Hardness Amplification for Weakly Verifiable Cryptographic Primitives

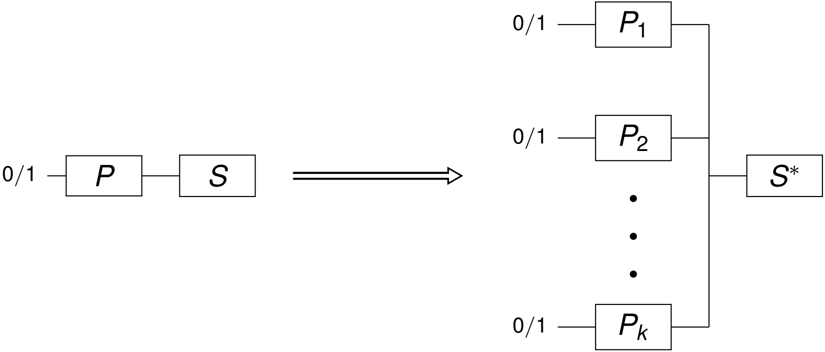
Grzegorz Mąkosa

Advisors: Prof. Dr. Thomas Holenstein, Dr. Robin Künzler

Department of Computer Science, ETH Zürich

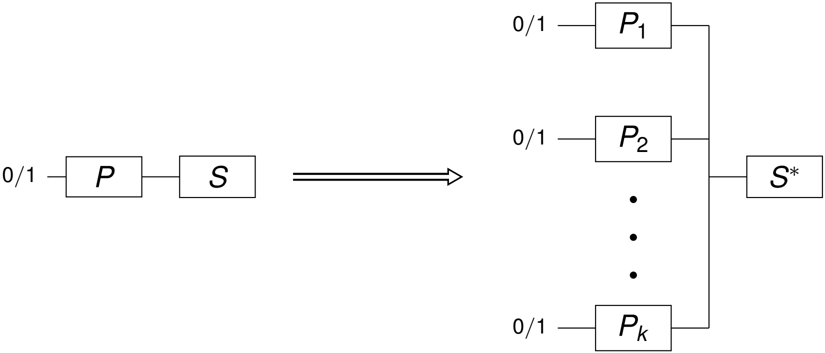
Hardness Amplification

- Is solving parallel repetition of problems substantially harder than a single instance?



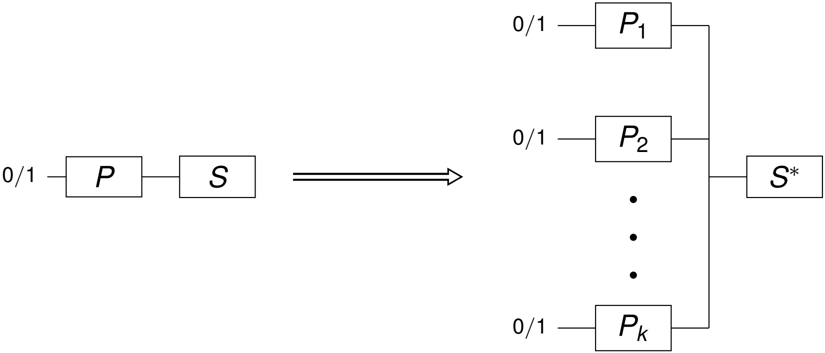
Hardness Amplification

- Weak one-way function $\Longrightarrow$ strong one-way function



Hardness Amplification

- Weak one-way function $\Longrightarrow$ strong one-way function
- What about MAC, signature schemes, CAPTCHAs?



