Bounded-Communication Leakage Resilience via Parity-Resilient Circuits

Vipul Goyal ¹  Yuval Ishai ²,³  Hemanta K. Maji ⁴
Amit Sahai ³  Alexander A. Sherstov ³

¹Microsoft Research, India
²Technion
³University of California, Los Angeles
⁴Purdue University

October 14, 2016 (FOCS–2016)
Motivation: Delegating Computation to Two Servers

Client with input $x$

$\hat{x}_1$ $\hat{x}_2$

$\hat{y}_1$ $\hat{y}_2$

c-bits

Client computes output $y$
## Assumptions on Viruses

<table>
<thead>
<tr>
<th>Assumptions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. <strong>Passive</strong>: Do not tamper with the server messages</td>
<td></td>
</tr>
<tr>
<td>2. <strong>Bounded Communication</strong>: Only $c$-bits of virus communication</td>
<td></td>
</tr>
</tbody>
</table>

### Justification

Virus Detection Mechanisms make tampering server messages and large communication between viruses difficult.

### Note

Viruses can store the entire server view before communicating.
Related Problem 1: Delegation to Single Server

Client with input $x$ computes $y$.

Solution

Fully Homomorphic Encryption [Gentry–09]

Concerns

- Quite far from practical
- Relies on a relatively narrow class of cryptographic hardness assumptions
- No information-theoretic analogue
Related Problem 2: Non-communicating Viruses

Solution
Secure Two-party Computation
[Yao–82, Goldreich–Micali–Wigderson–87]

Features
- Information-theoretic Security using OT or correlated private randomness
- Computational Security based on general cryptographic assumptions

Primary Concern
Yao and GMW are insecure even for 1-bit virus communication
Definition (Bounded Communication Leakage Resilience)

A $c$-BCL-resilient protocol delegates a computation to two servers, such that any $c$-bounded communication leakage reveals essentially nothing about the input.

Theorem (Our Main Result: Informal)

Given an $n$-bit input/output circuit $C_f$ of

- size $s$, and depth $h$

We construct a $c$-BCL-resilient protocol such that:

- Client is implemented by a circuit of size $\tilde{O}(n + c)$
- Servers are implemented by a circuit of size $\tilde{O}(s + ch + c^2)$
- Information-theoretic security given OT
- Computational security based on standard cryptographic assumptions
Comparison to Previous Work (1)

[Dziembowski–Faust–12]
Information-theoretic 2-server Solution using “Leak-free Hardware”

Drawback
The size of the “Leak-free Components” depends on the leakage bound and the statistical security parameter

Feature of our solution
The size of “Leak-free Components” (Oblivious Transfer functionality, which is minimal) is constant
  • Crucial to instantiating our construction with standard cryptographic assumptions
Information-theoretic solution using large-number of servers

Drawback
The number of servers is large

Feature of our solution
A 2-server solution (which is minimal)

Drawback


Feature of our solution

Milder cryptographic hardness assumptions like the intractability of factoring Blum Integers and the Decisional Diffie Hellman
Legend:
- Circuit size of an implementation of $f$: $s$
- Circuit size of BCL-resilient Protocol: $S$
- Bound on the communication complexity of viruses: $c$

Previous Works: Computational Overhead

Computational Overhead $S/s \geq c$

Our Solution: Computational Overhead

Computational Overhead $S/s = \text{polylog } c$, where $c \approx s^{1/2}$
Key Technical Idea: The Beginning

Two Distributions

- Let $\mu$ be a $\varepsilon$-biased distribution
- Let $R$ be a distribution with $(n - c)$ min-entropy

Theorem (Small-Bias Masking [Dodis–Smith–05])

$$\text{SD} (\mu + R, U_n) \leq 2^{c/2}\varepsilon$$
Reformulation in Two-Server Model

### Two Distributions
- Let $\mu$ be a $\varepsilon$-biased distribution
- Let $R$ be a uniform distribution over $n$-bit strings

### Two-server setting
- View of Server 1 is $R$, and View of Server 2 is $\mu + R$
- Virus 1 sends one $c$-bit message $L = \mathcal{L}(R)$ to Virus 2

### Note
$R$ conditioned on the leakage $L$ has high average min-entropy:
$$\tilde{H}_\infty(R|L) \geq (n - c)$$

### Theorem (Small-Bias Masking [Dodis–Smith–05])
$$\text{SD} \left( (\mu + R, L), (U_n, L) \right) \leq 2^{c/2}\varepsilon$$

