Compressing Proofs of k-Out-Of-n Partial Knowledge

Thomas Attema and Ronald Cramer and Serge Fehr

August 19, 2020

Proofs of k-out-of-n Partial Knowledge

Prover claims to know k (secret) solutions out of n (public) problem instances.

- Verifier does not learn which k-out-of-n secrets.
- Introduced by Cramer, Damgård and Schoenmakers at CRYPTO'94 [CDS94].

Some examples. A prover claims to know

1. k discrete logarithms (w.r.t. $g \in \mathbb{G}$) of n group elements $h_1, \ldots, h_n \in \mathbb{G}$,
2. the openings of k-out-of-n commitments $P_1, \ldots, P_n \in \mathbb{G}$,
3. the pre-images, w.r.t. some hash function h, of k-out-of-n bit strings $y_1, \ldots, y_n$.

Some Applications:

- Threshold and ring signature schemes,
- E-voting,
- Confidential transaction systems.

Indirect Circuit ZK Approach

Circuit ZK Protocols:

- Prove knowledge of a secret $\mathbf{x} \in \mathbb{Z}_q^n$ that satisfies a constraint $C(\mathbf{x}) = 0$.
- SNARKS, STARKS, Bulletproofs, Compressed Σ-Protocol Theory, ...
- Logarithmic or constant communication complexity.

Circuit ZK based Proofs of Partial Knowledge:

1. Capture the relation by an arithmetic circuit.
   - E.g. (1-out-of-n),
   $$C(x) = 0 \text{ if and only if } \exists i \text{ s.t. } g^x = P_i.$$
2. Apply a circuit ZK protocol.

- Works for arbitrary k and n.
- Logarithmic or even constant communication complexity.

Goal

Some Disadvantages of the Circuit ZK Approach:

- Conceptually superfluous,
- Asymptotically efficient, but practical overhead due to large circuits
  - Capturing group exponentiation, hash function, commitment functions, ...,
  - These circuits have to be generated/stored/processed,
  - A single group exponentiation: $\approx$ 2000 multiplication gates [HBHW20].

A *direct* approach, avoiding the overhead of (large) circuits.

- Arbitrary k and n,
- Logarithmic communication complexity.

Prior work does not achieve our goal

Proofs-of-Partial Knowledge [CDS94]:

- Ingredients: Σ-protocol + SHVZK simulator + linear secret-sharing scheme.
- Arbitrary k and n and works for a broad class of Σ-protocols.
- **Linear** communication complexity.

One-out-of-Many Proofs [GK15]:

- **Special case** $k = 1$. Does not generalize well to arbitrary k.
- Informal Approach: commit to the bits of the index of the known secret.
  - Σ-protocol.
- Logarithmic communication complexity.
- Lattice-based instantiation [ESS$^+$19].

Prior Work: Compressed Σ-Protocol Theory

In addition to [CDS94], we build on compressed Σ-protocol theory [AC20].

Compressed Σ-Protocol Theory [AC20]

- Compression mechanism for a basic Σ-protocol for linear statements.
- Circuit ZK with logarithmic communication complexity.
  - Arithmetic secret sharing to linearize non-linear instances.
- Instantiations from various hardness assumptions; RSIS, DL, Strong-RSA, KEA.

Natural question towards our goal:

Can we compress the proof of partial knowledge Σ-protocol of [CDS94]?

Our work

Compressing proofs of partial knowledge requires:

1. Novel twist on the basic compressed Σ-protocol for linear statements [AC20].
   - Opening arbitrary homomorphisms instead of linear forms.
2. Novel approach to [CDS94].
   - *Compressible* Σ-protocol for proving k-out-of-n partial knowledge.

Theorem (Main Result)

There exists a protocol for proving knowledge of k-out-of-n discrete logarithms. Its communication costs, from prover to verifier, are

$$4\left\lceil \log_2(2n-k+1)\right\rceil - 5 \text{ group elements},$$
$$4 \text{ field elements}.$$

$\Longrightarrow$ Works for arbitrary k and n.

