Compressed $\Sigma$-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics

Thomas Attema$^{1,2,3}$ Ronald Cramer$^{1,2}$

August 19, 2020

$^1$CWI - Cryptology Group
$^2$Leiden University- Mathematical Institute
$^3$TNO - Cyber Security and Robustness Department
Background

\textbf{Σ-Protocol Theory,}
- Well-established basis for zero-knowledge proofs,
- Linear communication complexity for circuit ZK (linear in circuit size).

\textbf{Bulletproofs} [BCC\textsuperscript{+}16, BBB\textsuperscript{+}18],
- Logarithmic communication complexity for circuit ZK.
- \textbf{Note:} Presented as a \textit{drop-in replacement} for Σ-protocols.

\textbf{This Work:}
Reconciling Bulletproofs with Σ-Protocol Theory:
\textit{Strengthening} of Σ-protocol theory, rather than replacement.
Solve linear instances first, and then linearize the non-linear instances.

- Natural mathematical problem solving strategy.
- Fits seamlessly with established $\Sigma$-protocol theory.
- **Note**: Bulletproofs pivotal protocol is a PoK for a *quadratic* relation.
Our High-Level Approach (2/2)

Our observations:

1. Adaptation of Bulletproofs gives compression for standard $\Sigma$-protocols.
   - \textit{Compact} commitment to long vectors.
   - \textit{Logarithmic} size HVZK PoK for opening arbitrary linear forms.

2. [CDP12]-Adaptation: \textit{Compact} commitment to long vectors of mult.-triples.
   - \textit{Logarithmic} size proof of correctness.
   - Method combines \textit{arithmetic secret sharing} with point 1.

3. Specialized reduction from circuit ZK to verifying multiplication triples.
   - Reduction combines point 1. and point 2.

\textbf{Note:} \textit{Constant} communication if instantiated from \textit{Knowledge of Exponent Assumption}.
Let \([\cdot] : \mathbb{Z}_q^n \rightarrow \mathbb{G}\) be a commitment scheme for some group \(\mathbb{G}\) (randomness implicit).

Let \(C : \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q\) be an arithmetic circuit.

For a public commitment \([x]\), a prover wishes to prove knowledge of

- a commitment opening \(x \in \mathbb{Z}_q^n\),
- such that \(C(x) = 0\).
- Honest Verifier Zero-Knowledge (HVZK).

We focus on the communication efficiency of the protocols.
Our Main Pivot - *Opening Linear Forms on Compactly Committed Vectors*

Compressing the Pivot

[CDP12]-adaptation to prove multiplicative relations

Circuit Zero-Knowledge

Auxiliary protocols for practical deployment: Compactification

Instantiations from various cryptographic assumptions

Recent Work: Proofs of Partial Knowledge
We consider multi-exponentiations:

\[ g^x := \prod g_i^{x_i} \in \mathbb{G}. \]  (1)

We assume that the prover does not know a non-trivial discrete log relation, i.e., an

\[ x \in \mathbb{Z}_q^n \setminus \{(0, \ldots, 0)\}, \text{ such that } g^x = 1. \]  (2)
A Pedersen (vector) commitment \([x]\) is a multi-exponentiation:

\[
\text{COM} : \mathbb{Z}_q^n \times \mathbb{Z}_q \rightarrow \mathbb{G}, \quad (x, \gamma) \mapsto g^x h^\gamma = \prod_{i=1}^{n} g_i^{x_i} h^\gamma
\]  

Moreover, it is

- Compact
- Unconditionally hiding
- Computationally binding under the discrete logarithm assumption
  - More precisely, the prover should not know a non-trivial discrete log relation
Let $L : \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q$ be a linear form.

$$R = \{(P \in \mathbb{G}, L \in \mathcal{L}(\mathbb{Z}_q^n), y \in \mathbb{Z}_q; x \in \mathbb{Z}_q^n) : g^x = P \land L(x) = y\}.$$  \hspace*{2cm} (4)

The protocols trivially generalize to opening *affine forms*,

\begin{itemize}
  \item i.e., linear forms plus a constant.
\end{itemize}
Pivotal Σ-Protocol for Opening Linear Forms (2/2)

**Input** \((P, L, y; x)\)

- \(P = g^x \in G\)
- \(y = L(x) \in \mathbb{Z}_q\)

**Prover**

\[ r \leftarrow_R \mathbb{Z}_q^n \]

\[ t = L(r), A = g^r \]

\[ z = cx + r \]

**Verifier**

\[ c \leftarrow_R \mathbb{Z}_q \]

