Generalized Special-Sound Interactive Proofs and their Knowledge Soundness

Thomas Attema and Serge Fehr and Nicolas Resch

TCC 2023 December 1, 2023 Taipei, Taiwan

Goal of an Interactive Proof (of Knowledge):

- Prove that a statement x admits a witness, or
- Prove knowledge of a witness w for a public statement x.

 $(x; w) \in R$ $\mathcal{P}(x; w)$ $\mathcal{V}(x)$ a_0 c_1 a_1 c_{μ} a_{μ} Accept/ Reject

Desirable Security Properties:

- Completeness: Honest provers always succeed in convincing a verifier.
- (Knowledge) Soundness: Dishonest provers (almost) never succeed.
- Zero-Knowledge: No information about the witness is revealed.

Knowledge soundness \iff existence of a *knowledge extractor*.

Knowledge extractor

- Input: Statement x and oracle access to a prover \mathcal{P}^* attacking the protocol.
- Goal: Compute a witness *w* for statement *x*.

Another Notion for IPs - Special-Soundness

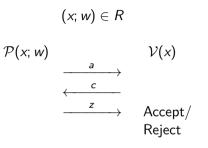
- Introduced in the context of Σ -protocols.
- Easier to prove special-soundness than knowledge soundness.

Another Notion for IPs - Special-Soundness

- Introduced in the context of Σ -protocols.
- Easier to prove special-soundness than knowledge soundness.

Definition

2-out-of-N special-soundness: Efficient algorithm to extract a witness w from 2 'colliding' protocol transcripts (a, c, z) and (a, c', z').



Another Notion for IPs - Special-Soundness

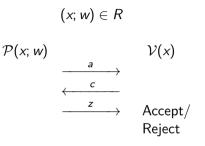
- Introduced in the context of Σ -protocols.
- Easier to prove special-soundness than knowledge soundness.

Definition

2-out-of-N special-soundness: Efficient algorithm to extract a witness w from 2 'colliding' protocol transcripts (a, c, z) and (a, c', z').

2-out-of-N special-soundness implies knowledge soundness with knowledge error 1/N.

• 1/N matches the trivial cheating probability.



• k-out-of-N special-soundness \implies knowledge error (k-1)/N.

• Requires k accepting transcripts with common first message a.

- k-out-of-N special-soundness \implies knowledge error (k-1)/N.
 - Requires k accepting transcripts with common first message a.
- (a) (k_1, \ldots, k_{μ}) -out-of- (N_1, \ldots, N_{μ}) special-sound multi-round interactive proofs:
 - Require a tree of transcripts to extract.

Non-Special-Sound Interactive Proofs - Amortization (1/2)

• Sometimes additional structure is required to extract from sets of accepting transcripts.

Non-Special-Sound Interactive Proofs - Amortization (1/2)

• Sometimes additional structure is required to extract from sets of accepting transcripts.

Proving Knowledge of *n* Pre-Images \mathbb{Z}_{q} -Module Homomorphism Ψ $\Psi(x_1) = P_1, \ldots, \Psi(x_n) = P_n$ $\mathcal{P}(x_1, P_1, \ldots, x_n, P_n)$ $\mathcal{V}(P_1,\ldots,P_n)$ $c_1,...,c_n$ $c_1,\ldots,c_n\leftarrow_R\mathbb{Z}_q$ $z = \sum_{i} c_i x_i$ $\Psi(z) \stackrel{?}{=} \sum_{i} c_i P_i$ z

Non-Special-Sound Interactive Proofs - Amortization (2/2)

$$\mathcal{P}(x_1, P_1, \dots, x_n, P_n) \qquad \qquad \mathcal{V}(P_1, \dots, P_n)$$

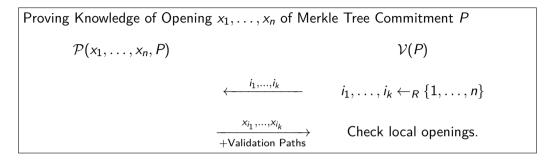
$$z = \sum_i c_i x_i \qquad \qquad \underbrace{\overset{c_1, \dots, c_n}{\longleftarrow} \qquad c_1, \dots, c_n \leftarrow_R \mathbb{Z}_q}_{\sum_i c_i P_i}$$

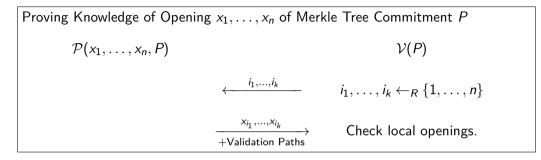
- Extraction requires:
 - Transcripts $(\mathbf{c}_1, z_1), \dots, (\mathbf{c}_n, z_n)$ s.t. $\mathbf{c}_1, \dots, \mathbf{c}_n$ is a basis of \mathbb{Z}_q^n .

