# Immunization against complete subversion without random oracles

**DOI:**
10.1016/j.tcs.2021.01.002

**Citation for published version (APA):**
Ateniese, G., Francati, D., Magri, B., & Venturi, D. (2021). Immunization against complete subversion without random oracles. *Theoretical Computer Science*, *859*, 1-36. https://doi.org/10.1016/j.tcs.2021.01.002

**Published in:**
Theoretical Computer Science

# Immunization against Complete Subversion without Random Oracles*

Giuseppe Ateniese[1], Danilo Francati[1], Bernardo Magri[2], and Daniele Venturi[3]

[1]*Stevens Institute of Technology, New Jersey, USA*
[2]*Department of Computer Science, Aarhus University, Denmark*
[3]*Department of Computer Science, Sapienza University of Rome, Italy*

January 3, 2021

**Abstract**

We seek constructions of general-purpose immunizers that take arbitrary cryptographic primitives, and transform them into ones that withstand a powerful "malicious but proud" adversary, who attempts to break security by possibly subverting the implementation of *all* algorithms (including the immunizer itself!), while trying not to be detected. This question is motivated by the recent evidence of cryptographic schemes being intentionally weakened, or designed together with hidden backdoors, *e.g.*, with the scope of mass surveillance.

Our main result is a subversion-secure immunizer in the plain model, that works for a fairly *large class* of *deterministic* primitives, *i.e.* cryptoschemes where a secret (but *tamperable*) random source is used to generate the keys and the public parameters, whereas all other algorithms are deterministic. The immunizer relies on an additional *independent* source of *public* randomness, which is used to sample a public seed.

Assuming the public source is untamperable, and that the subversion of the algorithms is chosen independently of the seed, we can instantiate our immunizer from any one-way function. In case the subversion is allowed to depend on the seed, and the public source is still untamperable, we obtain an instantiation from collision-resistant hash functions. In the more challenging scenario where the public source is also tamperable, we additionally need to assume that the initial cryptographic primitive has sub-exponential security.

Previous work in the area only obtained subversion-secure immunization for very restricted classes of primitives, often in weaker models of subversion and using random oracles.

**Keywords:** Complete subversion; Cliptography; Standard model.

*An abridged version of this work appeared in the proceedings of the 17th International Conference on Applied Cryptography and Network Security (ACNS 2019). This is the full version.

## Contents

# 1 Introduction

A common trend in modern cryptography is to design cryptographic schemes that come with a proof of security in a well-defined model; the proof is typically by reduction, meaning that violating the security of the scheme implies the existence of an efficient algorithm for solving some well-studied mathematical problem which is believed to be hard (*e.g.*, factoring certain integers, or inverting a one-way function). While having such a security proof is definitely a desirable feature, it is at least as important to make sure that the security model fits reality, as otherwise provably secure schemes are of little use in practice.

Unfortunately, security models often make idealized assumptions that are not always fulfilled in the real world. In this paper, we focus on one of those gaps, which is the discrepancy between the *specification* of a cryptographic scheme and its *implementation*. In particular, we consider the extreme case where the implementation is fully adversarial, *i.e.* the adversary is allowed to subvert or substitute some (or possibly all) algorithms in the original specification, with the purpose of weakening security.

The above scenario recently gained momentum due to the NSA leaks by Edward Snowden [27, 5, 23], and because of the EC_DUAL PRG[1] incident [11]. These hazards challenge modern cryptographers to design protection mechanisms withstanding subversion and tampering, as it was also highlighted by Phil Rogaway in his 2015 IACR Distinguished Lecture [28].

## 1.1 Background

Clearly, in order to guarantee some form of security in such an adversarial setting, we must put some restrictions on the adversary, as otherwise it is easy to subvert a cryptographic scheme in a way that becomes insecure (*e.g.*, the subverted scheme could always output the secret key). A natural restriction, which is also inspired by real-world attacks, is to demand that a subversion should be *undetectable* by honest users. In other words, the adversary's goal is to tamper with the specification of a cryptographic scheme in such a way that the produced outputs look indistinguishable from that of a faithful implementation, yet they allow to completely break security given some additional piece of information.

As it turns out, the possibility of such attacks was already uncovered more than twenty years ago by Young and Yung [35, 36], who dubbed the field kleptography (a.k.a. "cryptography against cryptography"). At Crypto 2014, Bellare, Paterson, and Rogaway [9] revisited this setting for the concrete case of symmetric encryption. In particular, on the one hand, they showed that it is possible to hide a backdoor in the encryption algorithm of any sufficiently randomized symmetric encryption scheme in such a way that the produced ciphertexts appear indistinguishable from honestly computed ones, yet knowledge of the backdoor allows to extract the secret key in full; on the other hand, they suggested that deterministic symmetric encryption schemes are secure against all subversion attacks that meet some form of undetectability. Their results were later extended in several ways [14, 8], while follow-up work studied similar questions for the case of digital signatures [3, 12], pseudorandom generators [16, 15], non-interactive zero knowledge [7], key encapsulation [4], hash functions [20, 31], and even hardware trojans [2].

