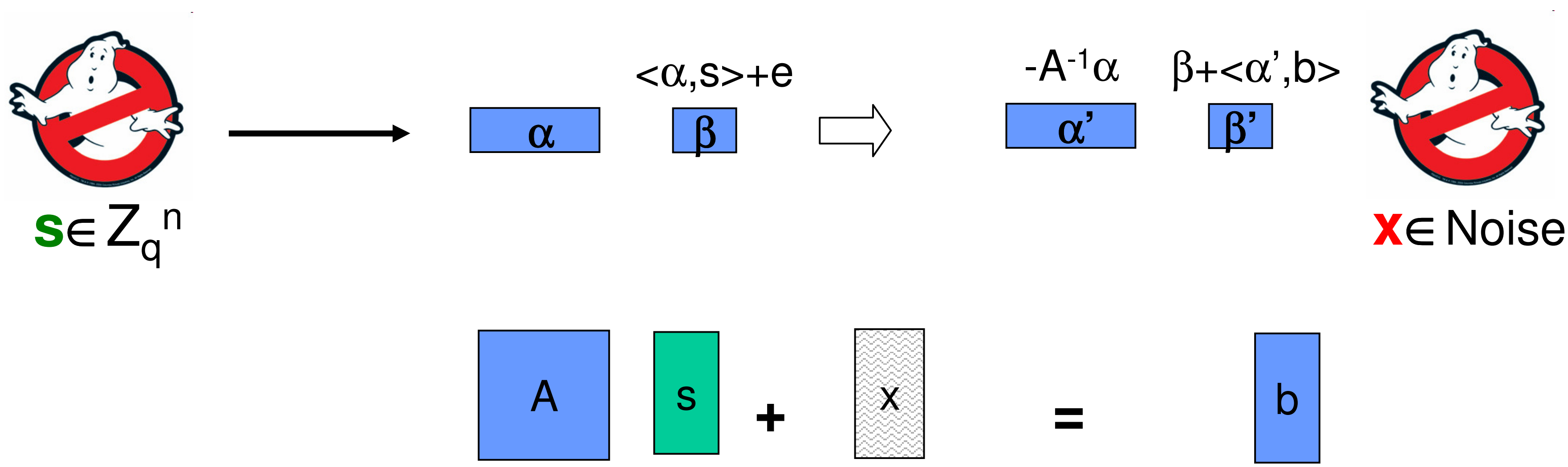
Convert standard LWE to LWE with s←Noise

- If $(\alpha,\beta) \leftarrow LWE_s$ then $(\alpha',\beta') \leftarrow LWE_x$

  Proof: $\beta' = \beta + \langle \alpha', b \rangle$

  $= \langle \alpha, s \rangle + e + \langle \alpha', As \rangle + \langle \alpha', x \rangle$

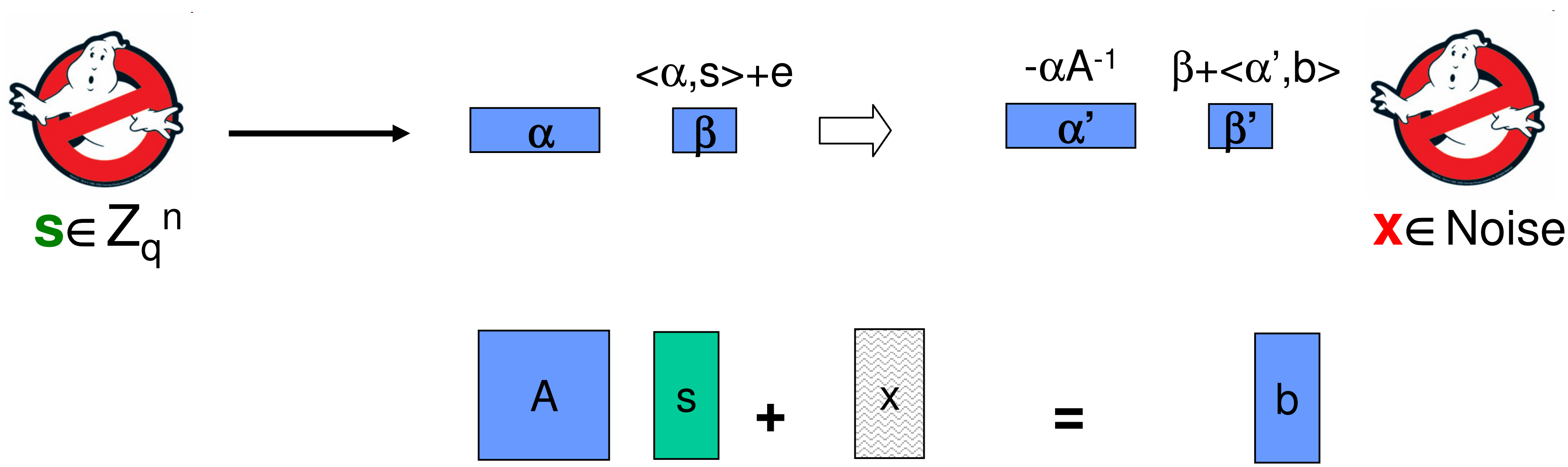
  $= \langle \cancel{\alpha, s} \rangle + e + \langle \cancel{-A^{-1}\alpha, A}s \rangle + \langle \alpha', x \rangle$