Non-Special-Sound Interactive Proofs - Amortization (2/2)

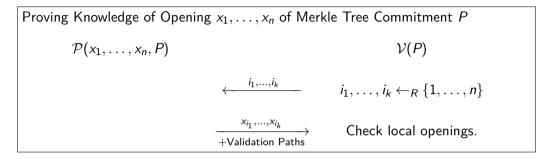
- Extraction requires:
 - Transcripts $(\mathbf{c}_1, z_1), \dots, (\mathbf{c}_n, z_n)$ s.t. $\mathbf{c}_1, \dots, \mathbf{c}_n$ is a basis of \mathbb{Z}_q^n .
- This IP is $(q^{n-1}+1)$ -special-sound;

- Extraction requires:
 - Transcripts $(\mathbf{c}_1, z_1), \dots, (\mathbf{c}_n, z_n)$ s.t. $\mathbf{c}_1, \dots, \mathbf{c}_n$ is a basis of \mathbb{Z}_q^n .
- This IP is $(q^{n-1}+1)$ -special-sound;
 - q is typically exponentially large \implies generic extractor is inefficient.

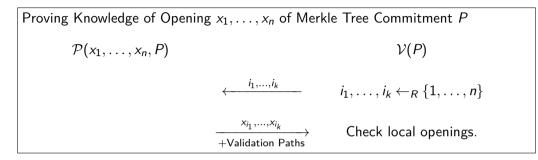




- Extraction requires:
 - Transcripts $(\mathbf{i}_1, \mathbf{x}_1), \dots, (\mathbf{i}_t, \mathbf{x}_t)$ s.t. $\mathbf{i}_1, \dots, \mathbf{i}_t$ cover $\{1, \dots, t\}$.



- Extraction requires:
 - Transcripts $(\mathbf{i}_1, \mathbf{x}_1), \dots, (\mathbf{i}_t, \mathbf{x}_t)$ s.t. $\mathbf{i}_1, \dots, \mathbf{i}_t$ cover $\{1, \dots, t\}$.
- This IP is $((n-1)^k + 1)$ -special-sound;



- Extraction requires:
 - Transcripts $(\mathbf{i}_1, \mathbf{x}_1), \dots, (\mathbf{i}_t, \mathbf{x}_t)$ s.t. $\mathbf{i}_1, \dots, \mathbf{i}_t$ cover $\{1, \dots, t\}$.
- This IP is $((n-1)^k + 1)$ -special-sound;

 \implies generic knowledge extractor is inefficient.

- A more general notion of special-soundness,
 - capturing the above examples;
- A novel knowledge extractor;
- A generalization to multi-round interactive proofs;
- Parallel repetition theorem;
- An application to the FRI-protocol (IOP).

$\Gamma \subseteq 2^{\mathcal{C}}$ is a monotone structure if

• $A \subseteq B \subseteq C$ and $A \in \Gamma$ implies $B \in \Gamma$ '.

 $\Gamma \subseteq 2^{\mathcal{C}}$ is a monotone structure if

• $A \subseteq B \subseteq C$ and $A \in \Gamma$ implies $B \in \Gamma'$.

A 3-round interactive proof with challenge set C is Γ -out-of-C special-sound, if there exists an efficient algorithm to extract a witness from accepting transcripts $(a, c_1, z_1), \ldots, (a, c_k, z_k)$ with $\{c_1, \ldots, c_k\} \in \Gamma$.

