Efficient Pseudorandom Correlation Generators:
MPC with Silent Preprocessing

Peter Scholl

21 January 2020, IISc Bangalore

Joint work with:
Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Peter Rindal
Outline

• **Pseudorandom correlation generators (PCGs)**
  ➢ Motivation: MPC in the preprocessing model

• Why **LPN** is a perfect match for HSS/PCGs

• PCG for **OT** from LPN:
  ➢ Two-round “silent” OT extension
  ➢ Practical

• PCG for **OLE** from LPN
  ➢ Concretely efficient under variant of ring-LPN
Secure Computation with Preprocessing

[Beaver ‘91]

Correlated randomness

Interactive protocol

Online phase

Dominates overall cost

\[ f(x, y) \]

\- Information-theoretic
\- Constant comp. and comm. overhead

\( x \)

\( y \)

Peter Scholl 3
Secure Computation with Silent Preprocessing

[BCGI 18, BCGIKS 19]

Correlated, short seeds

Silent expansion

Correlated pseudorandomness

“Small” setup protocol

• Less communication
• Lower storage costs

Online phase

\[ f(x, y) \]

\( x \)

\( y \)
Pseudorandom Correlation Generators

[BCGI 18, BCGIKS 19]

- Target correlation: \((R_0, R_1)\)
  - E.g. random OT \(((b, m_b), (m_0, m_1))\)
- Algorithms Gen, Expand:

\[
\begin{align*}
(k_0, k_1) & \leftarrow \text{Gen}(1^\lambda) \\
\tilde{R}_0 & \leftarrow \text{Expand}(k_0) \\
\tilde{R}_1 & \leftarrow \text{Expand}(k_1)
\end{align*}
\]

Security: \( (k_0, \tilde{R}_1) \approx (k_0, [R_1 \mid R_0 = \text{Expand}(k_0)]) \)
Landscape of PCGs

“Gentria”
- LWE+
  General additive correlations [BCGIKS 19]

“Cryptomania”
- DDH
- LWE + low-degree PRG
  Low-degree correlations (1/poly error) [BCGIO 17]
- LWE + low-degree PRG
  Low-degree correlations [BCGIKS 19]

“Lapland”
- LPN
  vector-OLE [BCGI 18]
  OT, constant degree correlations [BCGIKS 19]

“Minicrypt”
- OWF
  Linear correlations [GI 99, CDI 05]
  Truth tables [BCGIKS 19]
Background: LPN and LWE (spot the difference!)

Given $A \in \mathbb{Z}_p^{m \times n}$:

\[
\begin{array}{c}
A \\
\hline
s \\
\hline
+ \\
\hline
\text{mod } p \\
\hline
\approx \\
\hline
u
\end{array}
\]

**LWE**
- $p > 2$
- $s \leftarrow \mathbb{Z}_p^n$
- $\|e\|_\infty$ is small

**LPN**
- $p = 2$
- $\not\Rightarrow p \geq 2$ (arithmetic generalization/RLC)
- $s \leftarrow \mathbb{Z}_p^n$
- $\|e\|_\infty$ is small
LWE and LPN: what are they good for?

Minicrypt
- PRG
- SKE
- Sig.

Cryptomania
- Additive HE
- FHE
- FE
- PKE
- OT

Today
- Low noise
- "MPC-friendly" PRG

LPN
- Low noise
- *for low deg. functions

LWE
- *for low deg. functions
Simple PRGs from LPN

"Primal" construction

\[(s, e) \rightarrow m \quad A \quad n \quad s \quad + \quad e \quad (m - n) \quad H \quad e \quad e \]

\[t := HW(e)\]

\[n + m \cdot \log t\] bits

\[m \cdot \log t\] bits

Security: both equiv. to LPN (if \(H\) is parity-check matrix of code \(A\))

Limited to quadratic stretch

"Dual" construction

Arbitrary poly stretch

➢ best attack: \(\exp(t)\)
Blueprint: How to exploit sparse noise for PCGs

**Step 1**: Compress secret-shares of sparse vector with FSS

\[ e = e_0 + e_1 \]

**Step 2**: Use \( e \) as seed for PRG \( e \rightarrow H \cdot e \)
I: PCG for oblivious transfer from LPN
Oblivious Transfer

• **Problem:** OT is expensive ("public-key")

• **OT extension:** many OTs from a few base OTs + symmetric crypto \[\text{[IKNP 03]}\]

• **Problem:** communication $O(n \lambda)$ for $n$ OTs

• **Silent OT extension:** communication sublinear in $n$
Towards silent OT extension

**Goal:** a PCG for correlated OT

i.e. compression of:

\[ \vec{v} + \vec{w} = y \cdot \vec{b} \]
Silent OT Extension: Overview

Setup

Receiver

seed

PPRF

sparse $\vec{e}$

$\in \{0,1\}^m$

LPN

uniform

Shares of $y \cdot \vec{e}$:

$\in \mathbb{F}_2^\lambda$

$\iff$ correlated OTs

$\iff$ Random OTs

Sender

seed

local comp.

