TinyKeys: A new approach to efficient multi-party computation

Carmit Hazay, Emmanuela Orsini, Peter Scholl and Eduardo Soria-Vazquez
Motivation: Large Scale, Dishonest Majority

Large number of users want to conduct surveys, auctions, statistical analysis, measure network activity, etc.

- MPC between all users
- Outsource to a committee

Dishonest Majority:
More parties ⇒ More trustworthy
MPC setting in this talk

Main focus:
• Concrete efficiency for large numbers of parties (e.g. $n$ in $10s$, $100s$)

Adversary:
• Static, passive
• Dishonest majority ($t > n/2$)

Model of Computation:
• Boolean circuits
• Preprocessing phase
Corruption thresholds vs communication complexity of *practical* MPC

$n$ parties
$k$-bit security

Corruptions: $0 \quad \frac{n}{2} \quad n - 1$

Efficiency: $O(n \log n) \quad 0(n^2 k)$
Can we design concretely efficient MPC protocols where each honest party can be leveraged to increase efficiency?
Our results

New dishonest majority protocols exploiting more honest parties:

1. Passive GMW-style MPC based on Oblivious Transfer.
   • Up to 25x less communication compared with $n - 1$ corruptions.

2. Passive constant-round MPC from Garbled Circuits.
   • Up to 7x reduction in GC size and communication cost.
   • More efficient online phase: Up to 3x faster implementation (circuit-dependent).

Best improvements with 20+ parties when 10-30% are honest.
Introducing the TinyKeys technique
Warm-up: Distributed Encryption

\[
\text{Enc}(\kappa, \ldots, \kappa, \kappa) = \sum_{i} H_{i}(\kappa) + \approx
\]
Distributed Encryption with TinyKeys
Distributed Encryption with TinyKeys

\[ \text{Enc}(k_1, \ldots, k_n, \ell) = \sum_{i} H_i(k_i) + \approx \]
Breaking security

\(? \approx H_1(\ldots) + \cdots + H_h(\ldots)\)

$2^\ell$ keys
Breaking security

\[ \approx \quad H_1(\text{key}) + \cdots + H_h(\text{key}) \]
Breaking security

\[ (0, 0) \quad \cdots \quad (1, 0) \quad + \cdots + H(\ell) \]

Length $2^\ell$, Hamming Weight 1

\[ e_1 \]

Eduardo Soria-Vazquez
Breaking security
Breaking security

\[ \approx \]

\[ (0)^H \ldots (k)^H \]

\[ + \ldots + H_h(\ldots) \]

Length 2\(^{\ell}\)
Hamming Weight 1

\[ e_1 \]

\[ \vdots \]
Breaking security

Eduardo Soria-Vazquez
Breaking security

**Adv wins:** Given $H$ and $y = He$, distinguish $y$ from random
Breaking security: Regular Syndrome Decoding

Sample random $H \in \{0,1\}^{r \times m}$, and regular $e \in \{0,1\}^m$ of weight $h$

Adv wins: Given $H$ and $y = He$, find $e \iff$ distinguish $y$ from random.
Hardness of Regular Syndrome Decoding

• Used for SHA-3 candidate FSB [Augot Finiasz Sendrier 03]
  • Not much easier than syndrome decoding $\Leftrightarrow$ LPN
• Params: Message length $r$, key length $\ell$, #honest $h$.
  • Statistically hard for small $r$/large $h$. [FS09]

[Saa07] [MO15] [NCB11] [Kir11] [BJMM12]
[BM17] [CJ04] [BLP08] [MS09] [MMT11]
[BLN+09] [BLP11]
TinyKeys: A little honesty goes a long way

- Key length: $\ell \geq 1$
- Rest of the talk

- Key length: $\ell \geq 5$
- Many challenges:
  - High Fan-Out
  - Enabling FreeXOR
Quick recap of GMW

\[ x = x_1 + \ldots + x_n \in \{0,1\} \]
\[ y = y_1 + \ldots + y_n \in \{0,1\} \]
\[ x + y = (x_1 + y_1) + \ldots + (x_n + y_n) \]

\[ x \land y = (x_1 + \ldots + x_n) \cdot (y_1 + \ldots + y_n) \]

1-out-2 Bit OT

\[ r, r + y_j \in \{0,1\} \]

Eduardo Soria-Vázquez
“IKNP” OT extension with short keys!

Shrink the keys!

2\^\ell keys

\( b \in \{0,1\}^{r} \)

\( r \times 1\text{-out-2 Bit OTs} \)

\( (X_0^1, X_1^1), \ldots, (X_0^r, X_1^r) \in \{0,1\}^2 \)

\( L(b) \approx H(\cdot(b)) + b \)
Using leaky OT for GMW-style MPC

Sharings of zero:
\[ \sum_{i} s_{ij} = 0 \]

\[
x \land y = (x_1 + \cdots + x_n) \cdot (y_1 + \cdots + y_n)
= \sum_{j=1..n} (x_1 + \cdots + x_n) \cdot y_j
+ \sum_{j=1..n} (s_{1,j} + \cdots + s_{n,j}) \cdot y_j
\]

\[
H_1(k_{1,j}) + x_1 + s_{1,j} \approx \text{leaky OT}
H_2(k_{2,j}) + x_2 + s_{2,j} \approx \text{leaky OT}
\]

\[
\vdots
\]

\[
+ H_h(k_{h,j}) + x_h + s_{h,j} \approx \text{leaky OT}
\]

\[
\sum_{i=1..h} H_i(k_{i,j}) + x_i + s_{ij} \approx \text{leaky OT}
\]

\[
H_1(\ell) + \cdots + H_h(\ell) + \ell
\]

Eduardo Soria-Vázquez
GMW: Communication compl. (200 parties)
Conclusion and future directions

• New technique for distributing trust (more honesty ⇒ shorter keys).
• Improved protocols with 20+ parties.
  • GMW: Up to 25x in communication (vs multiparty [DKSSZZ17]).
  • BMR: Up to 7x in communication (vs [BLO16]). Online phase up to 3x faster.

Follow-up work: Active Security – TinyKeys for TinyOT (Asiacrypt ’18).

Future challenges:
• Optimizations, more cryptanalysis (conservative parameters at the moment).
• More applications,
Thank you! Questions?

Paper:  [Full version]
TinyKeys: A New Approach to Efficient Multi-Party Computation
Carmit Hazay, Emmanuela Orsini, Peter Scholl and Eduardo Soria-Vázquez

Mail:  eduardo.soria-vazquez@bristol.ac.uk
BMR: Multi-party garbled circuits

Each $P_i$ gets $A^i_0, A^i_1 \in \{0,1\}^*$ etc

Use distributed encryption: $E_{A,B}(C) = H(1 \| A^1 \| B^1) + \ldots + H(n \| A^n \| B^n) + (C^1, \ldots, C^n)$

For hash function $H : \{0,1\}^* \rightarrow \{0,1\}^{n\ell}$

Eduardo Soria-Vázquez
**BMR: Some technical challenges**

- **Reusing keys** reduces security in regular syndrome decoding
- **Problem for:**
  - High fan-out
  - Free-XOR
- **Solution:**
  - Splitter gates [Tate Xu 03] – can be garbled for free
  - Free-XOR enabled using different offsets (FleXOR style [CITE])

Eduardo Soria-Vázquez
Thank you! Questions?

Paper:  https://ia.cr/2017/214  [Full version]
TinyKeys: A New Approach to Efficient Multi-Party Computation
Carmit Hazay, Emmanuela Orsini, Peter Scholl and Eduardo Soria-Vázquez

Mail:  eduardo.soria-vazquez@bristol.ac.uk