**Proof**

\[ g^z \overset{?}{=} AP^c \]

\[ L(z) \overset{?}{=} cy + t \]
Presentation Outline

1. Our Main Pivot - *Opening Linear Forms on Compactly Committed Vectors*
2. Compressing the Pivot
3. [CDP12]-adaptation to prove multiplicative relations
4. Circuit Zero-Knowledge
5. Auxiliary protocols for practical deployment: Compactification
6. Instantiations from various cryptographic assumptions
7. Recent Work: Proofs of Partial Knowledge
INPUT($P, L, y; x$)

$P = g^x \in \mathbb{G}$
$y = L(x) \in \mathbb{Z}_q$

Prover

$r \leftarrow_R \mathbb{Z}_q^n$
$t = L(r), A = g^r$
$z = cx + r$

Verifier

$c \leftarrow_R \mathbb{Z}_q$

$g^z \overset{?}{=} AP^c$
$L(z) \overset{?}{=} cy + t$
Compressing the $\Sigma$-Protocol

The response $z \in \mathbb{Z}_q^n$ in $\Pi_0$ is a witness of another element in the same relation $R$, i.e.,

$$(\delta, L, cy + t; z) \in R = \{(P, L, y; x) : g^x = P \land L(x) = y\}. \quad (5)$$

Therefore, the final message $z$ is a (trivial) PoK for witness $z$.

- Any PoK for this relation suffices.
- It does not have to be zero-knowledge.

Notation:

- $g = (g_1, \ldots, g_n) \in G^n \Rightarrow g_L := (g_1, \ldots, g_{n/2}) \in G^{n/2}$.
- $L : \mathbb{Z}_q^n \to \mathbb{Z}_q$, $x \mapsto L(x, 0, \ldots, 0)$
Compression Mechanism $\Pi_1$

**Prover**

\[
P = g^x \land L(x) = y
\]

\[
A = g^{x_L}, \quad B = g^{x_R}
\]

\[
a = L_R(x_L), \quad b = L_L(x_R)
\]

\[
z = x_L + cx_R
\]

**Verifier**

\[
\begin{align*}
A, B, a, b &\quad \rightarrow \quad c \leftarrow_R \mathbb{Z}_q \\
\end{align*}
\]

\[
\begin{align*}
g' &:= g_L^c \ast g_R \\
L' &:= cL_L + L_R \\
\end{align*}
\]

\[
(g')^z \overset{?}{=} A^c P B c^2
\]

\[
L'(z) \overset{?}{=} a + yc + bc^2
\]
Given a challenge $c$, the prover encodes the secret exponent $x$.

$$\text{Enc}_c : \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q^n, \quad x \mapsto (0, x_L) + cx + c^2(x_R, 0).$$

Note that:

$$\text{Enc}_c(x) = (cz, z), \quad \text{where } z = (x_L + cx_R).$$

Hence, $\dim(\text{Enc}_c(\mathbb{Z}_q^n)) = n/2$ for any $c \in \mathbb{Z}_q$.

- The communication complexity is roughly halved.

Given 3 different encodings $\text{Enc}_{c_1}(x)$, $\text{Enc}_{c_2}(x)$ and $\text{Enc}_{c_3}(x)$, $x$ can be reconstructed.

- The protocol is 3-special sound.
Recursive composition of \( \Sigma \)-Protocol \( \Pi_0 \) and compression protocol \( \Pi_1 \).

\[ \Pi_c := \Pi_0 \diamond \Pi_1 \diamond \cdots \diamond \Pi_1. \]

\( \Pi_c \) is a SHVZK PoK for relation:

\[ R = \{(P, L, y; x) : g^x = P \land L(x) = y\}. \]

\( \Pi_c \) has logarithmic communication complexity.

We say \( \Pi_c \) is a compressed \( \Sigma \)-Protocol.
1. Our Main Pivot - *Opening Linear Forms on Compactly Committed Vectors*

2. Compressing the Pivot

3. [CDP12]-adaptation to prove multiplicative relations

4. Circuit Zero-Knowledge

5. Auxiliary protocols for practical deployment: Compactification

6. Instantiations from various cryptographic assumptions

7. Recent Work: Proofs of Partial Knowledge
We have seen how to solve linear instances,
- Now we linearize to handle with non-linear instances.

We use an adaptation from [CDP12] which uses a packed secret sharing scheme that has
- strong multiplicativity,
- 1-privacy.

