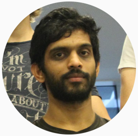
On Average Case Hardness in TFNP from One-Way Functions

TCC 2020

Pavel Hubáček   Chethan Kamath   Karel Král   **Veronika Slívová**

Computer Science Institute of Charles University

Lot of effort for proving average-case hardness in TFNP under various cryptographic assumptions [Pap94, Jeř16, BPR15, GPS16, HY17, KS17, CHK+19a, CHK+19b, EFKP20, BG20]

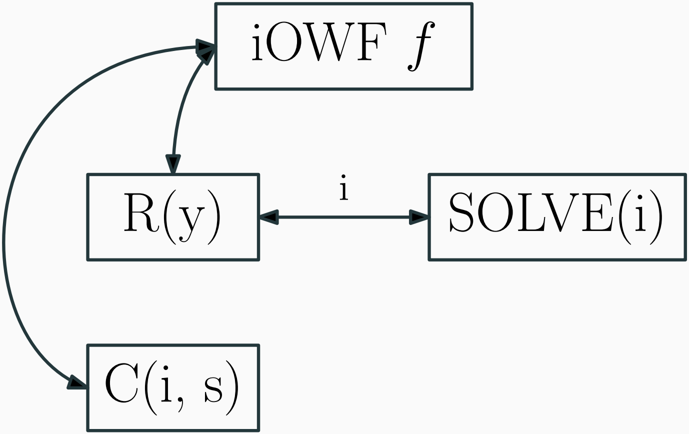
Can hardness be based on an unstructured assumption of (injective) OWF?

Previous work

Hard-on-average distributions in TFNP	
[BPR15, GPS16]	OWF + iO
[HNY17]	OWF + derandomization-style assumption
[KS17]	iOWF + private-key FE

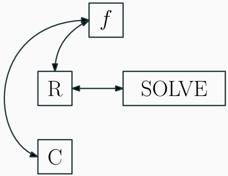
Impossibility results	
[RSS17]	many solutions from OWFs, CRHF, ...
this work	no **simple** construction from iOWFs

Fully black-box construction of hard TFNP problem from iOWF



Fully black-box construction of hard TFNP problem from iOWF

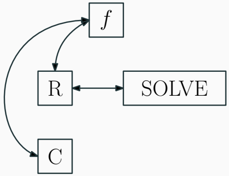
- R, C are poly-time algorithms
    - C is TFNP verifier
    - R is security reduction



Fully black-box construction of hard TFNP problem from iOWF

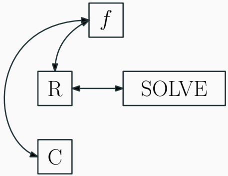
- R, C are poly-time algorithms
  - C is TFNP verifier
  - R is security reduction
- Correctness: C is always total.

  $\forall f\ \forall i\ \exists s\colon C^f(i,s) = 1$



Fully black-box construction of hard TFNP problem from iOWF

- R, C are poly-time algorithms
  - C is TFNP verifier
  - R is security reduction
- Correctness: C is always total.
  $\forall f\ \forall i\ \exists s \colon C^f(i,s) = 1$
- Security: If Solve always solves then R inverts with nonnegligible probability.



$\exists p$ polynomial s.t. $\forall f\ \forall \text{Solve}$

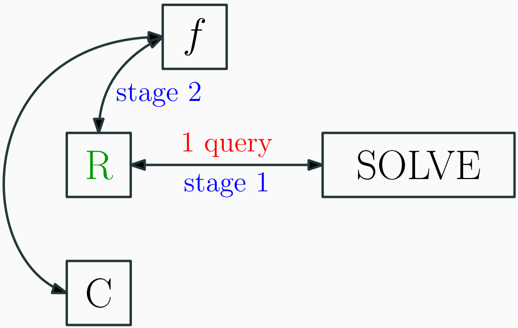
if

$$\forall i \colon \text{Solve}^f(i) = s \text{ s.t. } C^f(i,s) = 1$$

then for infinitely many $n \in \mathbb{N}$,

$$\Pr_{x \leftarrow \{0,1\}^n}[f(R^{f,\text{Solve}}(1^n, f(x))) = f(x)] \geq \frac{1}{p(n)}$$