**Complete subversion.** A common feature of the aforementioned works is that only some of the algorithms underlying a given cryptographic scheme are subject to subversion, while the others are assumed to faithfully follow the original specification. Motivated by this limitation, Russell *et al.* [29] put forward a new framework where the adversary is allowed to subvert *all*

[1]The PRG was standardized by NIST in 2006, and later withdrawn in 2014 as including a potential backdoor allowing to predict future outputs of the PRG algorithm.

algorithms; furthermore, in order to cast undetectability, they introduced a trusted third party, a so-called watchdog, whose goal is to test whether the (possibly subverted) implementation is compliant with the original specification of a cryptographic scheme. In a nutshell, a primitive is subversion secure if there exists a universal watchdog such that either no adversary subverting all algorithms can break security of the scheme, or, if instead a subversion attack is successful, the watchdog can detect it with non-negligible probability.

The testing procedure executed by the watchdog is typically performed only once, before the (possibly subverted) scheme is used "in the wild". This is known as the *offline* watchdog model. Unfortunately, there are subversion attacks that cannot be detected in an offline fashion. Think, *e.g.*, of a signature scheme where the signature algorithm is identical to the original specification, except that upon input a special message (that is also hard-wired in the implementation) it compromises security (*e.g.*, it returns the secret key). Now, assuming that the message space is large enough, an offline watchdog has a negligible chance of hitting this hidden trigger, so that the subverted implementation will pass the test phase; yet, the subverted scheme is clearly insecure (in the standard sense of unforgeability against chosen-message attacks).

To cast such attacks, [29] introduces the *online* watchdog model, where the watchdog is essentially allowed to additionally monitor the public interaction between users while the scheme is being used "in the wild" (on top of performing the same offline testing, as before).[2]

**Cliptography.** The main contribution of Russell *et al.* [29], apart from introducing the model of complete subversion, is to propose a methodology to clip the power of subversion attacks against one-way (trapdoor) permutations. Moreover, they show how to rely on such subversion-secure one-way permutations to derive subversion-secure pseudorandom generators and digital signatures. All their results are in the random oracle model (ROM) of Bellare and Rogaway [10].

In a follow-up paper [30], the same authors show how to obtain public-key chosen-plaintext attack secure encryption resisting complete subversion, again in the ROM. This result (inherently) requires the assumption of two *independent* secret, but tamperable, sources of randomness. They further show that their construction can be instantiated in the standard model (*i.e.*, without random oracles).

**Open questions.** The works of [29, 30] only cover a limited set of cryptographic primitives. Hence, the natural question:

> *Is it possible to protect other primitives against complete subversion, by relying on a single source of secret, but tamperable, randomness, and without assuming random oracles?*

## 1.2 Our Contributions

In this paper we make significant progress towards answering the above question. Our starting point is a notion of subversion-resistant immunizer $\Psi$, whose goal is to take an arbitrary primitive $\Pi$ that is secure w.r.t. some game $\mathbf{G}$, and transform it into an immunized primitive $\Pi^* = \Psi(\Pi)$ (for the same cryptographic task) that is secure w.r.t. $\mathbf{G}$ under complete subversion (in the sense of [29]). The immunizer leverages two *independent* random sources, which we denote by $\mathsf{R}$ and $\mathsf{S}$: The source $\mathsf{R}$ is an $m$-bit source which is assumed to be *secret*, but tamperable; the source $\mathsf{S}$ is an $m'$-bit source which is assumed to be *public* and (in some cases) *untamperable*. In particular, we consider 3 different models for subversion-secure immunization in the plain model:

[2] One can imagine even more powerful watchdogs monitoring public transcripts while being given the user's secret keys; these are known as *omniscent* watchdogs, but will not be considered in this paper.

- **Semi-Private Immunization:** Here, the public source $\mathsf{S}$ is assumed to be untamperable and uniform, while all other algorithms are subject to subversion. Moreover, the adversary has to commit to the subversion $\widetilde{\Pi}$ of the immunized cryptographic scheme $\Pi^*$ before the seed $s$, generated by sampling from $\mathsf{S}$ and later used within $\widetilde{\Pi}$, is made public.
- **Public Immunization:** Here, the public source $\mathsf{S}$ is again assumed to be untamperable and uniform, while all other algorithms are subject to subversion. However, the subversion $\widetilde{\Pi}$ is now allowed to depend on the seed $s$ (*i.e.*, first $s$ is sampled and made public, and then the adversary subverts $\Pi^*$).
- **Transparent Immunization:** Here, the public source $\mathsf{S}$ is tamperable, so that the adversary is allowed to specify its own public source $\widetilde{\mathsf{S}}$. However, tampering with the public source happens independently of the subversion of the immunized scheme (*i.e.*, first $s$ is generated from $\widetilde{\mathsf{S}}$ and made public, and then the adversary subverts $\Pi^*$).

