# 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

**Σ-Protocol Theory**,

- Well-established basis for zero-knowledge proofs,
- Linear communication complexity for circuit ZK (linear in circuit size).

**Bulletproofs** [BCC+16, BBB+18],

- Logarithmic communication complexity for circuit ZK.
- **Note**: Presented as a *drop-in replacement* for Σ-protocols.

**This Work:**

Reconciling Bulletproofs with Σ-Protocol Theory:
*Strengthening* of Σ-protocol theory, rather than replacement.

# Our High-Level Approach (1/2)

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.
   - *Compact* commitment to long vectors.
   - *Logarithmic* size HVZK PoK for opening arbitrary linear forms.

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

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

**Note:** *Constant* communication if instantiated from *Knowledge of Exponent Assumption*.

# Setting - Finding Efficient Circuit Zero-Knowledge Protocols

- Let $[\cdot] : \mathbb{Z}_q^n \to \mathbb{G}$ be a commitment scheme for some group $\mathbb{G}$ (randomness implicit).
- Let $C : \mathbb{Z}_q^n \to \mathbb{Z}_q$ be an arithmetic circuit.

For a public commitment $[\mathbf{x}]$, a prover wishes to prove knowledge of

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

We focus on the *communication* efficiency of the protocols.

# 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

# Multi-Exponentiations

- $\mathbb{G}$ is a group of prime order $q$,
- $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{G}^n$ is a vector of generators for $\mathbb{G}$.

We consider multi-exponentiations:

$$\mathbf{g}^{\mathbf{x}} := \prod g_i^{x_i} \in \mathbb{G}. \tag{1}$$

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

$$\mathbf{x} \in \mathbb{Z}_q^n \setminus \{(0, \ldots, 0)\}, \quad \text{such that} \quad \mathbf{g}^{\mathbf{x}} = 1. \tag{2}$$

# Pedersen Vector Commitment Scheme

A Pedersen (vector) commitment $[\mathbf{x}]$ is a multi-exponentiation:

$$\textsc{Com} : \mathbb{Z}_q^n \times \mathbb{Z}_q \to \mathbb{G}, \quad (\mathbf{x}, \gamma) \mapsto \mathbf{g}^{\mathbf{x}} h^{\gamma} = \prod_{i=1}^{n} g_i^{x_i} h^{\gamma} \tag{3}$$

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

# Pivotal Σ-Protocol for Opening Linear Forms (1/2)

Let $L : \mathbb{Z}_q^n \to \mathbb{Z}_q$ be a linear form.

$$R = \{(P \in \mathbb{G}, L \in \mathcal{L}(\mathbb{Z}_q^n), y \in \mathbb{Z}_q; \mathbf{x} \in \mathbb{Z}_q^n) : \mathbf{g}^{\mathbf{x}} = P \wedge L(\mathbf{x}) = y\}. \tag{4}$$

The protocols trivially generalize to opening *affine forms*,

- i.e., linear forms plus a constant.

# Pivotal Σ-Protocol for Opening Linear Forms (2/2)

$$\textsc{Input}(P, L, y; \mathbf{x})$$
$$P = \mathbf{g}^{\mathbf{x}} \in \mathbb{G}$$
$$y = L(\mathbf{x}) \in \mathbb{Z}_q$$

| Prover | | Verifier |
|---|---|---|
| $\mathbf{r} \leftarrow_R \mathbb{Z}_q^n$ | | |
| $t = L(\mathbf{r}), A = \mathbf{g}^{\mathbf{r}}$ | $\xrightarrow{t, A}$ | |
| | | $c \leftarrow_R \mathbb{Z}_q$ |
| | $\xleftarrow{c}$ | |
| $\mathbf{z} = c\mathbf{x} + \mathbf{r}$ | | |
| | $\xrightarrow{\mathbf{z}}$ | |
| | | $\mathbf{g}^{\mathbf{z}} \stackrel{?}{=} AP^c$ |
| | | $L(\mathbf{z}) \stackrel{?}{=} cy + t$ |

# 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

# Pivotal Σ-Protocol for Opening Linear Forms (2/2)

$$\textsc{Input}(P, L, y; \mathbf{x})$$
$$P = \mathbf{g}^{\mathbf{x}} \in \mathbb{G}$$
$$y = L(\mathbf{x}) \in \mathbb{Z}_q$$

$$\begin{array}{ccc}
\text{Prover} & & \text{Verifier} \\
\mathbf{r} \leftarrow_R \mathbb{Z}_q^n & & \\
t = L(\mathbf{r}), A = \mathbf{g}^{\mathbf{r}} & \xrightarrow{\quad t, A \quad} & \\
 & & c \leftarrow_R \mathbb{Z}_q \\
 & \xleftarrow{\quad c \quad} & \\
\mathbf{z} = c\mathbf{x} + \mathbf{r} & & \\
 & \xrightarrow{\quad \mathbf{z} \quad} & \\
 & & \mathbf{g}^{\mathbf{z}} \stackrel{?}{=} AP^c \\
 & & L(\mathbf{z}) \stackrel{?}{=} cy + t
\end{array}$$

# Compressing the $\Sigma$-Protocol

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

$$(AP^c, L, cy + t; \mathbf{z}) \in R = \{(P, L, y; \mathbf{x}) : \mathbf{g}^{\mathbf{x}} = P \wedge L(\mathbf{x}) = y\}. \tag{5}$$