Extensions

- Reducing communication complexity with factor 2 by a pairing based commitment scheme.
- Extension to (vector) commitments and multi-exponentiations.
- Techniques are compatible with circuit ZK.
  - Proving that k-out-of-n secrets satisfy an arbitrary constraint $C(x_1, \ldots, x_k) = 0$.
  - Plug-and-play nature of (compressed) Σ-protocol theory.
- Application: succinct threshold signature with transparent set-up [ACR20].
- Lattice instantiation should be possible.
  - Using lattice-based compressed Σ-Protocols [ACK21].
  - Ingredients: vector commitment scheme, efficient proof for linear relations and secret sharing scheme.

Competitive Concrete Communication Costs

- Comparable to dedicated solutions for the case $k = 1$.
- For $k = \Omega(n/\log n)$: asymptotic improvement over the indirect circuit ZK approach.

[CDS94] Approach

k-out-of-n partial knowledge:

- Public: $P_1, \ldots, P_n \in \mathbb{G}$.
- Secret: $\mathbf{x} \in \mathbb{Z}_q^n$, $S \subset \{1, \ldots, n\}$.
- Such that: $|S| = k$ and $g^{x_i} = P_i \quad \forall i \in S$.

Run n parallel instances of the basic Σ-protocol Π in the following manner:

1. First message:
   - For $i \in S$: the prover runs honest executions of Π.
   - For $i \notin S$: run the simulator for Π. Informally, the prover guesses random challenges c_i.
2. Second message: The verifier sends a single random challenge $c \in \mathbb{Z}_q$.
3. Final response:
   - Prover computes $(n - k + 1, n)$-secret sharing $(c_1, \ldots, c_n)$ of c, such that for $i \notin S$ these challenges correspond to the simulated transcripts.
   - The prover computes the responses for the n instantiations of Π corresponding to the challenges c_i.

Our Twist on a Compressed Σ-Protocol

A central protocol of [AC20]:

- Open *arbitrary linear forms* on compactly committed vectors.
- Logarithmic communication complexity.

More precisely, prove knowledge of commitment opening $\mathbf{x} \in \mathbb{Z}_q^n$ s.t. $L(\mathbf{x}) = y$.

- for public linear form $L \colon \mathbb{Z}_q^n \to \mathbb{Z}_q$ and a public $y \in \mathbb{Z}_q$.

This functionality extends to *opening arbitrary homomorphisms*:

$$f : \mathbb{Z}_q^n \to \mathbb{G}.$$

Cost of this extension: factor 2 increase in communication complexity.

The [CDS94] Proof of Partial Knowledge Σ-Protocol

Even with our generalization the [CDS94] Σ-Protocol is not *compressible*.

- Size of the first message is already linear in n.
- The final response contains a secret sharing $(c_1, \ldots, c_n)$.

Novel technical approach to [CDS94]

Step 1: Reduce the k-out-of-n case to the n-out-of-n case where the prover knows all DLs.

- Eliminating the exponents that the prover does not know.
  - Use an "elimination" vector $(s_1, \ldots, s_n)$ with $s_i = 0$ for all $i \notin S$.

Prove knowledge of the DLs of

$$Q_i := P_i^{s_i} \quad \forall 1 \leq i \leq n.$$

Note that the prover knows the DLs of all Q_i

1. For $i \in S$: $g^{s_i x_i} = Q_i$.
2. For $i \notin S$: $g^0 = Id = Q_i$.

Novel technical approach to [CDS94] - Remaining Challenges 1/2

The verifier can not learn the "elimination" vector $\mathbf{s}$.

- Therefore, we use the protocol for opening homomorphisms.

We define the following homomorphisms, for $1 \leq i \leq n$,

$$f_i : \mathbb{Z}_q^{2n} \to \mathbb{G}, \quad \mathbf{y} \mapsto g^{-y_{i+n}} P_i^{y_i}.$$

Then, prover commits to

$$\mathbf{y} = (s_1, \ldots, s_n, s_1 x_1, ..., s_n x_n) \in \mathbb{Z}_q^{2n}$$

and shows that $\mathbf{y}$ satisfies $f_i(\mathbf{y}) = Id$ for all i.
Note that

$$f_i(\mathbf{y}) = g^{-s_i x_i} P_i^{s_i} = Id.$$

Novel technical approach to [CDS94]- Remaining Challenges 2/2

The vector $\mathbf{s}$ should contain at most $n-k$ zero's.