Next, we show how to construct a subversion-secure immunizer for each of the above models. Our immunizer is tailored to protect *deterministic* primitives $\Pi$ (secure w.r.t. some game $\mathbf{G}$), where the latter means that the original specification of $\Pi$ consists of a secret random $m$-bit source $\mathsf{R}$ that is sampled in order to generate the public/secret keys of the scheme (via an algorithm $\mathsf{K}$), and the public parameters (via an algorithm $\mathsf{P}$), whereas every other algorithm $\mathsf{F}_i$ underlying $\Pi$ is deterministic. For the semi-private model, our immunizer can be instantiated using any one-way function, and works starting with an arbitrary deterministic primitive. For the public model, our immunizer can be instantiated using sufficiently strong collision-resistant hash functions, but for certain primitives $\Pi$ an additional property is required; see §1.3 for details. For the transparent model, we obtain the same results as in the public model but we now have to further assume that $\Pi$ has sub-exponential security.

Interestingly, our results allow to protect new cryptographic primitives against complete subversion; examples include: (weak) pseudorandom functions and permutations, message authentication codes, collision/second pre-image/pre-image resistant hash functions, deterministic symmetric encryption, and more. Previously to our work, for the aforementioned primitives, it was only known how to obtain security in weaker models of subversion, or with random oracles. We refer the reader to Table 1 for a comparison of our results with the state of the art.

**Immunization in the random oracle model.** To complement our results, in §8 we revisit the question of subversion-secure immunization in the ROM. In particular, we generalize a construction proposed in [29] in order to protect specific primitives, and show that it can be used to successfully immunize all deterministic primitives in the ROM by leveraging a *single* and *tamperable* random source.

Along the way, we clarify a subtle issue that was overlooked by [29]. In particular, we argue that the original proof breaks if one explicitly considers subversion of the random oracle in the offline watchdog model. Intuitively, this is because an offline watchdog is not powerful enough to detect subversion of the random oracle. We refer the reader to §8.4.2 for a more detailed discussion of this aspect.

## 1.3 Techniques

We turn to a high-level description of the techniques behind our results. Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a *deterministic* cryptographic scheme. As explained above, algorithms $\mathsf{P}$ and $\mathsf{K}$ are responsible to generate, respectively, global public parameters $\rho$ and a public/secret key pair $(pk, sk)$ that are taken as input by all other algorithms $\mathsf{F}_i$.[3] Importantly, all algorithms are

[3]The string $pk$ might be empty for secret-key primitives.

| Reference | Primitive | Complete Subversion | Watchdog | # of Sources | | Additional Assumptions | Notes |
|---|---|---|---|---|---|---|---|
| | | | | Pub | Sec | | |
| [9] | CPA-SKE | ✗ | Off | 0 | 1 🟩 | – | Unique ciphertexts |
| [3] | SIG | ✗ | Off | 0 | 1 🟩 | – | Unique signatures |
| [16] | PRG | ✗ | Off | 1 🟩 | 1 🟩 | ROM | – |
| [29] | OWF/TDF | ✓ | Off | 0 | 1 | ROM | – |
| | PRG | ✓ | Off | 0 | 1 | ROM | – |
| | SIG | ✓ | On | 0 | 1 | ROM | – |
| [30] | CPA-PKE | ✓ | Off | 0 | 1 | ROM | – |
| | CPA-PKE | ✓ | Off | 0 | 1 | OWF | – |
| [12] | SIG | ✓ | Off | 0 | 1 | ROM | – |
| §4 | $\forall$ det-const | ✓ | Off | 1 🟩 | 1 | OWF | Semi-Private |
| §5 | $\forall$ det-const-unp | ✓ | Off | 1 🟩 | 1 | CRH | Public |
| | $\forall$ det-const-ind$^2$ | ✓ | Off | 1 🟩 | 1 | CRH | Single instance |
| §6 | $\forall$ det-const-unp | ✓ | Off | 1 | 1 | CRH | Transparent |
| | $\forall$ det-const-ind$^2$ | ✓ | Off | 1 | 1 | CRH$^\dagger$ | Single instance |
| §7.1 | $\forall$ det-unconst | ✓ | On | 1 🟩 | 1 | OWF | Semi-Private |
| §7.2 | $\forall$ det-unconst-unp | ✓ | On | 1 🟩 | 1 | CRH | Public |
| | $\forall$ det-unconst-ind$^2$ | ✓ | On | 1 🟩 | 1 | CRH | Single instance |
| §7.3 | $\forall$ det-unconst-unp | ✓ | On | 1 | 1 | CRH | Transparent |
| | $\forall$ det-unconst-ind$^2$ | ✓ | On | 1 | 1 | CRH$^\dagger$ | Single instance |
| §8 | $\forall$ det | ✓ | On | 0 | 1 | ROM | – |