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

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

**Notation:**

- $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{G}^n \quad \implies \mathbf{g}_L := (g_1, \ldots, g_{n/2}) \in \mathbb{G}^{n/2}$.
- $L : \mathbb{Z}_q^n \to \mathbb{Z}_q, \quad \implies \quad L_L : \mathbb{Z}_q^{n/2} \to \mathbb{Z}_q, \quad \mathbf{x} \mapsto L(\mathbf{x}, 0, \ldots, 0)$

# Compression Mechanism $\Pi_1$

$$\textsc{Input}(P, L, y; \mathbf{x})$$
$$P = \mathbf{g}^{\mathbf{x}} \wedge L(\mathbf{x}) = y$$

| Prover | | Verifier |
|---|---|---|
| $A = \mathbf{g}_R^{\mathbf{x}_L}, B = \mathbf{g}_L^{\mathbf{x}_R}$ <br> $a = L_R(\mathbf{x}_L), b = L_L(\mathbf{x}_R)$ | $\xrightarrow{A,B,a,b}$ | |
| | $\xleftarrow{c}$ | $c \leftarrow_R \mathbb{Z}_q$ |
| | $\mathbf{g}' := \mathbf{g}_L^c * \mathbf{g}_R$ <br> $L' := cL_L + L_R$ | |
| $\mathbf{z} = \mathbf{x}_L + c\mathbf{x}_R$ | $\xrightarrow{\mathbf{z}}$ | $(\mathbf{g}')^{\mathbf{z}} \stackrel{?}{=} AP^cB^{c^2}$ <br> $L'(\mathbf{z}) \stackrel{?}{=} a + yc + bc^2$ |

# Compression Mechanism $\Pi_1$

Given a challenge $c$, the prover encodes the secret exponent $\mathbf{x}$.

$$\mathrm{Enc}_c : \mathbb{Z}_q^n \to \mathbb{Z}_q^n, \quad \mathbf{x} \mapsto (0, \mathbf{x}_L) + c\mathbf{x} + c^2(\mathbf{x}_R, 0).$$

Note that:

$$\mathrm{Enc}_c(\mathbf{x}) = (c\mathbf{z}, \mathbf{z}), \quad \text{where } \mathbf{z} = (\mathbf{x}_L + c\mathbf{x}_R).$$

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

- The communication complexity is roughly halved.

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

- The protocol is 3-special sound.

## Our Compressed Pivot

- 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:

$$R = \{(P, L, y; \mathbf{x}) : \mathbf{g}^{\mathbf{x}} = P \wedge L(\mathbf{x}) = y\}.$$

- $\Pi_c$ has logarithmic communication complexity.
- We say $\Pi_c$ is a compressed $\Sigma$-Protocol.

# 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

# Proving Non-Linear Relations

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

1. strong multiplicativity,
2. 1-privacy.

We consider Shamir's secret sharing scheme.

# Multiplicative Relations - Adaptation of [CDP12] (1/3)

Verifying multiplicative relations efficiently.

$$\mathbf{a} = (a_1, \ldots, a_m), \qquad \mathbf{b} = (b_1, \ldots, b_m), \qquad \mathbf{c} = \mathbf{a} * \mathbf{b}.$$

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

$f(X)$ 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).

# Multiplicative Relations - Adaptation of [CDP12] (2/3)

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 $\mathbf{c}$ (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 $\mathbf{a} * \mathbf{b} = \mathbf{c}$ with probability at least $1 - 2m/(q-m)$.

# Multiplicative Relations - Adaptation of [CDP12] (3/3)

Protocol for proving multiplicative relations:

- Combination with our compressed $\Sigma$-protocol for opening linear forms.

1. Prover commits to the (coefficients of the) polynomials $f(X)$, $g(X)$ and $h(X)$ in a single compact commitment.
2. 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 $\mathbf{x}$ (plus auxiliary information).
- Prove that $C(\mathbf{x}) = 0$.

**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
$$(\mathbf{x}, \mathrm{aux}) := (\mathbf{x}, f(0), g(0), h(0), \ldots, h(2m)).$$
3. Verifier ask the prover to open $C(\mathbf{x}), f(\alpha), g(\alpha)$ and $h(\alpha)$ for random challenge $\alpha \in \mathbb{Z}_q \setminus \{1, \ldots, m\}$.
4. Verifier checks that $C(\mathbf{x}) = 0$ and $f(\alpha)g(\alpha) = h(\alpha)$.

# 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

# Auxiliary protocols for practical deployment: Compactification

*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 $\mathbf{x}$.
  2. Input $\mathbf{x}$ 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

# Cryptographic Assumptions

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.

# 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

# 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!

# Bibliography I


Thomas Attema, Ronald Cramer, and Serge Fehr.
Compressing proofs of $k$-out-of-$n$-partial knowledge.
*IACR Cryptol. ePrint Arch.*, 2020:753, 2020.

Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell.
Bulletproofs: Short proofs for confidential transactions and more.
In *IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society, 2018.

Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit.
Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting.
In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 327–357. Springer, 2016.

# Bibliography II


Benedikt Bünz, Ben Fisch, and Alan Szepieniec.
Transparent snarks from DARK compilers.
In *EUROCRYPT (1)*, volume 12105 of *Lecture Notes in Computer Science*, pages 677–706. Springer, 2020.

Ronald Cramer, Ivan Damgård, and Valerio Pastro.
On the amortized complexity of zero knowledge protocols for multiplicative relations.
In *ICITS*, volume 7412 of *Lecture Notes in Computer Science*, pages 62–79. Springer, 2012.

Jens Groth.
Short pairing-based non-interactive zero-knowledge arguments.
In *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 321–340. Springer, 2010.