Simple fully black-box construction of hard TFNP problem from iOWF

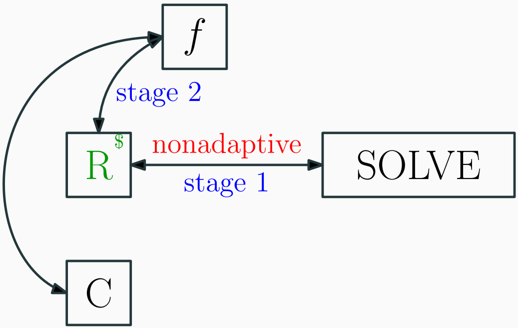


many-one: At most 1 query to Solve

deterministic: Algorithm R is deterministic

f-oblivious: Queries R makes to Solve are independent of f

Simple fully black-box construction of hard TFNP problem from iOWF



nonadaptive: Queries to Solve are nonadaptive

randomized: Algorithm R is randomized

f-oblivious: Queries R makes to Solve are independent of f

Our results

Main theorem
There is no randomized fully black-box non-adaptive f-oblivious construction of average-case hard TFNP problem from iOWF.

Our results

Main theorem
There is no randomized fully black-box non-adaptive f-oblivious construction of average-case hard TFNP problem from iOWF.

Special case of our Main theorem
There is no deterministic fully black-box many-one f-oblivious construction of average-case hard TFNP problem from iOWF.

Black-box separation - proof technique

The two oracle technique by [HR04] (goes back to [Sim98]):

Define an oracle $\mathcal{O}$ such that

1. iOWF exists with respect to $\mathcal{O}$
2. TFNP is easy with respect to $\mathcal{O}$

OWP vs. iOWF

OWP

- Any OWP $\pi\colon \{0,1\}^n \to \{0,1\}^n$ gives rise to a hard-on-average TFNP problem.

iOWF

- Simple reductions are impossible.

OWP vs. iOWF

OWP

- Any OWP $\pi\colon \{0,1\}^n \to \{0,1\}^n$ gives rise to a hard-on-average TFNP problem.
- For any $y \in \{0,1\}^n$, the preimage $\pi^{-1}(y)$ exists.

iOWF

- Simple reductions are impossible.
- For any iOWF $f \in \{0,1\}^n \to \{0,1\}^{n+1}$, only $y \in \mathrm{Im}(f)$ have a preimage under f.

How would a construction look like?

Computation of $R^f(y)$: $\ldots$ query $\mathsf{Solve}(i_y)$ $\ldots$

Correctness: $\forall f\ \exists s\colon C^f(i_y, s) = 1$

How would a construction look like?

Computation of $R^f(y)$: $\ldots$ query $\mathsf{Solve}(i_y)$ $\ldots$

Correctness: $\forall f\ \exists s\colon C^f(i_y, s) = 1$

Even for g such that $y \notin \mathrm{Im}(g)$, some solution s must exists!

How would a construction look like?

Even for g such that $y \notin \mathrm{Im}(g)$, some solution s must exists!

0^n	$g(0^n)$
$0^{n-1}1$	$g(0^{n-1}1)$
a	
b	
1^n	$g(1^n)$

0^n	$f(0^n)$
$0^{n-1}1$	$f(0^{n-1}1)$
a	
	y
b	
1^n	$f(1^n)$

How would a construction look like?

Even for g such that $y \notin \mathrm{Im}(g)$, some solution s must exists!

0^n	$g(0^n)$
$0^{n-1}1$	$g(0^{n-1}1)$
a	
b	
1^n	$g(1^n)$

0^n	$f(0^n)$
$0^{n-1}1$	$f(0^{n-1}1)$
a	
	y
b	
1^n	$f(1^n)$

$C^g(i,s)$, $C^f(i,s)$ query only a, b,

How would a construction look like?

Even for g such that $y \notin \mathrm{Im}(g)$, some solution s must exists!

0^n	$g(0^n)$
$0^{n-1}1$	$g(0^{n-1}1)$
a	
b	
1^n	$g(1^n)$

0^n	$f(0^n)$
$0^{n-1}1$	$f(0^{n-1}1)$
a	
	y
b	
1^n	$f(1^n)$

$C^g(i,s)$, $C^f(i,s)$ query only a, b, thus $C^g(i,s) = C^f(i,s) = 1$.

Solution s is useless for inverting challenge y.

How to identify a useless solution?

Solve does not know the challenge y.

How to identify a useless solution?

Solve does not know the challenge y.

Security
The reduction is successful in inverting given access to any algorithm Solve solving the TFNP problem.