Hardness of LWE with $s \leftarrow$Noise

Convert standard LWE to LWE with $s \leftarrow$Noise

- If $(\alpha,\beta) \leftarrow LWE_s$ then $(\alpha',\beta') \leftarrow LWE_x$
- If (α,β) are uniform then (α',β') also uniform
- Hence distinguisher for LWE_x yields a distinguisher for LWE_s



Hardness of LWE with s←Noise

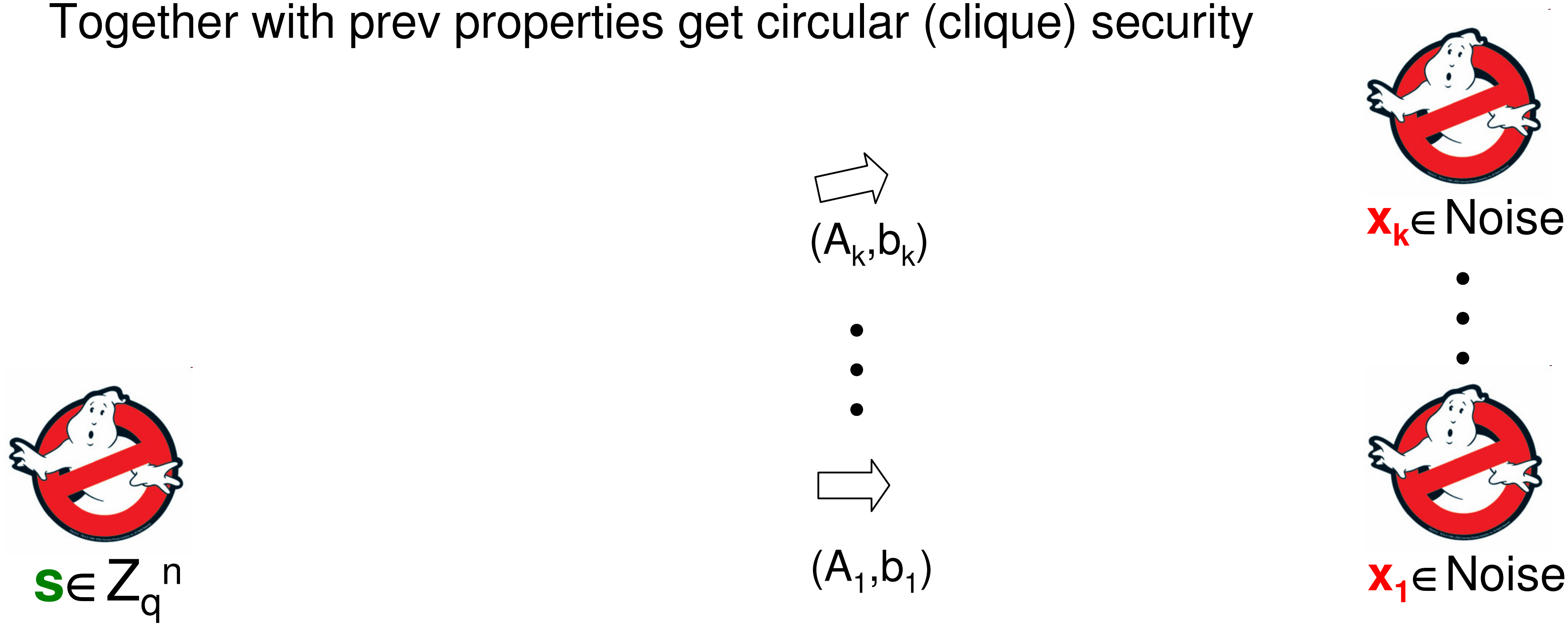
- Reduction generates invertible linear mapping $f_{A,b}$:s $\rightarrow$ x