Table 1: Comparing our constructions with other results for security under subversion. We use the following abbreviations: "Pub" for public, "Sec" for secret, "Off" for offline, "On" for online, "CPA-SKE/CPA-PKE" for public/secret-key encryption under chosen-plaintext attacks, "PRG" for pseudo-random generator, "OWF/TDF" for one-way (trapdoor) function, "CRH" for collision-resistant hash function, "ROM" for random oracle model, "$\forall$ det" for all deterministic primitives, "$\forall$ det-const" for all deterministic primitives with security w.r.t. an input-constrained game, "$\forall$ det-unconst" for all deterministic primitives with security w.r.t. an input-unconstrained game (cf. §2.2), "$\forall$ det-const-unp" (resp. "$\forall$ det-unconst-unp") for all deterministic primitives with security w.r.t. an input-constrained (resp. input-unconstrained) unpredictability game, "$\forall$ det-const-ind$^2$" (resp. "$\forall$ det-unconst-ind$^2$") for all deterministic primitives with *square* security w.r.t. an input-constrained (resp. input-unconstrained) indistinguishability game (cf. §2.3). The green color means the source is assumed to be untamperable. $^\dagger$Requires complexity leveraging.

deterministic, except for $\mathsf{P}$ and $\mathsf{K}$ which further take as input independent random coins $r \in \{0,1\}^m$ generated by sampling a secret, uniformly random, source $\mathsf{R}$.

Our immunization strategy follows the design principle of "decomposition and trusted amalgamation" introduced in [30], by means of hash functions $h_{s_1}, h_{s_2} : \{0,1\}^n \to \{0,1\}^m$ with seeds $s_1, s_2 \in \{0,1\}^\ell$, generated by sampling (possibly multiple times) from the public $m'$-bit source $\mathsf{S}$. More in details, we generate $2k' := 2\ell/m'$ samples $s_1^1, \ldots, s_{k'}^1$ and $s_1^2, \ldots, s_{k'}^2$ from the (possibly subverted) public source $\mathsf{S}$, and set $s_1 = s_1^1 || \cdots || s_{k'}^1$ and $s_2 = s_1^2 || \cdots || s_{k'}^2$. We adopt a similar approach to generate the random coins for $\mathsf{P}$ and $\mathsf{K}$, *i.e.*, we take $2k := 2n/m$ samples $r_1^1, \ldots, r_k^1$ and $r_1^2, \ldots, r_k^2$ from the (possibly subverted) secret source $\mathsf{R}$, and set $r_1 := r_1^1 || \cdots || r_k^1$ and $r_2 := r_1^2 || \cdots || r_k^2$. Then, we hash the amalgamated strings $r_1$ and $r_2$, respectively, using seeds $s_1$ and $s_2$, so that the immunized parameter generation algorithm $\mathsf{P}^*$ runs $\mathsf{P}(1^\lambda; h_{s_1}(r_1))$, whereas the immunized key generation algorithm $\mathsf{K}^*$ runs $\mathsf{K}(1^\lambda; h_{s_2}(r_2))$; the algorithms $(\mathsf{F}_i)_{i \in N}$ are not modified. As explained below, the security analysis will crucially rely on the assumption that the above amalgamation is trusted and cannot be subverted.

Intuitively, the above immunizer tries to sanitize the randomness used for parameters/keys generation in such a way that it is harder for an adversary to generate such values together with a backdoor. We stress that the trick of hashing the random coins for key generation was introduced by [29], although there it was applied only to immunize trapdoor permutations in the ROM, whereas we generalize their approach in such a way that it can be applied to a large class of deterministic primitives (as defined above) in the plain model.

**Input constrained/unconstrained games.** Recall that for some primitives it is inherently impossible to obtain subversion security in the offline watchdog model. Hence, in our analysis of the above immunizer, we identify a natural property of cryptographic games which allows us to prove security in the *offline* watchdog model; for games not satisfying this property we instead obtain security in the *online* watchdog model.

More in details, a game $\mathbf{G}$ for some primitive $\Pi$ consists of an interaction between an adversary $\mathsf{A}$ and a challenger $\mathsf{C}$, where $\mathsf{C}$ is given oracle access to the algorithms underlying $\Pi$ in order to answer queries from $\mathsf{A}$, and determine whether $\mathsf{A}$ wins the game or not. We call $\mathbf{G}$ *input constrained*, if the inputs $x_i$ upon which each (deterministic) algorithm $\mathsf{F}_i$ is queried during the game are sampled by $\mathsf{C}$ via some public distribution $D_i$ that is independent of the adversary. On the other hand, a game that is not input constrained is called *input unconstrained.* Examples of input-constrained games $\mathbf{G}$ include, *e.g.*, the standard security games for weak pseudorandom functions and one-way permutations. See §2.2 for more examples.