*Virus 2’s view looks essentially random*
Generalization Goal

Two Directions

- Generalize “$\varepsilon$-bias” to “$\varepsilon$-indistinguishability”
  - Let $\mu_0$ and $\mu_1$ be two distributions that are indistinguishable by linear tests
  - We want: $(\mu_0 + R, L)$ and $(\mu_1 + R, L)$ to look similar
- Generalize “one-round $c$-bit message” by “arbitrary $c$-bit communication”
General Small-bias Masking

Theorem (Generalized Small-bias Masking)

Let $\mu_0$ and $\mu_1$ be probability distribution that are $\varepsilon$-indistinguishable by linear tests. Then a $c$-bit communication protocol $\pi$ that outputs a bit obeys:

$$\left| \mathbb{E}_{w \sim \mu_0, r \leftarrow \{0,1\}^n} [\pi(r, w + r)] - \mathbb{E}_{w \sim \mu_1, r \leftarrow \{0,1\}^n} [\pi(r, w + r)] \right| \leq 2^{c/2\varepsilon}$$
What we achieved: Reduction to Parity-Resilient Circuit

\[ \mu_0 \equiv C[x_0] \longrightarrow \text{If Indistinguishable By Linear Tests} \longrightarrow \mu_1 \equiv C[x_1] \]

Server 1 View

\[ R \]

Server 2 View

\[ \mu_0 + R \]

\[ \pi(R, \mu_0 + R) \]

\[ \pi(R, \mu_1 + R) \]

Then Indistinguishable
Algorithms \((I', C', O')\) such that

\[
\begin{align*}
  x & \rightarrow \hat{x} & \text{Client Encodes using } I' \\
  & \rightarrow \hat{y} & \text{Evaluation of Private Circuit } C' \\
  & \rightarrow y & \text{Client Decodes using } O'
\end{align*}
\]

Definition (Private Circuits)

Probing \(k\)-wires of \(C'\) reveals nothing about the client input \(x\)
Algorithms \((I, C, O)\) such that

\[
\begin{align*}
\text{Client Encodes using } I & \quad \rightarrow \quad \hat{x} \\
\text{Evaluation of Parity-Resilient Circuit } C & \quad \rightarrow \quad \hat{y} \\
\text{Client Decodes using } O & \quad \rightarrow \quad y
\end{align*}
\]

Definition (Parity-Resilient Circuits)

Parity of wire-values of any subset of wires of \(C\) reveals nothing about the client input \(x\)

Construction of \(C\) from \(C'\)

Every wire \(w\) in \(C'\) is encoded as 3 wires in \(C\) whose majority is \(w\)

Caution

The actual encoding used in the paper is slightly more complicated than what is presented here. This complication is necessitated due to the fact that the randomness used to encode the wire \(w\) is also present in the circuit \(C\)
Parity-resilient Circuit: The NAND-Gadget

NAND-Gadget: An 8-bit input and 3-bit output Function

\[
\begin{align*}
\hat{y}_0 &= \text{Maj}(\text{Encoder}(x_1, x_2)) \\
\hat{y}_1 &= \text{Maj}(\text{Encoder}(x_1, x_2)) \\
\hat{y}_2 &= \text{Maj}(\text{Encoder}(x_1, x_2))
\end{align*}
\]
## Why does it work?

- *Small* parity tests are fooled by the privacy guarantee.
- *Big* parity tests are fooled because the XOR of a large number of independent & small-biased bits is close to uniform.
Overall Construction

- Private Circuits
- Construction of Small-bias Distribution
- Parity-resilient Circuits using small trusted-hardware
- Generalization of Small-bias Masking
- BCL-Resilient Protocol using small trusted-hardware
- Joint Simulation Security
- BCL-Resilient Protocol using OT
- Non-committing Encryption
- Computational BCL-Resilient Protocol
Thank You!

Open Problems
- Continual Leakage Setting
- Information-theoretic construction for 3-Servers in the plain model

Summary of Our Construction

Private Circuits
- Construction of Small-bias Distribution
- Parity-resilient Circuits using small trusted-hardware
  - Generalization of Small-bias Masking
  - BCL-Resilient Protocol using small trusted-hardware
    - Joint Simulation Security
    - BCL-Resilient Protocol using OT
      - Non-committing Encryption
      - Computational BCL-Resilient Protocol