s$\in Z_q^n$ $\Rightarrow$ (A,b) x$\in$ Noise

A s + x = b

Hardness of LWE with s←Noise

- Reduction generates invertible linear mapping $f_{A,b}$:s → x
- Key Hom: get pk's whose sk's $x_1,..,x_k$ satisfy known linear-relation
- Together with prev properties get circular (clique) security



- Improve efficiency via amortized version of [PVW08]

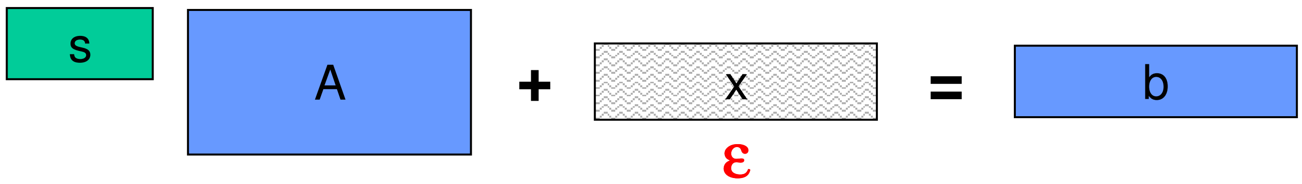
Open Questions

- LWE vs. LPN ?
  - LWE follows from worst-case lattice assumptions [Regev05, Peikert09]
  - LWE many important crypto applications [GPV08,PVW08,PW08,CPS09]
  - LWE can be broken in "NP$\cap$ co-NP" unknown for LPN
  - LPN central in learning ("complete" for learning via Fourier) [FeldmanGopalanKhotPonnuswami06]

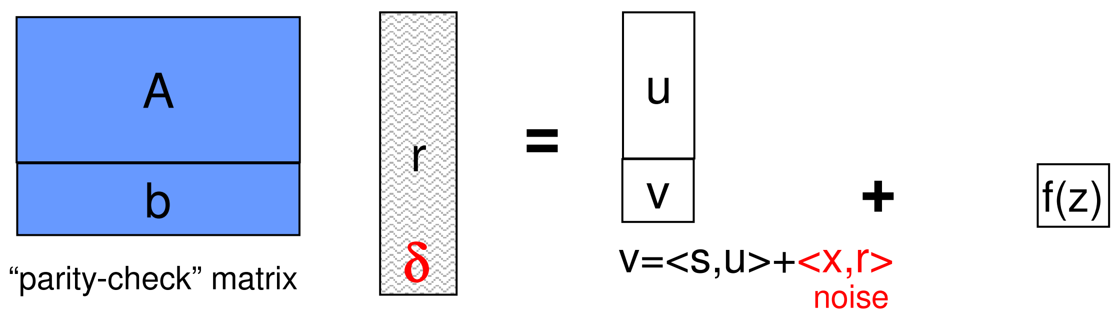
- Circular Security vs. Leakage Resistance ?
  - Current constructions coincident
  - LPN/Regev/BHHO constructions resist key-leakage [AkaviaGoldwasserVaikuntanathan09, DodisKalaiLovett09, NaorSegev09]
  - common natural ancestor?