Examples of Γ-special-sound IPs

- *k*-special-sound IPs:
 - Challenge set C;
 - $\Gamma = \{S \subseteq \mathcal{C} : |S| \ge k\}.$

Examples of **F**-special-sound IPs

- *k*-special-sound IPs:
 - Challenge set C;
 - $\Gamma = \{S \subseteq \mathcal{C} : |S| \ge k\}.$
- Amortization:
 - Challenge set \mathbb{Z}_a^n ;
 - $\Gamma = \{S \subseteq \mathbb{Z}_q^n : \operatorname{span}(S) = \mathbb{Z}_q^n\}.$

Examples of **F**-special-sound IPs

- *k*-special-sound IPs:
 - Challenge set C;
 - $\Gamma = \{S \subseteq \mathcal{C} : |S| \ge k\}.$
- Amortization:
 - Challenge set Zⁿ_q;
 Γ = {S ⊆ Zⁿ_q : span(S) = Zⁿ_q}.
- Merkle tree IP:
 - Challenge set $\{A \subseteq \{1, \ldots, n\} : |A| \le k\};$
 - $\Gamma = \{S \subseteq \mathcal{C} : \cup_{A \in S} A = \{1, \ldots, n\}\}.$

Key Observation:

 \bullet At any stage the extractor can partition ${\cal C}$ into a set of "useful" and "useless" challenges.

Key Observation:

 \bullet At any stage the extractor can partition ${\cal C}$ into a set of "useful" and "useless" challenges.

Suppose the extractor has found accepting transcripts for challenges $A \subseteq C$ with $A \notin \Gamma$.

The function $U_{\Gamma}(A)$ defines the useful challenges.

Key Observation:

 \bullet At any stage the extractor can partition ${\cal C}$ into a set of "useful" and "useless" challenges.

Suppose the extractor has found accepting transcripts for challenges $A \subseteq C$ with $A \notin \Gamma$.

The function $U_{\Gamma}(A)$ defines the useful challenges.

- k-special-sound IPs:
 - $U_{\Gamma}(A) = \mathcal{C} \setminus A$.

Key Observation:

 \bullet At any stage the extractor can partition ${\cal C}$ into a set of "useful" and "useless" challenges.

Suppose the extractor has found accepting transcripts for challenges $A \subseteq C$ with $A \notin \Gamma$.

The function $U_{\Gamma}(A)$ defines the useful challenges.

Examples:

• *k*-special-sound IPs:

•
$$U_{\Gamma}(A) = \mathcal{C} \setminus A$$
.

Amortization:

•
$$\mathcal{C} = \mathbb{Z}_q^n$$
;
• $U_{\Gamma}(A) = \mathcal{C} \setminus \operatorname{span}(A)$.

Key Observation:

 \bullet At any stage the extractor can partition ${\cal C}$ into a set of "useful" and "useless" challenges.

Suppose the extractor has found accepting transcripts for challenges $A \subseteq C$ with $A \notin \Gamma$.

The function $U_{\Gamma}(A)$ defines the useful challenges.

Examples:

• *k*-special-sound IPs:

•
$$U_{\Gamma}(A) = \mathcal{C} \setminus A$$
.

- Amortization:
 - $C = \mathbb{Z}_q^n$; • $U_{\Gamma}(A) = C \setminus \operatorname{span}(A)$.
- Merkle tree IP:

•
$$C = \{S \subseteq \{1, \ldots, n\} : |S| \le k\};$$

• $U_{\Gamma}(A) = \{B \in C : B \nsubseteq \cup_{S \in A} S\}.$

 Formal Definition

 $U_{\Gamma}: 2^{\mathcal{C}} \rightarrow 2^{\mathcal{C}}, \quad S \mapsto \{c \in \mathcal{C} \setminus S: \exists A \in \Gamma \text{ s.t. } S \subset A \land A \setminus \{c\} \notin \Gamma\}$

Formal Definition

$$U_{\Gamma} \colon 2^{\mathcal{C}} \to 2^{\mathcal{C}} \,, \quad S \mapsto \{c \in \mathcal{C} \setminus S \colon \exists A \in \Gamma \text{ s.t. } S \subset A \land A \setminus \{c\} \notin \Gamma\}$$

It holds that:

•
$$U_{\Gamma}(A) \cap A = \emptyset$$
 for all A ;

Formal Definition

$$U_{\Gamma} \colon 2^{\mathcal{C}} \to 2^{\mathcal{C}} \,, \quad S \mapsto \{ c \in \mathcal{C} \setminus S \colon \exists A \in \Gamma \text{ s.t. } S \subset A \land A \setminus \{c\} \notin \Gamma \}$$