**Semi-private immunization.** In the semi-private model, we prove security of the above immunizer assuming the hash functions $h_{s_1}, h_{s_2}$ are strong computational randomness extractors (which in turn can be obtained from one-way functions [13]). In this case, we can further assume $m' = \ell$ (*i.e.*, the public seeds $s_1, s_2$ are sampled directly from $\mathsf{S}$, without requiring trusted amalgamation).

Let us start by describing the original subversion game. Here, first the adversary specifies a subversion $\widetilde{\Pi}$ for the immunized cryptosystem, and then the seeds $s_1, s_2 \in \{0,1\}^\ell$ are generated by sampling the public source $\mathsf{S}$; hence, the adversary (now knowing the seeds $s_1, s_2$) interacts with the challenger, which first obtains random coins $r_1 = r_1^1||\cdots||r_k^1$ and $r_2 = r_1^2||\cdots||r_k^2$ by amalgamating $2k = 2n/m$ independent samples from the subverted source $\widetilde{\mathsf{R}}$ , and then plays[4] the game $\mathbf{G}$ for $\Pi$, given oracle access to the subverted algorithms $\widetilde{\mathsf{P}}, \widetilde{\mathsf{K}}, (\widetilde{\mathsf{F}}_i)_{i\in[N]}$ using seeds $s_1, s_2$. By contradiction, assume that there is an adversary $\mathsf{A}$ that wins the subversion game, but for which no watchdog $\mathsf{W}$ can detect the subversion. We then proceed with a sequence of hybrids, as outlined below:

1. In the 1st hybrid, we replace algorithms $\widetilde{\mathsf{K}}$, $\widetilde{\mathsf{P}}$, and $\widetilde{\mathsf{F}}_i$, with their genuine immunized implementation $\mathsf{K}^*(1^\lambda;\cdot) = \mathsf{K}(1^\lambda; h_{s_1}(\cdot))$, $\mathsf{P}^*(1^\lambda;\cdot) = \mathsf{P}(1^\lambda; h_{s_2}(\cdot))$, and $(\mathsf{F}_i^*)_{i\in[N]} = (\mathsf{F}_i)_{i\in[N]}$. One can show that any distinguisher between the original game and this hybrid can be turned into an efficient offline watchdog $\mathsf{W}$ detecting the subversion of $\mathsf{A}$. Thus, the two experiments are computationally close.

2. In the 2nd hybrid, we now generate the public parameters and the keys by running $\mathsf{P}(1^\lambda; r_1)$ and $\mathsf{K}(1^\lambda; r_2)$, where $r_1, r_2$ are uniformly random. To argue indistinguishability, assume for simplicity that the subverted source $\widetilde{\mathsf{R}}$ is stateless.[5] First, we show that $\widetilde{\mathsf{R}}$ must have at least one bit of min-entropy, as otherwise it is again possible to construct a watchdog $\mathsf{W}$ that detects subversion.

   Second, we argue that since $\widetilde{\mathsf{R}}$ is stateless and efficiently sampleable, the strings $r_1 = r_1^1||\cdots||r_k^1$ and $r_2 = r_1^2||\cdots||r_k^2$ have min-entropy at least $k$, so that indistinguishability of the two experiments follows by security of the strong computational randomness extractor. Note that the last step is possible because the public random source $\mathsf{S}$ is untamperable, and moreover, the subverted random source $\widetilde{\mathsf{R}}$ is independent of $\mathsf{S}$, as the adversary chooses it before the actual seed for the extractor is sampled.

[4]In case the game requires to query oracles $\mathsf{P}, \mathsf{K}$ multiple times, we can simply amalgamate more samples from the subverted source $\widetilde{\mathsf{R}}$ in the same way.

[5]The case of stateful subversion can be reduced to that of stateless subversion if we assume that watchdogs are allowed to reset the state of a tested implementation, a trick due to [29].

3. Finally, we observe that the last hybrid is identical to the original security game $\mathbf{G}$ for the primitive $\Pi$, which allows us to invoke the security of $\widetilde{\Pi}$ and conclude the proof.

The proof of security for the case of input unconstrained games is very similar, the main difference being that during the transition between the first and the second hybrid we need the power of an online watchdog in order to detect the subversion. Intuitively, this is because the adversary in the security game $\mathbf{G}$ for $\Pi$ can choose the inputs for the oracle queries of the challenger, which enables, *e.g.*, input-triggered attacks.