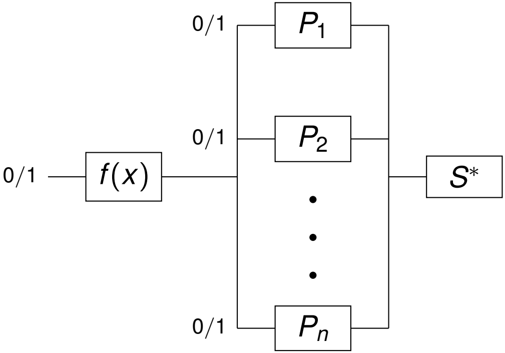
Agenda

- Motivation
- Background
    - Weakly Verifiable Puzzles
    - Threshold and Monotone Functions
    - Dynamic Puzzles
    - Interactive Puzzles
- Previous Works
- My Results
- Discussion and Questions

Threshold and Monotone Functions

Threshold function

$$f_K(b_1, \dots, b_n) = \begin{cases} 1 \text{ if } \sum_{i=1}^{n} b_i \geq K \\ 0 \text{ otherwise.} \end{cases}$$



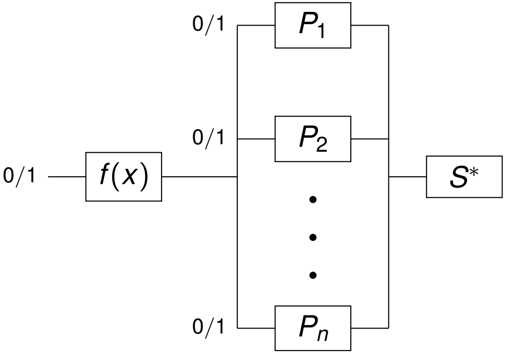
Threshold and Monotone Functions

Threshold function

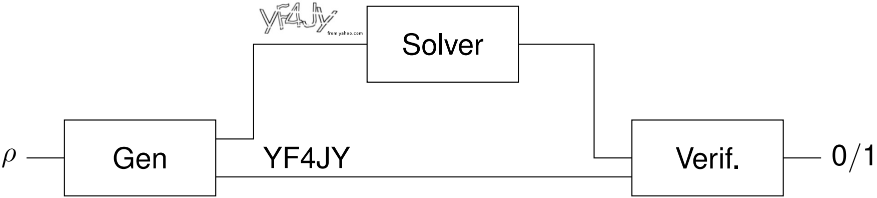
$$f_K(b_1,\ldots,b_n) = \begin{cases} 1 \text{ if } \sum_{i=1}^{n} b_i \geq K \\ 0 \text{ otherwise.} \end{cases}$$

Monotone function

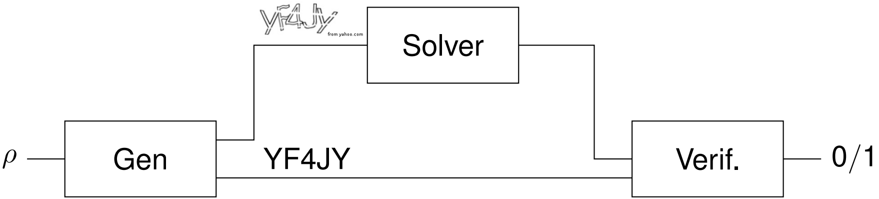
$$f(b_0,\ldots,b_n) : \{0,1\}^n \to \{0,1\}$$



Weakly Verifiable Puzzles - CAPTCHA

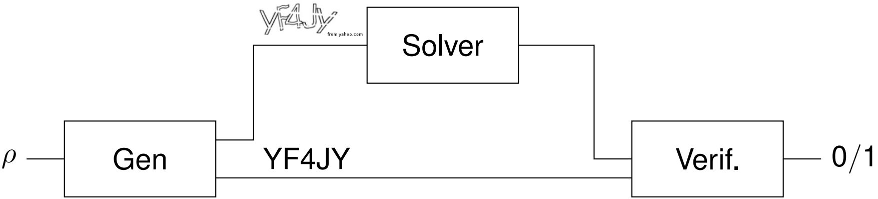


Weakly Verifiable Puzzles - CAPTCHA



- Small solutions space.

Weakly Verifiable Puzzles - CAPTCHA



ρ

Gen

YF4JY

Solver

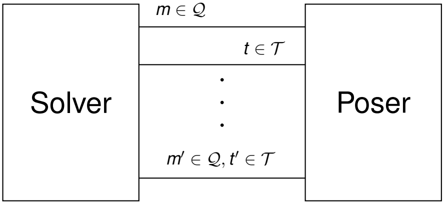
Verif.