We consider Shamir’s secret sharing scheme.
Verifying multiplicative relations efficiently.

\[ a = (a_1, \ldots, a_m), \quad b = (b_1, \ldots, b_m), \quad c = a \ast b. \]

\[ f(X) \in \mathbb{Z}_q[X]_{\leq m} \quad \text{s.t.} \quad f(i) = a_i \quad \forall 1 \leq i \leq m \quad \text{and} \quad f(0) = r_a \leftarrow_R \mathbb{Z}_q. \]

\[ g(X) \in \mathbb{Z}_q[X]_{\leq m} \quad \text{s.t.} \quad g(i) = b_i \quad \forall 1 \leq i \leq m \quad \text{and} \quad g(0) = r_b \leftarrow_R \mathbb{Z}_q. \]

\( f(X) \) and \( g(X) \) define packed secret sharings of \( a \) and \( b \):

- Any \( m + 1 \) distinct evaluations allow reconstruction.
- Any evaluation outside \((1, \ldots, m)\) is uniformly random (1-privacy).
We define the product polynomial $h(X) := f(X)g(X) \in \mathbb{Z}_q[X]_{\leq 2m}$

- $h(i) = c_i \quad \forall 1 \leq i \leq m.$
- $h(X)$ is defined by any $2m + 1$ evaluations.
- $h(X)$ defines a secret sharing of $c$ (strong multiplicativity).

For a uniformly random $\alpha \in \mathbb{Z}_q \setminus \{1, \ldots, m\}$:

- $f(\alpha), g(\alpha), h(\alpha)$ is a uniformly random multiplication triple.
- $f(\alpha)g(\alpha) = h(\alpha)$ implies $f(X)g(X) = h(X)$ and $a \ast b = c$ with probability at least $1 - 2m/(q - m)$. 
Protocol for proving multiplicative relations:

1. Combination with our compressed Σ-protocol for opening linear forms.

2. Prover commits to the (coefficients of the) polynomials $f(X)$, $g(X)$ and $h(X)$ in a single compact commitment.

3. Prover opens $f(\alpha)$, $g(\alpha)$ and $h(\alpha)$ for a random challenge $\alpha$.
   - The evaluations are all linear combinations of the committed coefficients.
Presentation Outline

1. Our Main Pivot - *Opening Linear Forms on Compactly Committed Vectors*
2. Compressing the Pivot
3. [CDP12]-adaptation to prove multiplicative relations
4. Circuit Zero-Knowledge
5. Auxiliary protocols for practical deployment: Compactification
6. Instantiations from various cryptographic assumptions
7. Recent Work: Proofs of Partial Knowledge
Circuit Zero-Knowledge

Textbook Scenario:
- Commit to a vector $x$ (plus auxiliary information).
- Prove that $C(x) = 0$.

Protocol:
1. Prover defines $f(X), g(X), h(X)$ as packed secret sharings of the inputs and outputs of the multiplication gates of $C$.
2. Prover commits to $(x, aux) := (x, f(0), g(0), h(0), \ldots, h(2m))$.
3. Verifier asks the prover to open $C(x), f(\alpha), g(\alpha)$ and $h(\alpha)$ for random challenge $\alpha \in \mathbb{Z}_q \setminus \{1, \ldots, m\}$.
4. Verifier checks that $C(x) = 0$ and $f(\alpha)g(\alpha) = h(\alpha)$.
1. Our Main Pivot - *Opening Linear Forms on Compactly Committed Vectors*

2. Compressing the Pivot

3. [CDP12]-adaptation to prove multiplicative relations

4. Circuit Zero-Knowledge

5. Auxiliary protocols for practical deployment: Compactification

6. Instantiations from various cryptographic assumptions

7. Recent Work: Proofs of Partial Knowledge
Compactification is required for most interesting practical applications.

- Combining various commitments in a single compact commitment.
- Efficiently handles practical ZK scenarios, e.g.,
  1. Prover is already compactly committed to input $x$.
  2. Input $x$ is dispersed over multiple commitments.

Additionally: Various amortization techniques to further improve efficiency.
Presentation Outline

1. Our Main Pivot - *Opening Linear Forms on Compactly Committed Vectors*
2. Compressing the Pivot
3. [CDP12]-adaptation to prove multiplicative relations
4. Circuit Zero-Knowledge
5. Auxiliary protocols for practical deployment: Compactification
6. Instantiations from various cryptographic assumptions
7. Recent Work: Proofs of Partial Knowledge
We show how to instantiate this framework from *three different cryptographic assumptions*:

- Discrete Log Assumption.
- Strong-RSA assumption (following the approach of [BFS20]).
- Knowledge of Exponent Assumption (following the approach of [Gro10]).
  Achieves *constant* communication as in ZK-SNARKS.
Our Main Pivot - *Opening Linear Forms on Compactly Committed Vectors*

1. Compressing the Pivot

2. [CDP12]-adaptation to prove multiplicative relations

3. Circuit Zero-Knowledge

4. Auxiliary protocols for practical deployment: Compactification

5. Instantiations from various cryptographic assumptions

6. Recent Work: Proofs of Partial Knowledge
Recent Work

Adaptation of the compressed $\Sigma$-protocol gives a *proof of partial knowledge*.

- Proving knowledge of $k$-out-of-$n$ discrete logarithms.
- Follow-up work available on ePrint [ACF20].
Thanks!