It holds that:

- $U_{\Gamma}(A) \cap A = \emptyset$ for all A;
- $U_{\Gamma}(B) \subseteq U_{\Gamma}(A)$ for all $A \subseteq B$;

Formal Definition

$$U_{\Gamma} \colon 2^{\mathcal{C}} \to 2^{\mathcal{C}} \,, \quad S \mapsto \{ c \in \mathcal{C} \setminus S \colon \exists A \in \Gamma \text{ s.t. } S \subset A \land A \setminus \{c\} \notin \Gamma \}$$

It holds that:

- $U_{\Gamma}(A) \cap A = \emptyset$ for all A;
- $U_{\Gamma}(B) \subseteq U_{\Gamma}(A)$ for all $A \subseteq B$;
- $U_{\Gamma}(A) = \emptyset$ for all $A \in \Gamma$.

Main idea.

• To find the ℓ -th transcript:

rewind and sample new challenge from $U_{\Gamma}(\{c_1, \ldots, c_{\ell-1}\})$.

Main idea.

• To find the ℓ -th transcript:

rewind and sample new challenge from $U_{\Gamma}(\{c_1, \ldots, c_{\ell-1}\})$.

More precisely, we adapt the extractor for k-special-sound IPs.

- Fails for the extractor from [ACK21];
- Works for the extractor introduced to handle parallel repetition [AF22].

Crucial parameter in the analysis:

$$t_{\Gamma} := \max iggl\{ k \in \mathbb{N}_0 : egin{array}{c} \exists c_1, \dots, c_k \in \mathcal{C} ext{ s.t.} \ c_i \in U_{\Gamma}igl(\{c_1, \dots, c_{i-1}\}igr) \ orall i igr\} \end{cases}$$

Crucial parameter in the analysis:

$$t_{\Gamma} := \max \left\{ k \in \mathbb{N}_0 : \begin{array}{c} \exists c_1, \dots, c_k \in \mathcal{C} \text{ s.t.} \\ c_i \in U_{\Gamma} (\{c_1, \dots, c_{i-1}\}) \ \forall i \end{array} \right\}$$

Expected running time grows linearly in t_{Γ} :

• \implies knowledge soundness if t_{Γ} is polynomial.

Examples

- k_{Γ} : Threshold special-soundness parameters.
- t_{Γ} : Refined special-soundness parameters.

	Challenge Set	kΓ	tΓ
<i>k</i> -special-sound П	${\mathcal C}$	k	k
Amortization	\mathbb{Z}_q^n	$q^{n-1}+1$	n
Merkle Tree Opening	$\{1,\ldots,n\}^k$	$(n-1)^k+1$	n-k+1
<i>t</i> -fold Parallel of Π^1	\mathcal{C}^t	$(k-1)^t+1$	$t \cdot (k-1) + 1$

Another example: Local Special-Soundness

¹Correction of the paper. Parallel repetition *is* appropriately captured by this generalization.

Summary

This presentation:

- Introduced a more general notion of special-soundness,
 - together with the tools to analyze this property;
- Claimed that it implies knowledge soundness.

In the paper:

- The knowledge extractor;
- A generalization to multi-round interactive proofs;
- Parallel repetition theorem;
- An application to the FRI-protocol (IOP).
 - With a discussion on the limitations of this approach.

Next:

- Analyzing the Fiat-Shamir transformation of $(\Gamma_1, \ldots, \Gamma_\mu)$ -special-sound IPs.
 - Follow-up work to appear soon.
- Improve our FRI-extractor.
 - Open question.

Thanks!

Thomas Attema, Ronald Cramer, and Lisa Kohl. A compressed Σ-protocol theory for lattices. In CRYPTO (2), volume 12826 of Lecture Notes in Computer Science, pages 549–579. Springer, 2021.

Thomas Attema and Serge Fehr.

Parallel repetition of (k_1, \ldots, k_{μ}) -special-sound multi-round interactive proofs. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO*, volume 13507 of *Lecture*

Notes in Computer Science, pages 415-443. Springer, 2022.

Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity.

In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic, volume 107 of LIPIcs, pages 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.