**Public immunization.** In order for the above proof strategy to work in the public immunization model, we would need a randomness extractor for the so-called *seed-dependent* setting, where the distribution of the source can depend on the seed. This natural scenario was studied by Trevisan and Vadhan [34], who came to the pessimistic conclusion that such extractors cannot exist, unless the complexity of the extractor is dependent on the complexity of the source sampler (*i.e.*, the adversary). To overcome this obstacle, we replace the randomness extractor with a min-entropy condenser, which only guarantees that the output has min-entropy; luckily, such condensers can exist even for the challenging seed-dependent setting, and were already constructed by Dodis *et al.* [18] using sufficiently strong collision-resistant hash functions.

Using randomness condensers, we can show that the 1st hybrid experiment is computationally close to an experiment in which we use $\mathsf{P}(1^\lambda; y_1)$ and $\mathsf{K}(1^\lambda; y_1)$ in order to sample the public parameters and the keys, where $y_1, y_2$ are now only guaranteed to have $m - k$ bits of min-entropy. Finally, in order to conclude the proof, we exploit the framework of "overcoming weak expectations" by Dodis and Yu [19], who established that for a *large class* of primitives[6] there is a natural trade off between concrete security and the capacity to withstand a certain entropy deficiency $d$ on the distribution of the key (see §2.3 for more details). A technical challenge here comes from the fact that this framework only applies to cryptosystems $\Pi$ where the secret key is uniformly random (and moreover there are no public parameters, or those are generated using uniform randomness). However, we show a similar tradeoff still holds for our specific setting, at least for *single-instance* games where the original random source $\mathsf{R}$ is sampled only twice (one for generating the public parameters, and one for sampling the keys).[7]

**Transparent immunization.** We finally turn to the scenario in which the public source $\mathsf{S}$ is tamperable. In this case, we cannot assume anymore that the seed for the condenser is uniform. However, note that the watchdog still guarantees that the subverted public source $\widetilde{\mathsf{S}}$ has at least one bit of min-entropy, and we can leverage trusted amalgamation in order to ensures that the seeds $s_1 = s_1^1 || \cdots || s_{k'}^1$ and $s_2 = s_1^2 || \cdots || s_{k'}^2$ obtained by concatenating $2k' = 2\ell/m'$ independent samples from $\widetilde{\mathsf{S}}$, have min-entropy at least $k'$. Hence, in order for the proof to go through, we need a stronger type of seed-dependent condenser which works even under weak seeds.

We show how to construct such condensers from any ensemble of collision-resistant hash functions with strong-enough security, by exploiting again the framework of Dodis and Yu [19]. The drawback is that this construction suffers from a higher entropy loss $d \in \omega(\log(\lambda))$, with the consequence that we are only able to prove security of our immunizer in the transparent model when starting with a cryptographic primitive $\Pi$ with sub-exponential security.

[6]In particular, the result of [19] applies to all unpredictability primitives, and to all indistinguishability primitives meeting so called *square* security; see §2.3 for details.

[7]Hence, our results for public/transparent immunization do not cover, *e.g.*, multi-instance games where several public parameters and keys might be generated.

## 1.4 Comparison with Russell *et al.* [29, 30]

The trick of splitting a cryptographic algorithm into several sub-components (as we do for $\mathsf{P}, \mathsf{K}, \mathsf{R}$) was originally introduced in [29], and later refined in [30], under the name of "split-program" methodology. Remarkably, [30] shows that for semantically-secure public-key encryption (an inherently *randomized* primitive) de-coupling the encryption algorithm in a randomized component $\mathsf{R}$ (for generating the random coins) and a deterministic component $\mathsf{Enc}$ (for computing the ciphertext) is not sufficient to defeat kleptographic attacks. For this reason, they propose a "double-splitting" technique in the ROM where $\mathsf{R}$ is further split into two (tamperable) components $\mathsf{R}_1, \mathsf{R}_2$.[8] In this perspective, our immunization strategy can be thought of as a form of "double splitting", where one of the two sources is assumed to be untamperable but made *public*, defining a more challenging scenario in which the adversary may exploit the knowledge of the seed $s$.[9] We also note that in [30], each component $\mathsf{R}_i$ can be simulated by sampling from a single source $\mathsf{R}$ multiple times if the watchdog is able to certify that the output distribution of $\mathsf{R}$ produces the required amount of entropy (*e.g.*, min-entropy $\omega(\log(\lambda))$).

The fact that subversion-secure immunization in the offline watchdog model only works for input-constrained games is reminiscent of a general observation made in [29] stating that an offline watchdog can always detect the subversion of deterministic algorithms with public input distributions (see [29, Lemma 2.3]).

Finally, we would like to stress that our work only covers immunization against complete subversion in the form of algorithm-substitution attacks. In particular, the adversary always specifies an algorithm $\widetilde{\mathsf{P}}$ that is used for sampling the public parameters during the security game. Hence, our immunizers do not provide any guarantee in the "adversarially chosen parameters model" considered in [16, 29, 15, 20] (where the adversary specifies directly the malicious public parameters).