Peter Scholl
Main tool: puncturable PRF

- PRF $F : \{0,1\}^\lambda \times \{1, \ldots, N\} \rightarrow \{0,1\}^\lambda$

- $k \leftarrow \text{Gen}(1^\lambda)$
  - Master key: allows evaluating $F(k, x)$ for all $x$

- $k^* \leftarrow \text{Punc}(k, \alpha)$
  - Punctured key: can evaluate at all points except for $x = \alpha$

- Security: $F(k, \alpha)$ is pseudorandom, given $k^*$

Simple tree-based construction from a PRG: $|k| = \lambda$, $|k^*| = \lambda \cdot \log N$

[1][BW13], [BGI 13], [KPTZ 13]
Key observation: puncturable PRF compresses sparse vectors

**Setup**

\[
\alpha \leftarrow \{1, \ldots, N\} \\
k \leftarrow \text{Gen}(1^\lambda) \\
k^* \leftarrow \text{Punc}(k, \alpha)
\]

**Receiver**

- \(\alpha, k^*\)
- \(z = F(k, \alpha) + y\)
- \(\emptyset\) at pos. \(\alpha\)

**Sender**

- \(k\)
- \(y \in \mathbb{F}_{2^\lambda}\)

**Eval at all** \(x \neq \alpha\)

\[
\mathcal{B}(k, \alpha) = \begin{bmatrix}
0 & \ldots & 0 & 1 & 0 & \ldots & 0\end{bmatrix}
\]

- Shares **compressed** from \(\lambda \cdot N\) to \(\approx \lambda \cdot \log N\) bits
- Can tweak to multiply by arbitrary \(y \in \mathbb{F}_{2^\lambda}\)
From weight-1 vectors to weight-\(t\) vectors

**Approach 1: addition**

\[ y \cdot e_1 + \cdots + y \cdot e_t \]

Weight e.g. \(t = 4\)

**Expansion cost:** \(O(t \cdot N)\) (naïve)

\(O(N)\) (batch codes [BCGI18, SGRR 19])

**Approach 2: concatenation**

\[ O \left( t \cdot \frac{N}{t} \right) = O(N) \]

**Note:** regular error pattern
From sparse products to correlated OT

• Recall, have shares:
• Want: **uniform vector**

\[ \text{Public linear } H \]

\[ \text{Pseudorandom under LPN!} \]
Setup protocol: inside the puncturable PRF

Suppose Receiver has \( \alpha \) for first 2 levels:

Use OT to transfer next \( \alpha \):

Left/right

Recover

(\( \text{sum of L, sum of R} \))

Based on [Doerner-shelat ‘17]

\( \alpha \)

OTs for all levels can be done in parallel!

(Unlike [Ds 17] for DPF)
Recap: silent OT extension

• Setup protocol: 2 rounds from any 2-round OT
  ➢ Cost: $O(\lambda \log N)$ base Ots

• Silent expansion ($N$ OTs):
  ➢ $O(N \log N)$ PRF evaluations
  ➢ 1 multiplication $H \cdot x$

• Implies two-round OT extension on chosen inputs
  ➢ Can convert from random $\rightarrow$ chosen in parallel with setup
  ➢ First concretely efficient two-round OT extension
    (bypass [GMMM 18] impossibility via LPN)
Extras: active security, implementation

• Active security:
  ➢ Lightweight PPRF consistency checks for malicious sender
    o Allows selective failure attacks – sender can guess 1 bit of LPN error
    o Assume problem is hard with 1-bit leakage
  ➢ 10-20% overhead on top of semi-honest

• Implementation:
  ➢ Main challenge: fast mult. by $H$
  ➢ Quasi-cyclic $H$: polynomial mult. mod $X^n - 1$
  ➢ Security based on quasi-cyclic syndrome decoding / ring-LPN
Runtimes (ms) for $n=10$ million random OTs