$0/1$

- Small solutions space.
- Solver cannot efficiently verify correctness of solutions.

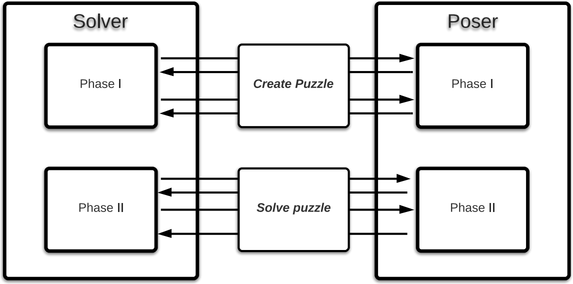
Dynamic Puzzles Example

- Game-based security definition of MAC.



- Set of messages $\mathcal{Q}$
- Hint - solution for $q \in \mathcal{Q}$
- Set of hint indices $\mathcal{H} \subseteq \mathcal{Q}$
- Verification query solution for $q \in \mathcal{Q} \setminus \mathcal{H}$.
- Number of hint and verification queries limited.

Interactive puzzle - commitment protocols



Hardness amplification results

- Weakly verifiable puzzles e.g. CAPTCHA, [CHS05]

Hardness amplification results

- Weakly verifiable puzzles e.g. CAPTCHA, [CHS05]
- Dynamic weakly verifiable puzzles + threshold functions e.g. MAC,[DIJK09]

Hardness amplification results

- Weakly verifiable puzzles e.g. CAPTCHA, [CHS05]
- Dynamic weakly verifiable puzzles $+$ threshold functions e.g. MAC,[DIJK09]
- Interactive weakly verifiable puzzles $+$ monotone function e.g. commitment protocols, [HS11]

Goal

- Define a type of puzzles that generalize MAC, CAPTCHA, bit commitments.
- Amplify hardness by parallel repetition.

Monotone functions + Dynamic weakly verifiable puzzles + Interactive weakly verifiable puzzles

Reduction

- A - solving a single puzzle is hard
- B - solving parallel repetition is hard

$$A \implies B$$

$$\neg B \implies \neg A$$

- Given a good solver C for parallel repetition
- Reduce C to a solver for single puzzle

Problem: conflicting hint queries

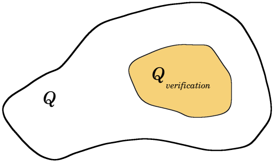
- The solver *C* can be run multiple times.

Problem: conflicting hint queries

- The solver *C* can be run multiple times.
- Hint queries can prevent verification queries from succeeding.

Problem: conflicting hint queries

- The solver *C* can be run multiple times.
- Hint queries can prevent verification queries from succeeding.
- Use hash function to partition query domain [DIJK09].



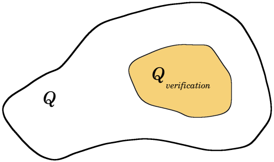
$$hash \leftarrow \mathcal{H}$$

$$hash : \mathcal{Q} \rightarrow \{0, 1, \ldots, 2(h + v) - 1\}$$

$$\mathcal{Q}_{verification} := q \in \mathcal{Q} : hash(q) = 0$$

Problem: conflicting hint queries

- The solver C can be run multiple times.
- Hint queries can prevent verification queries from succeeding.
- Use hash function to partition query domain [DIJK09].
- Can ask hints only on $\mathcal{Q} \setminus \mathcal{Q}_{verification}$.