## 1.5 Further Related Work

The original attacks in the kleptographic setting extended previous work on subliminal channels by Simmons [32, 33]. This research is also intimately connected to the problem of steganography, whose goal in the context of secret communication is to hide the mere fact that messages are being exchanged [25].

Dodis *et al.* [16], study different immunization strategies for backdoored pseudorandom generators. While they do not consider complete subversion, as the immunizer and the PRG algorithm are assumed to be trusted, they deal with the case where a cryptographic scheme might be subverted "by design" (*e.g.*, because it is standardized with maliciously generated public parameters).

Another line of work suggests to defeat subversion attacks by means of a cryptographic reverse firewall [26, 17, 3, 22]. Such a firewall is used to re-randomize the incoming/outgoing messages of a potentially subverted primitive. The firewall itself is assumed to be trusted, and moreover it relies on a secret, and untamperable, random source. Yet another approach consists of designing self-guarding schemes [21], which allow to defeat subversion without relying on external parties (such as watchdogs or reverse firewalls), at the price of assuming a secure initialization phase where the primitive to protect was not under subversion.

**Conference version.** An abridged version of this work appeared in [1]. The original paper only contained our result for public immunization, whereas the current version adds the results

[8]They additionally show that their technique can be extended in the standard model by splitting $\mathsf{R}$ into $\ell = \lambda^\delta / \log(\lambda) + \log(\lambda)$ (for some small $\delta$) components $\mathsf{R}_1, \ldots, \mathsf{R}_\ell$.

[9]In the transparent model we allow the adversary to tamper even with the public source $\mathsf{S}$.

for semi-private and transparent immunization, and the instantiation in the ROM.

### 1.6 Organization

After setting-up some notation and recalling a few standard notions in §2, we formalize the notions of subversion-secure immunization the semi-private/public/transparent model in §3. The description of our immunizer, together with its security analysis, appear in §4 (semi-private immunization), §5 (public immunization), and §7 (transparent immunization). In §8, we analyze the security of our immunizer in the ROM. Finally, we conclude the paper and state possible directions for future research in §9.

## 2 Preliminaries

### 2.1 Notation

We use the notation $[n] := \{1, \dots, n\}$. Capital letters (such as $X$) are used to denote random variables, caligraphic letters (such as $\mathcal{X}$) to denote sets, and sans serif letters (such as $\mathsf{A}$) to denote algorithms. All algorithms in this paper are modelled as (possibly interactive) Turing machines.

For a string $x \in \{0,1\}^*$, we let $|x|$ be its length; if $\mathcal{X}$ is a set, $|\mathcal{X}|$ represents the number of elements in $\mathcal{X}$. When $x$ is chosen randomly in $\mathcal{X}$, we write $x \stackrel{\$}{\leftarrow} \mathcal{X}$. If $\mathsf{A}$ is an algorithm, we write $y \stackrel{\$}{\leftarrow} \mathsf{A}(x)$ to denote a run of $\mathsf{A}$ on input $x$ and output $y$; if $\mathsf{A}$ is randomized, then $y$ is a random variable and $\mathsf{A}(x; r)$ denotes a run of $\mathsf{A}$ on input $x$ and (uniform) randomness $r$. An algorithm $\mathsf{A}$ is *probabilistic polynomial-time* (PPT) if $\mathsf{A}$ is randomized and for any input $x, r \in \{0,1\}^*$ the computation of $\mathsf{A}(x; r)$ terminates in a polynomial number of steps (in the size of the input). We denote the expected value of a random variable $X$ as $\mathbb{E}[X]$, and the probability that $X$ takes at $x \in \mathcal{X}$ as $\mathbb{P}[X = x]$.

**Negligible functions.** Throughout the paper, we denote by $\lambda \in \mathbb{N}$ the security parameter. A function $\nu : \mathbb{N} \to [0,1]$ is called *negligible* in the security parameter $\lambda$ if it vanishes faster than the inverse of any polynomial in $\lambda$, *i.e.* $\nu(\lambda) \in O(1/p(\lambda))$ for all positive polynomials $p(\lambda)$. We sometimes write $\mathsf{negl}(\lambda)$ (resp., $\mathsf{poly}(\lambda)$) to denote all negligiblie functions (resp., polynomial functions) in the security parameter.

**Unpredictability and indistinguishability.** The min-entropy of a random variable $X \in \mathcal{X}$ is $\mathbb{H}_\infty(X) := -\log \max_{x \in \mathcal{X}} \mathbb{P}[X = x]$, and intuitively it measures the best chance to predict $X$ (by a computationally unbounded algorithm). For conditional distributions, unpredictability is measured by the conditional average min-entropy $\widetilde{\mathbb{H}}_\infty(X|Y) := -\log \mathbb{E}_y \left[2^{-\mathbb{H}_\infty(X|Y=y)}\right]$.

