EUROCRYPT 2021 PRESENTS

THE RISE OF PAILLIER

Homomorphic Secret Sharing and Public-Key Silent OT

Starring:
CLAUDIO ORLANDI
PETER SCHOLL
SOPHIA YAKOUBOV

Aarhus University
Outline

- Homomorphic Secret Sharing
  - Background
  - Share conversion and distributed multiplication for Paillier

- Pseudorandom correlation functions
  - Producing correlated randomness
  - Public-key setup for vector-OLE and oblivious transfer
Homomorphic Secret Sharing

[Boyle Gilboa Ishai 16]

- Security: $x_0$ hides $x$, $x_1$ hides $x$
- Correctness: $\text{Eval}_P(x_0) + \text{Eval}_P(x_1) = P(x)$
HSS Landscape

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Program type</th>
<th>Error pr.</th>
<th>Msg space</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Ben86]</td>
<td>linear</td>
<td>negl</td>
<td>exp</td>
</tr>
<tr>
<td>[DHRW16, BGI15, BGILT16]</td>
<td>LWE+</td>
<td>any</td>
<td>negl</td>
</tr>
<tr>
<td>[GI14, BGI15]</td>
<td>OWF</td>
<td>simple (e.g. point)</td>
<td>negl</td>
</tr>
<tr>
<td>[BCGIKS19]</td>
<td>LPN</td>
<td>low-deg polynomials</td>
<td>negl</td>
</tr>
<tr>
<td>[BKS19]</td>
<td>LWE</td>
<td>Branching programs</td>
<td>negl</td>
</tr>
<tr>
<td>[BGI16, FGJS17]</td>
<td>DDH, Paillier</td>
<td>Branching programs</td>
<td>1/poly</td>
</tr>
<tr>
<td>This work</td>
<td>Paillier</td>
<td>Branching programs</td>
<td>negl</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
    x &\rightarrow \text{Share} \\
    x_0 \rightarrow \text{Eval}_P &\rightarrow P(x)_0 \\
    x_1 \rightarrow \text{Eval}_P &\rightarrow P(x)_1 \\
    + &\rightarrow P(x)
\end{align*}
\]
HSS for Branching Programs: High-Level Template  [Boyle Gilboa Ishai 16]

**Branching program model:** circuit, where every multiplication involves an input wire

Value types:
- **Input:** ciphertext
- **Memory:** linear shares $y_0, y_1$

Operations:
- $\text{Enc}(x)$
- $+$ : via linearity
- $\times$ : ?
- $\rightarrow$ output : reconstruction
Blueprint for Multiplication

Enc(x) \rightarrow y_0 \rightarrow \text{Mult} \rightarrow (xy)_0

Enc(x) \rightarrow y_1 \rightarrow \text{Mult} \rightarrow (xy)_1
Blueprint for Multiplication

Step 1

\[ \frac{g_0}{g_1} = g^{xy} \]

\[ (xy)_0 + (xy)_1 = xy \]

\( d: \) Secret key for Enc
Distributed Discrete Log

[Boyle Gilboa Ishai 16]

- $g_0 / g_1 = g^{xy}$
- $(xy)_1 - (xy)_0 = xy$
- Problem: what if $(xy)_0, (xy)_1$ are large?
  - Have many h’s
  - Poly-size message space
- Problem: error if parties hit different h
  - Gives 1/poly error!
- Various optimizations: still 1/poly error, poly message space
  [BGI16, BGI17, BCGIO17, DKK18]
- Variant in Paillier groups: same limitations [FGJS 17]
DDLog: Paillier

- $g_0/g_1 = (1 + N)^{xy} \mod N^2$
- Use just one $h$:
  - $h/g_i = (1 + N)^{(xy)_i} \rightarrow (xy)_i h$
  - Use $h := g_1 \mod N = g_0 \mod N$ (in $\mathbb{Z}_{N^2}$)
- $h$ is in the same coset!
  - Intuition: $g_1 \mod N$ uniquely represents the coset defined by $g_1$
  - $(h \mod N) = h \Rightarrow h$ is in the coset
- Large message space, negl error!

Paillier 101:
- Paillier group: $\mathbb{Z}_{N^2}^*$, $N = pq$
- $1 + N$ generates an easy DLog subgroup
HSS from Paillier: summary

- Basic construction:
  - Negligible correctness error, exponential message space
  - To repeatedly multiply, need circular security of Paillier
  - Concurrent work: [Roy-Singh 21] using Damgård-Jurik

- Public-key variant:
  - Share input without knowing private key
  - Use ElGamal over $\mathbb{Z}_N^*$

- Circular secure variant:
  - Based on [Brakerski-Goldwasser 10] encryption

Share size: $O(1)$ group elements

$O(\lambda)$ group elements
II: Pseudorandom Correlation Functions
Pseudorandom Correlation Function

[BCGIKS20]

<table>
<thead>
<tr>
<th>Correlation</th>
<th>$R_0$</th>
<th>$R_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Oblivious Transfer (OT)</td>
<td>$b, s_b$</td>
<td>$s_0, s_1$</td>
</tr>
<tr>
<td>Oblivious Linear Evaluation (OLE)</td>
<td>$x, (xy)_0$</td>
<td>$y, (xy)_1$</td>
</tr>
<tr>
<td>Vector OLE (VOLE)</td>
<td>$x_i, (x_iy)_0$</td>
<td>$(x_iy)_1$</td>
</tr>
</tbody>
</table>
Pseudorandom Correlation Function

[BCGIKS20]

Previous constructions:
- additive correlations from LWE (expensive)
- OT, deg-2 correlations from Variable-density LPN (new assumption)

This work: VOLE, OT from Paillier, QR

- Note on efficiency
  - All require exponentiations!
    - slower than LPN-based alternatives
  - Advantages: smaller keys, simpler constructions, standard assumptions
PCF for VOLE

Important: obliviously sampleable!

Vector OLE (VOLE)

| $x_i, (x_iy)_0$ | $(x_iy)_1$ | $y$ |
Setup?

Public-key setup:

- One message from Alice/Bob
- With non-interactive (vector)-OLE based on DDLog
Public Key PCF: Protocol Flow

CRS Setup
Give out $N' = p'q'$

Nonce $\rightarrow$ HSS Mult $\rightarrow$ Nonce

$pk_A$ $\rightarrow$ HSS Mult $\leftarrow$ $pk_B$

Nonce $(dy)_0$ $\rightarrow$ HSS Mult $\leftarrow$ Nonce $(dy)_1$
Conclusion

Share conversion for Paillier:

Locally convert multiplicative shares of \((1 + N)^x\) into additive shares of \(x\)

Homomorphic secret sharing for branching programs
  o Negligible error, large plaintexts

Pseudorandom correlation functions
  o Produce arbitrary quantity of VOLE or OT
  o Based on oblivious ciphertext sampling
  o Public-key setup
Open problems

- Improve OT efficiency
  - $O(\lambda)$ exponentiations

- Remove CRS $N'$ from public-key setup

- More correlations: OLE from Paillier?

- Public-key PCFs from other assumptions (LPN?)

- Beyond two parties?