- We enforce $\mathbf{s}$ to be an $(n-k+1, n)$-secret sharing of 1.

Such a secret sharing can be defined by a polynomial of degree at most $n-k$,

$$p(X) = 1 + \sum_{i=1}^{n-k} a_i X^i.$$

The protocol and homomorphisms are adapted by having the prover commit to

$$\mathbf{y} = (a_1, \ldots, a_{n-k}, p(1)x_1, ..., p(n)x_n) \in \mathbb{Z}_q^{2n-k}$$

instead of

$$\mathbf{y} = (s_1, \ldots, s_n, s_1 x_1, ..., s_n x_n) \in \mathbb{Z}_q^{2n}$$

Proof of k-out-of-n Partial Knowledge

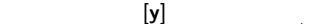
$\textsc{Input}\,(g, P_1, \ldots, P_n, k; S, \mathbf{x})$

$g^{x_i} = P_i$ for $i \in S$

Prover

Verifier

$p(X) = 1 + \sum_{i=1}^{n-k} a_i X^i$ s.t.
$p(i) = 0 \quad \forall i \notin S$
$\mathbf{y} = (a_1, \ldots, a_{n-k},$
$p(1)x_1, \ldots, p(n)x_n)$



Prove that $\mathbf{y}$ satisfies

$$g^{y_{i+n-k}} P_i^{-\sum_j y_j i^j} = P_i \quad \forall\, i \in \{1, \ldots, n\}$$

Communication Costs

The costs of opening n homomorphisms can be amortized.

- The total communication complexity is equal to the communication complexity for opening only one linear form.
- Roughly $4 \log_2(n)$ items have to be sent from the prover to the verifier.

Remark on Extension - Reducing Communication Complexity

A pairing based commitment scheme allows the communication costs to be reduced by a factor 2.

- In [AC20], a prover can commit to $(\mathbf{x}, L(\mathbf{x})) \in \mathbb{Z}_q^{n+1}$; reduces communication.
- Our generalization requires a commitment scheme for *mixed* vectors $(\mathbf{x}, f(\mathbf{x})) \in \mathbb{Z}_q^n \times \mathbb{G}$.
  - Pairing based commitment scheme.

Thanks!

Bibliography I

Thomas Attema and Ronald Cramer.
Compressed sigma-protocol theory and practical application to plug & play secure algorithmics.
In *CRYPTO (3)*, volume 12172 of *Lecture Notes in Computer Science*, pages 513–543. Springer, 2020.

Thomas Attema, Ronald Cramer, and Lisa Kohl.
A compressed Σ-protocol theory for lattices.
IACR Cryptol. ePrint Arch., 2021:307, 2021.

Thomas Attema, Ronald Cramer, and Matthieu Rambaud.
Compressed sigma-protocols for bilinear circuits and applications to logarithmic-sized transparent threshold signature schemes.
IACR Cryptol. ePrint Arch., 2020:1447, 2020.

Bibliography II

Ronald Cramer, Ivan Damgård, and Berry Schoenmakers.
Proofs of partial knowledge and simplified design of witness hiding protocols.
In *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.

Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu.
Short lattice-based one-out-of-many proofs and applications to ring signatures.
In *ACNS*, volume 11464 of *Lecture Notes in Computer Science*, pages 67–88. Springer, 2019.

Jens Groth and Markulf Kohlweiss.
One-out-of-many proofs: Or how to leak a secret and spend a coin.
In *EUROCRYPT (2)*, volume 9057 of *Lecture Notes in Computer Science*, pages 253–280. Springer, 2015.

Bibliography III


Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox.
Zcash Protocol Specication - Version 2020.1.7, 2020.