Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Leakage resilient one-way functions: The auxiliary-input setting ☆

Ilan Komargodski [1]

Cornell Tech, New York, NY 10044, USA

ARTICLE INFO

Article history:
Received 1 February 2018
Received in revised form 27 May 2018
Accepted 5 June 2018
Available online xxxx
Communicated by G. Persiano

Keywords:
One-way functions
Computational leakage resilience
Auxiliary input

ABSTRACT

Most cryptographic schemes are designed in a model where perfect secrecy of the secret key is assumed. In most physical implementations, however, some form of information leakage is inherent and unavoidable. To deal with this, a flurry of works showed how to construct basic cryptographic primitives that are resilient to various forms of leakage.

Dodis et al. (FOCS '10) formalized and constructed leakage resilient one-way functions. These are one-way functions f such that given a random image $f(x)$ and leakage $g(x)$ it is still hard to invert $f(x)$. Based on any one-way function, Dodis et al. constructed such a one-way function that is leakage resilient assuming that an attacker can leak any lossy function g of the input.

In this work we consider the problem of constructing leakage resilient one-way functions that are secure with respect to *arbitrary computationally hiding* leakage (a.k.a. auxiliary-input). We consider both types of leakage – selective and adaptive – and prove various possibility and impossibility results.

On the negative side, we show that if the leakage is an adaptively-chosen arbitrary one-way function, then it is *impossible* to construct leakage resilient one-way functions. The latter is proved both in the random oracle model (without any further assumptions) and in the standard model based on a strong vector-variant of DDH. On the positive side, we observe that when the leakage is chosen ahead of time, there are leakage resilient one-way functions based on a variety of assumption.

1. Introduction

The holy grail of cryptography is designing systems that remain secure in the presence of adversarial behavior. For this, one has to specify (1) a cryptographic primitive of interest (e.g. an encryption scheme or a signature scheme), and (2) a model that captures the power of a potential adversary and what it means for it to break the system.

One of the most common assumptions is that secret keys are perfectly secret and are completely unknown to an adversary. However, in many physical implementations some information does leak due to various side-channel attacks, reuse of randomness, and more.

☆ A preliminary version of this paper appeared in the 14th International Conference on Theory of Cryptography (TCC 2016-B) [29].

E-mail address: komargodski@cornell.edu.

[1] Supported in part by a Packard Foundation Fellowship and by an AFOSR grant FA9550-15-1-0262. Work done at the Weizmann Institute of Science, supported in part by a Levzion fellowship, by grants from the Israel Science Foundation (grant no. 1255/12), BSF and from the I-CORE Program of the Planning and Budgeting Committee and the Israel Science Foundation (grant no. 4/11).

https://doi.org/10.1016/j.tcs.2018.06.014
0304-3975/© 2018 Elsevier B.V. All rights reserved.

This deficiency raised the necessity to build a theory of security against classes of side-channel attacks. Starting with the works of [14,27,31], a flurry of works in which different classes of side channel attacks have been defined and different cryptographic primitives have been designed to provably withstand these attacks (see, for example, [14,27,22,31,20,34,18,3,1,16,23,21,5,7,15,6,32,11,24]).

We consider the problem of constructing the most basic cryptographic primitive, a one-way function, in a setting where an adversary obtains side-channel information (this notion was first formalized by [2,17]). A one-way function f is an efficiently computable function such that given $f(x)$ for a random input x, any efficient adversary cannot find an x' such that $f(x) = f(x')$. A leakage resilient one-way function f is a one-way function such that given $f(x)$ as above and $g(x)$, where g is adversarially chosen, it is still hard to invert f and recover such an x'.

To obtain some sort of security, one clearly has to restrict the adversary to choose g from some collection of functions that do not trivially reveal x by themselves. Indeed, if g is the identity function, no leakage resilient one-way function f exists. Thus, several assumptions on the power of the adversary have been considered. Already in the work of Canetti et al. [14], the authors showed how to obtain a leakage-resilient one-way function assuming that the attacker can leak an arbitrary but sufficiently small subset of the bits of the input. However, this may be overly restrictive as it provides no guarantees if the attacker can learn the XOR of all the input bits. This issue was addressed in several works (see, for example, [2,17,15]) showing that there exists a leakage-resilient one-way function assuming that the attacker can leak any lossy function of the input, namely, any function whose image size is significantly smaller than the domain size. The leakage-resilience in both settings is proven based on the existence of any one-way function which is the weakest assumption possible. For completeness, we provide a proof of the following theorem in Appendix A.

Theorem 1.1 *([2,17], Informal). Assuming that one-way functions exist, there exists a one-way function f, such that for any adversarially-chosen lossy function g, given $f(x)$ and $g(x)$ for a random x, it is computationally hard to invert f.*

Motivated by the positive results for a wide class of leakage functions, we study the question of designing leakage-resilient one-way functions that are secure with respect to *arbitrary computationally hiding* leakage functions. We model this by allowing the leakage to be an arbitrary one-way function, even such that fully determine the input.[2] We consider both an *adaptive* notion of security in which the leakage function is adversarially chosen (from a restricted pre-defined collection) after f is fixed, and a selective notion in which the leakage is chosen ahead of time, before f is.

1.1. Our contributions

Adaptively-chosen leakage. We show that if the leakage can be an arbitrary one-way function, then there cannot be a leakage resilient one-way function f. More precisely, we show that for every one-way function f, there exists a one-way function g (that depends on f) such that when one gets both $f(x)$ and $g(x)$, it is easy to invert f.

We prove this result in two ways: in the random oracle model and in the standard model based on a strong vector-variant of DDH. Specifically, we first show that if the leakage function has access to a random oracle O, then we can construct an oracle-aided function g^{O} which is one-way and $g^{\mathsf{O}}(x)$ together with $f(x)$ allow to recover x. For the result in the standard model, we rely on multi-bit point obfuscators that exist based on a strong vector-variant of the DDH assumption [13,4]; see Section 2.3 and Assumption 2.10.

Theorem 1.2 *(Informal). Let O be a random oracle. For every one-way function f, there is a one-way function g^{O} such that for every x given $f(x)$ and $g(x)$ it is easy to recover x.*

Theorem 1.3 *(Informal). Assuming multi-bit point obfuscators, for every one-way function f, there is a one-way function g such that for every x given $f(x)$ and $g(x)$ it is easy to recover x.*

Moreover, such multi-bit point obfuscators can be constructed from a strong vector-variant of the DDH assumption.

Selectively-chosen leakage. We show that if the leakage function g is fixed ahead of time, then there exists a leakage resilient one-way function f for g from various assumptions. To this end, we observe that one-wayness with respect to selectively-chosen leakage is tightly related to extracting polynomially-many hard-core bits.

Theorem 1.4 *(Informal). For every leakage one-way function g, a hardcore function for g that outputs polynomially-many hard-core bits is a leakage-resilient one-way function for g.*

If g is a sub-exponentially hard one-way function, then extracting polynomially-many hard-core bits is possible due to Goldreich and Levin [26] (and any pseudorandom generator). Bellare, Stepanovs, and Tessaro [10] (see also the follow-up work of Brzuska and Mittelbach [9]) were the first to show how to extract *any* polynomial number of hard-core bits from

[2] This setting is sometimes referred to as the *auxiliary-input* setting (see, for example, [25,18,16]).