Regev's Scheme - [GPV-PVW08] variant

- Public-key: $(A,b) \in Z_q^{n\times m} \times Z_q^m$ Secret-key: $s \in Z_q^n$

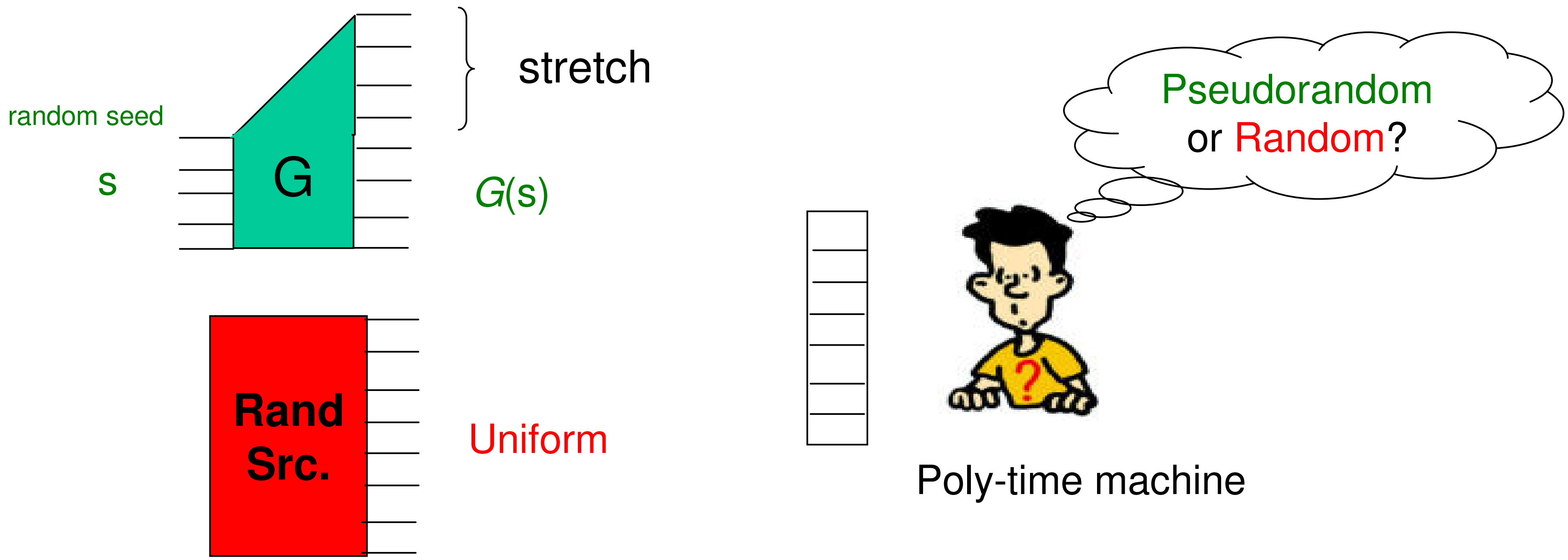


- Encrypt $z \in Z_p \subset Z_q$ by $(u,v+f(z))$ where $f: Z_p \to Z_q$ is linear ECC, i.e., $f(z)=az$



- To Decrypt (u,c): compute $c-\langle s,u\rangle=f(z)+\langle x,r\rangle$ and decode
- Security [R05,GPV]: If b was truly random then (u,v) is random and get OTP
- Want: Plaintext homomorphic, Self referential, Key homomorphic
- Plaintext hom: let message space be subgroup of Z_q by taking $q=p^2$

Pseudorandom Generator (PRG)



- Can be constructed from any one-way function [HILL90]
- Stretch of 1 bit $\Rightarrow$ Stretch of polynomially many bits [BM-Y, GM84]

Circuit Complexity of PRGs

Pseudorandom generator $G:\{0,1\}^n \rightarrow \{0,1\}^{2n}$

- At least $\Omega(n)$ circuit size
- Can we get low overhead of $O(n)$ or $n \cdot \text{polylog}(n)$?
  - natural question
  - [IKOS08] PRG with low overhead $\Rightarrow$ low-overhead cryptography
    e.g., PK-encryption in time O(|message|), for sufficiently large message.

