Multi-Clients Verifiable Computation via Conditional Disclosure of Secrets

Rishabh Bhadauria
Carmit Hazay
Bar-Ilan University
Conditional Disclosure of Secrets (CDS) [GIKM00]

Function $f$: $\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$

- **Correctness**: $s$ is recovered if $f(x_1,x_2) = 1$
- **Secrecy**: $\text{Sim}(x_1,x_2) \approx (m_1,m_2)$ if $f(x_1,x_2) = 0$

Alice

$\begin{align*}
x_1 &\in \{0,1\}^n \\
m_1 &\leftarrow \text{Private Randomness } r \\
\text{Secret } s
\end{align*}$

Claire

Bob

$\begin{align*}
x_2 &\in \{0,1\}^n \\
m_2 &\leftarrow \text{Secret } s
\end{align*}$
Motivation for CDS

Private Information Retrieval [GIKM00]

Secret-Sharing [BIKK14, LV18, BP18, ABFNP19, ABNP20]

Attributed-based Encryption [Attrapadung14, Wee14, GKW15]
Verifiable Computation from CDS [PRV12]

$$f: \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$$

Server

$$CDS_f \quad CDS_{\overline{f}}$$

$$x_1 \in \{0,1\}^n$$  $$x_2 \in \{0,1\}^n$$

$$r,s_0,s_1$$

Alice  Bob
Multi-Clients Verifiable Computation (MVC) [CCKC13]

- Correctness: $y$ is computed correctly by server
- Privacy (opt.): Nothing about inputs other than $y$ is revealed

Function $f$ is learned by

- Server
- Alice
- Bob

Client inputs $x_1$ and $x_2$ lead to $y=f(x_1,x_2)$.
Example of CDS

\[ f(x_1, x_2) = x_1 \oplus x_2 \]

\[ s = m_1 - m_2 \]

\[ m_1 = s + r_{x_1} \]

\[ m_2 = r_{1-x_2} \]

\[ r = (r_0, r_1) \]

Secret s

\[ x_1 \in \{0,1\}^n \]

\[ x_2 \in \{0,1\}^n \]
Our Results

Explore connection between MVC and CDS

Extend definition of CDS: Private CDS & Oblivious CDS

Construct new classes of private CDS for equality, inequality, private set-intersection (PSI) cardinality and more
Advantages of using CDS

Non-interactive solutions

Batching

Transparent setup
Variants of CDS

**Private CDS** *(Input privacy)*

**Correctness**: The secret $s$ is recovered if $f(x_1, x_2) = 1$

**Privacy**: Claire learns nothing about $x_1$ and $x_2$ other than what is revealed by $f(x_1, x_2)$

**Oblivious CDS** *(Input & output privacy)*

**Correctness**: The secret $s$ is recovered if $f(x_1, x_2) = 1$

**Obliviousness**: Claire learns nothing about $x_1$, $x_2$, or $f(x_1, x_2)$
Oblivious CDS for Equality

\[ f: \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\} \]

\[ m_1 = r_1 x_1 + r_2 + s \]

\[ m_2 = r_1 x_2 + r_2 \]

\[ s' = m_1 - m_2 \]

Alice

Bob

Private Randomness R

Secret s

\[ x_1 \in \{0,1\}^n \]

\[ x_2 \in \{0,1\}^n \]
Sigma Protocol

3-round public coin zkPOK (Zero-Knowledge Proof of Knowledge)

Prover proves that it knows a witness \( w \) s.t. \( (x,w) \in R \)

Verifier’s randomness is public
Schnorr’s Protocol

$x = g^w$

$a = g^r$

$e$

$z = r + ew \mod q$

V checks:

$g^z = ax^e$
Private CDS for Equality

Parties embed their inputs into the transcript of Sigma protocol

Security argument utilizes the special soundness property to extract the secret

\[ m_1 = (x, a_1, e_1, z_1) \]

\[ m_2 = (x, a_2, e_2, z_2) \]

\[ f: \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\} \]

\[ s = \text{Ext} (m_1, m_2) \]

\[ x_1 \in \{0,1\}^n \]

\[ x_2 \in \{0,1\}^n \]

\[ \text{Secret } s = w \]
Private CDS for Inequality

Parties embed their inputs into the transcript of Sigma protocol

Security argument utilizes the special soundness property to extract the secret

\[ f: \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\} \]

Claire

\[ s = \text{Ext}(m_1, m_2) \]

Alice

\[ m_1 = (x, a_1, e_1, z_1) \]

Bob

\[ m_2 = (x, a_2, e_2, z_2) \]

Private Randomness R

Secret \( s = w \)

\[ x_1 \in \{0,1\}^n \]

\[ x_2 \in \{0,1\}^n \]
Additional CDS Constructions

CDS for PSI cardinality based on protocols from [LRG19]

Techniques can be extended to set-union cardinality, set-membership and small domain range predicates

Prior work [PTT11, CPPT14] relies on stronger assumptions but achieves additional properties
Verifiable Computation from CDS [PRV12]

\[ f: \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\} \]

Server

New VC based on our CDS
Malicious security

CDS_\text{f} \quad \text{CDS}_{\bar{f}}

\[ r, s_0, s_1 \]

Alice

x_1 \in \{0,1\}^n

Bob

x_2 \in \{0,1\}^n

\[ s_y, s'_0, s'_1 \]
# 2-Client Verifiable Computation for Equality

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[Cou18]</td>
<td>Correlated*</td>
<td>( \geq 3 )</td>
<td>( O(\lambda \ell) )</td>
<td>( 3\ell + o(\ell) )</td>
<td>OWF + OT</td>
<td>Passive</td>
</tr>
<tr>
<td>[MR18]</td>
<td>Correlated</td>
<td>3</td>
<td>( O(\lambda \ell) )</td>
<td>( O(\lambda \ell) )</td>
<td>OWF+OT</td>
<td>Active</td>
</tr>
<tr>
<td>[BGI19]</td>
<td>Correlated**</td>
<td>2</td>
<td>( \ell )</td>
<td>( \lambda \ell )</td>
<td>OWF</td>
<td>Passive</td>
</tr>
<tr>
<td>Our Work</td>
<td>Uniform</td>
<td>2</td>
<td>( 3\ell )</td>
<td>( 6\ell )</td>
<td>OWF</td>
<td>Passive</td>
</tr>
<tr>
<td>Our Work</td>
<td>Uniform</td>
<td>2</td>
<td>( 10\ell )</td>
<td>( 7\ell )</td>
<td>OWF+\Sigma-protocol***</td>
<td>Active</td>
</tr>
</tbody>
</table>

Table 1: A comparison of our equality protocol with prior work where \( \lambda \) is the security parameter, the inputs are of size \( \ell \) bits and OT is oblivious transfer.

* This work uses two types of correlated randomness that are generated using OT for XOR and AND shares.

** This correlation requires keys for computing distributed point functions.

*** We concretely rely here on the hardness of discrete logarithm in groups.
Thank You