How to identify a useless solution?

Solve does not know the challenge y.

Security
The reduction is successful in inverting given access to any algorithm Solve solving the TFNP problem.

Try to identify challenge y from the instance i by simulating the reduction R on all possible challenges.

Solve

Solve$^{f}_{R,C}(i)$**:**

1. Compute set of protected $Y = \left\{ y \mid R^f(y) \text{ queries } i \right\}$
2. Compute set of solutions $S = \left\{ s \mid C^f(i, s) = 1 \right\}$

    3.1 If $\exists s \in S$ s.t. preimage of any $y \in Y$ is not queried, return s

Solve

Solve$^f_{R,C}(i)$:

1. Compute set of protected $Y = \left\{y \mid R^f(y) \text{ queries } i\right\}$
2. Compute set of solutions $S = \left\{s \mid C^f(i,s) = 1\right\}$
3. while True
    3.1 If $\exists s \in S$ s.t. preimage of any $y \in Y$ is not queried, return s
    3.2 Carefully remove some y's from Y.

Solve

Solve$^f_{R,C}(i)$:

1. Compute set of protected $Y = \left\{y \mid R^f(y) \text{ queries } i\right\}$
2. Compute set of solutions $S = \left\{s \mid C^f(i, s) = 1\right\}$
3. while True
   3.1 If $\exists s \in S$ s.t. preimage of any $y \in Y$ is not queried, return s
   3.2 Carefully remove some y's from Y.

Given access to (f, Solve):

1. The TFNP problem is easy – Solve always returns a correct solution
2. Reduction R does not invert f – incompressibility argument

Conclusions

If it is possible to construct a hard TFNP problem from iOWF, then the reduction must be quite involved.

Conclusions

If it is possible to construct a hard TFNP problem from iOWF, then the reduction must be quite involved.

Can we get the same impossibility result

- even without the f-obliviousness requirement or
- even when we allow nonadaptive queries to Solve?

Conclusions

If it is possible to construct a hard TFNP problem from iOWF, then the reduction must be quite involved.

Can we get the same impossibility result

- even without the f-obliviousness requirement or
- even when we allow nonadaptive queries to Solve?

Thank you for your attention.

ia.cr/2020/1162

Nir Bitansky and Idan Gerichter.
On the cryptographic hardness of local search.
In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, pages 6:1–6:29, 2020.

Nir Bitansky, Omer Paneth, and Alon Rosen.
On the cryptographic hardness of finding a Nash equilibrium.
In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1480–1498, 2015.

Arka Rai Choudhuri, Pavel Hubáček, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N. Rothblum.
Finding a Nash equilibrium is no easier than breaking Fiat-Shamir.
In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1103–1114. ACM, 2019.

Arka Rai Choudhuri, Pavel Hubáček, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N. Rothblum.
PPAD-hardness via iterated squaring modulo a composite.
IACR Cryptology ePrint Archive, 2019:667, 2019.

Naomi Ephraim, Cody Freitag, Ilan Komargodski, and Rafael Pass.
Continuous verifiable delay functions.
In *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 125–154, 2020.

Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan.
Revisiting the cryptographic hardness of finding a Nash equilibrium.
In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 579–604, 2016.

Pavel Hubáček, Moni Naor, and Eylon Yogev.
The journey from NP to TFNP hardness.
In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 60:1–60:21, 2017.

Chun-Yuan Hsiao and Leonid Reyzin.
Finding collisions on a public road, or do secure hash functions need secret coins?
In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 92–105. Springer, 2004.

Pavel Hubáček and Eylon Yogev.
Hardness of continuous local search: Query complexity and cryptographic lower bounds.
In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1352–1371, 2017.

Emil Jeřábek.
Integer factoring and modular square roots.
J. Comput. Syst. Sci., 82(2):380–394, 2016.

Ilan Komargodski and Gil Segev.
From Minicrypt to Obfustopia via private-key functional encryption.
In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, pages 122–151, 2017.

Christos H. Papadimitriou.
On the complexity of the parity argument and other inefficient proofs of existence.
J. Comput. Syst. Sci., 48(3):498–532, 1994.

Alon Rosen, Gil Segev, and Ido Shahaf.
Can PPAD hardness be based on standard cryptographic assumptions?
In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 747–776. Springer, 2017.

Daniel R. Simon.
Finding collisions on a one-way street: Can secure hash functions be based on general assumptions?
In *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer, 1998.