Construction	Assumption	Time (circuit size)
[BM84, GM84]	1-bit PRG G'	$n \cdot \text{Time}(G') > n^2$
[Gen00,DRV02, DN02]	Number Theoretic	More than n^2
[BFKL94, FS96]	LPN	n^2
[AIK06]	sparse-LPN (non-standard)	n
This work	LPN (standard)	$n \cdot \text{polylog}(n)$

Circuit Complexity of PRGs

Pseudorandom generator $G:\{0,1\}^n \rightarrow \{0,1\}^{2n}$

- Can we get low overhead of $O(n)$ or $n \cdot \mathrm{polylog}(n)$?
  - natural question
  - [IKOS08] PRG with low overhead $\Rightarrow$ low-overhead cryptography
    e.g., PK-encryption in time O(|message|), for sufficiently large message.

Construction	Assumption	Time (circuit size)
[BlumMicali84, GoldreichMicali84]	1-bit PRG G'	$n \cdot \mathrm{Time}(G') > n^2$
[Genarro00, DedicReyzinVadhan02, DamgardNielsen02]	Number Theoretic	More than n^2
[BlumFurstKearnsLipton94, FischerStern96]	LPN	n^2
[A-IshaiKushilevitz06]	sparse-LPN (non-standard)	n
This work	LPN (standard)	$n \cdot \mathrm{polylog}(n)$

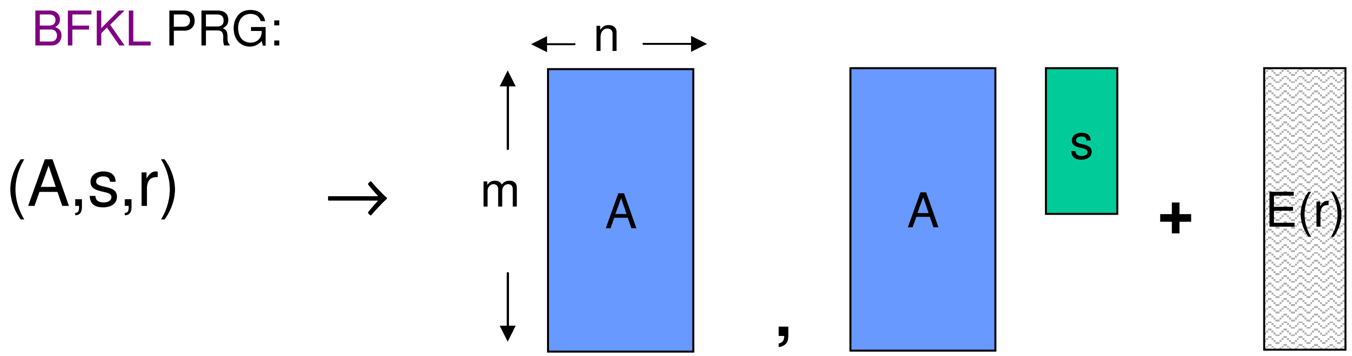
The [BFKL] generator

BFKL generator: $G(A, s, r) = (A, As + Err(r))$

- input: $nm+n+mH_2(\varepsilon)$ output: $nm+m$ stretch: $m(1-n/m - H_2(\varepsilon))$
- Efficiency: only bit operations !
- Bottleneck 1: at least $\Omega(mn)$ due to matrix-vector multiplication
- Bottleneck 2: Sampling Err(r) (with low randomness complexity) takes time

[FischerStern96] : quadratic time on a RAM machine

BFKL PRG:



Solving 1: Amortization

BFKL generator: G(A, s, r)= (A,As+ Err(r))

- Bottleneck 1: at least $\Omega(mn)$ due to matrix-vector multiplication
- Sol: Amortization
- Use many different s's with the same A
- Preserves pseudorandomness since A is public