**LAN (10 Gbps)**
- Runtimes: 268 ms

**WAN (100 MBps)**
- Runtimes: 13728 ms
- 5x improvement over LAN

**WAN (10 MBps)**
- Runtimes: 128854 ms
- 47x improvement over LAN

**IKNP vs silent OT**

Total comm: **160 MB vs 127 kB**

Peter Scholl
II: PCG for OLE correlations from LPN and ring-LPN
Degree-2 correlation: Oblivious Linear Evaluation (OLE)

\[ y = ax + b \]

Related: multiplication triples
  - Obtained from 2 random OLEs (two parties)
Main tool: FSS for point functions

- Point function $f_{\alpha,\beta} : \{1, \ldots, N\} \rightarrow \{0,1\}^\lambda$

$$f_{\alpha,\beta}(x) = \begin{cases} 
\beta & \text{if } x = \alpha \\
0 & \text{o. w.}
\end{cases}$$
PCG for tensor product from LPN and FSS
[BCGIKS ’19]

• Pick $e, f$ with $HW \ t$
• Tensor product $e \otimes f$ is sparse
• Distribute shares of $e, f$ and $e \otimes f$
  ➢ With FSS for $O(t^2)$ points
PCG for tensor product from LPN and FSS

$[\text{BCGIKS '19}]$

$t$-sparse $e, f$

$t^2$-point FSS for $e \otimes f$

Eval at $\{1, \ldots, n^2\}$

$R_0 + R_1 = e \otimes f$

$H \cdot (R_0 + R_1) \cdot H^T = (He) \otimes (Hf)$
Applications of PCG for tensor product

• Deg-2 correlations:
  ➢\(n\) OLEs or Beaver triples with \(o(n)\) communication
  ➢Computation: \(\Omega(n^2)\)
  ➢Extends to deg-\(d\) (cost: \(\Omega(n^d)\))

• PCG for deg-\(d\) ⇒ \textbf{homomorphic secret-sharing} for deg-\(d\) functions
  ➢Let (Gen, Expand) be PCG for \(R = [r, r \otimes r, \ldots, r \otimes^d r]\)
  ➢\textbf{Share}(x): apply Gen and make \(x' = x + r\) public
  ➢\textbf{Eval}_p: write \(p(x)\) as \(p'(r)\), where \(p'\) is determined by \(x'\), and linear in \(R\)
Efficient PCG for OLE from ring-LPN
[ongoing work]

• Idea:
  ➢ Replace tensor product with polynomial multiplication
  ➢ Similar to [BV11] for FHE

• Take sparse polys $e, e', f, f'$
• Distribute shares of $(e, e') \boxtimes (f, f')$
• Output

$$\left((he + e') \cdot (hf + f')\right) \mod (X^n + 1)$$

for public, random $h \in Z_p[X]$
Efficient PCG for OLE from ring-LPN
[ongoing work]

• **Cost**: for 1 OLE in $\mathbb{Z}_p[X]/(X^n + 1)$
  \[O(t^2 + n \log n)\] computation

Gives $n$ OLEs in $\mathbb{Z}_p$ if $X^n + 1$ splits into linear factors mod $p$

• **Security**:
  \[\text{Arithmetic ring-LPN}\]

  \[(h, h \cdot s + e) \mod (p, F(X))\]

  \[\text{Does not appear significantly weaker}\]
Conclusion

• PCG for OT from LPN
  ➢ Random OT (and correlated OT): practical, almost zero communication
  ➢ (previously: \(n\) bits per OT)
  ➢ Two-round OT extension

• PCG for OLE
  ➢ From LPN (expensive)
  ➢ Efficient from fully splitting ring-LPN

• Open problems:
  ➢ Optimize OT: better codes
  ➢ Security of arithmetic ring-LPN
  ➢ Efficient PCGs for more correlations:
    o Truth tables (active security), random bits (\(\mathbb{Z}_p\)), garbled circuits...
Thank you!

Efficient Pseudorandom Correlation Generators: Silent OT Extension and More
Boyle, Couteau, Gilboa, Ishai, Kohl, Scholl
https://ia.cr/2019/129

Two-Round OT Extension and Silent Non-Interactive Secure Computation
BCGIKS + Rindal
https://ia.cr/2019/1159
Code: https://github.com/osu-crypto/libOTe