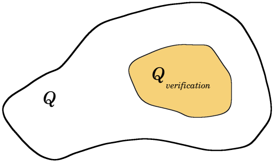
$$hash \leftarrow \mathcal{H}$$

$$hash : \mathcal{Q} \rightarrow \{0, 1, \ldots, 2(h + v) - 1\}$$

$$\mathcal{Q}_{verification} := q \in \mathcal{Q} : hash(q) = 0$$

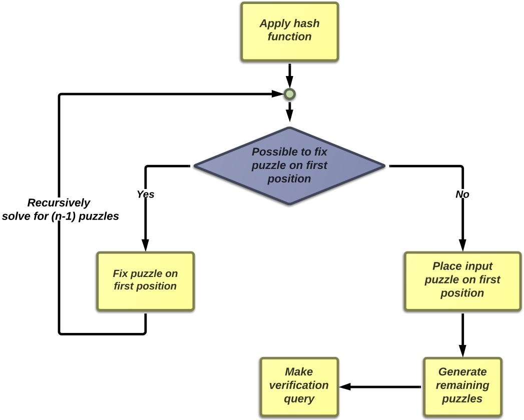
Problem: conflicting hint queries

- The solver *C* can be run multiple times.
- Hint queries can prevent verification queries from succeeding.
- Use hash function to partition query domain [DIJK09].
- Can ask hints only on $\mathcal{Q} \setminus \mathcal{Q}_{verification}$.
- Substantial success probability for partitioned domain.



$$hash \leftarrow \mathcal{H}$$
$$hash : \mathcal{Q} \rightarrow \{0, 1, \ldots, 2(h + v) - 1\}$$
$$\mathcal{Q}_{verification} := q \in \mathcal{Q} : hash(q) = 0$$

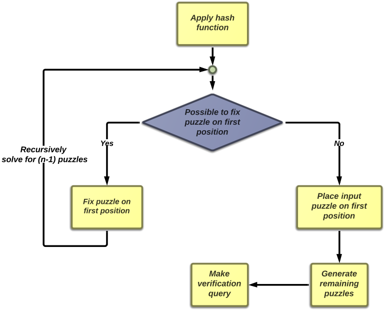
Approach overview


Apply hash function
Possible to fix puzzle on first position
Yes
No
Recursively solve for (n-1) puzzles
Fix puzzle on first position
Place input puzzle on first position
Generate remaining puzzles
Make verification query

Problem: verifying the solution

- Cannot check whether the solution is correct.
- For a special case where all puzzles have to be solved.
- Look at the remaining $n - 1$ puzzles that are generated.

$$\underbrace{\boxed{?}\;\overbrace{\boxed{1}\,\boxed{1}\;\cdot\;\cdot\;\cdot\;\cdot\;\cdot\;\cdot\;\boxed{1}}^{< \delta^{k-1}}}_{\geq \delta^{k}}$$



Apply hash function

Possible to fix puzzle on first position

Yes

No

Recursively solve for (n-1) puzzles

Fix puzzle on first position

Place input puzzle on first position

Generate remaining puzzles

Make verification query

Result

Given a solver for parallel repetition of puzzles that satisfies

$$\geq \delta^k + \varepsilon \qquad\qquad \geq \mathrm{Pr}[g(u_1, \ldots, u_k) = 1] + \varepsilon,$$

where $\mathrm{Pr}[u_i = 1] = \delta$.

We devise a solver for a single puzzle that satisfies (almost surely)

$$\geq \frac{1}{16(h+v)}\left(\delta + \frac{\varepsilon}{6k}\right).$$

Discussion

- Not clear whether it is possible to improve the result

$$\geq \frac{1}{16(h+v)}\left(\delta + \frac{\varepsilon}{6k}\right).$$

- Tried to improve it. ✗
- Tried to show it is optimal. ✗

Questions

Bibliography

Ran Canetti, Shai Halevi, and Michael Steiner.
Hardness amplification of weakly verifiable puzzles.
In *Theory of Cryptography*, pages 17–33. Springer, 2005.

Yevgeniy Dodis, Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets.
Security amplification for interactive cryptographic primitives.
In *Theory of cryptography*, pages 128–145. Springer, 2009.

Thomas Holenstein and Grant Schoenebeck.
General hardness amplification of predicates and puzzles.
In *Theory of Cryptography*, pages 19–36. Springer, 2011.