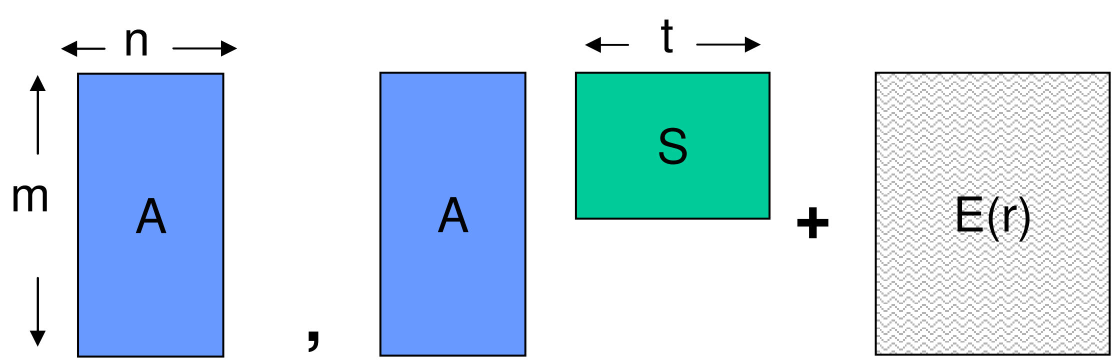
    -Proof via Hybrid argument

- If matrices are very rectangular can multiply in quasi-linear time [Cop82]

    - E.g., t=n and m=n^6

PRG:

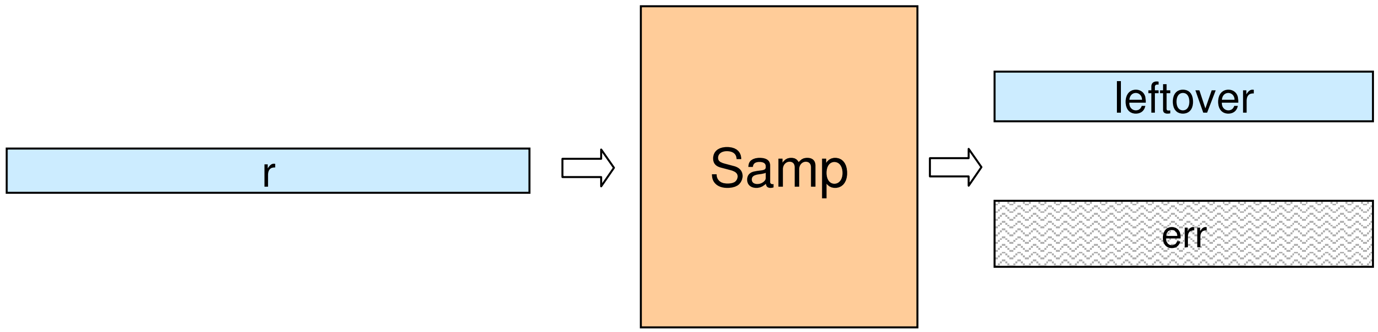
(A,S,r) $\rightarrow$



Solving 2: Sampling with leftovers

Bottleneck 2: Sampling noise w/low randomness takes $O(n^2)$

- Sol: [AIK06] Samp(r)= (err, leftover)



- PRG G(A,S,r)= (A, AS+err, leftover)
- How to sample w/leftovers?
  - If $\varepsilon=1/4$ partition r to pairs and let $err_i = r_{2i-1} \cdot r_{2i}$
  - r has a lot of entropy given err, so can extract the leftover
  - Can get linear time with leftover of linear length
- G has linear stretch and computable in quasi-linear time

Open Questions

- LWE vs. LPN ?
  - LWE follows from worst-case lattice assumptions [Regev05, Peikert09]
  - LWE many important crypto applications [GPV08,PVW08,PW08,CPS09]
  - LWE can be broken in “NP$\cap$ co-NP” unknown for LPN
  - LPN central in learning (“complete” for learning via Fourier) [FGKP06]
- Circular Security vs. Leakage Resistance ?
  - Current constructions coincident
  - LPN/Regev/BHHO constructions resist key-leakage [AGV09,DKL09,NS09]
  - common natural ancestor?

Conclusion and Open Questions

- DRLC is useful for private-key primitives that need
  - fast hardware implementation
  - special homomorphic properties

- Find more crypto application for DRLC
  - collision resistance hash-functions
  - public-key crypto [Alekh03] uses $m=O(n)$, $\varepsilon=\mathrm{sqrt}(n)$