The statistical distance between two random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, is defined as $\mathbb{SD}(X; Y) := \frac{1}{2} \sum_{v \in \mathcal{X} \cup \mathcal{Y}} |\mathbb{P}[X = v] - \mathbb{P}[Y = v]|$. Let $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ be two ensembles of random variables. We say that $X$ and $Y$ are *statistically* indistinguishable, denoted $X \approx_s Y$, as a shortening for $\mathbb{SD}(X_\lambda; Y_\lambda) \in \mathsf{negl}(\lambda)$. Similarly, we say that $X$ and $Y$ are *computationally* indistinguishable, denoted $X \approx_c Y$, if for all PPT distinguishers $\mathsf{D}$ we have $\Delta_{\mathsf{D}}(X_\lambda; Y_\lambda) \in \mathsf{negl}(\lambda)$, where

$$\Delta_{\mathsf{D}}(X_\lambda; Y_\lambda) := \left| \mathbb{P}\left[\mathsf{D}(1^\lambda, X_\lambda) = 1\right] - \mathbb{P}\left[\mathsf{D}(1^\lambda, Y_\lambda) = 1\right] \right|.$$

An ensemble $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ is efficiently sampleable if there exists a PPT algorithm $\mathsf{Samp}$ such that, for each $\lambda \in \mathbb{N}$, the output of $\mathsf{Samp}(1^\lambda)$ is distributed identically to $X_\lambda$.

## 2.2 Abstract Games

In this work, we deal with abstract cryptographic schemes. Usually, a cryptographic scheme is just a sequence of (possibly randomized) efficient algorithms. However, for our purpose, it will be convenient to specify two special algorithms which are common to any cryptographic scheme; those are the algorithms for generating the public/secret keys and the public parameters (if any). Moreover, our focus will be on *deterministic* schemes (see below).

In this vein, a deterministic cryptographic scheme is a sequence of efficient algorithms $\Pi := (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$, where:

- $\mathsf{P}$ is a deterministic algorithm that upon input the security parameter $1^\lambda$, and random coins $r \in \mathcal{R}$, outputs public parameters $\rho \in \mathcal{P}$;
- $\mathsf{K}$ is a deterministic algorithm that upon input the security parameter $1^\lambda$, and random coins $r \in \mathcal{R}$,[10] outputs a pair of keys $(pk, sk) \in \mathcal{PK} \times \mathcal{SK}$;
- The random coins for $(\mathsf{P}, \mathsf{K})$ are obtained via independent calls to algorithm $\mathsf{R}$, which outputs a uniformly random string $r \in \mathcal{R}$ upon each invocation.
- For each $i \in [N]$, algorithm $\mathsf{F}_i : \mathcal{X}_i \to \mathcal{Y}_i$ is deterministic.

We stress that the above syntax is meant to capture both secret-key and public-key primitives; in the former case the public key is simply equal to the empty string $pk = \varepsilon$, and $\mathcal{PK} = \emptyset$. Further, without loss of generality, we assume that all algorithms $\mathsf{F}_1, \ldots, \mathsf{F}_N$ take as (implicit) input both $\rho$ and $(pk, sk)$; the key generation algorithm also receives $\rho$ as additional input.

Typically, a cryptographic scheme must meet two properties. The first is a *correctness* requirement, which essentially says that $\Pi$ correctly implements the desired functionality;[11] although we will not define correctness in general, we will later assume $\Pi$ meets some well-defined correctness property. The second is a *security* requirement, which we model as an interactive process (a.k.a. game) between an adversary and a challenger.

**Definition 1** (Cryptographic game)**.** A cryptographic game $\mathbf{G} := (\mathsf{C}, \gamma)$ is defined by a challenger $\mathsf{C}$ and a constant $\gamma \in [0, 1)$; the game is (implicitly) parametrized by a cryptographic scheme $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$, an adversary $\mathsf{A}$, and the security parameter $\lambda \in \mathbb{N}$. In an execution of the game the (efficient) challenger $\mathsf{C}(1^\lambda)$ interacts with the (efficient) adversary $\mathsf{A}(1^\lambda)$, and at the end the challenger outputs a decision bit $d \in \{0, 1\}$. We denote the output of the game as $d \overset{\$}{\leftarrow} \langle \mathsf{A}(1^\lambda), \mathsf{C}^{\mathsf{P},\mathsf{K},\mathsf{R},(\mathsf{F}_i)_{i\in[N]}}(1^\lambda)\rangle$; we sometimes also write $(d, \tau) \overset{\$}{\leftarrow} (\mathsf{A}(1^\lambda) \leftrightarrows \mathsf{C}^{\mathsf{P},\mathsf{K},\mathsf{R},(\mathsf{F}_i)_{i\in[N]}}(1^\lambda))$ for a transcript of the interaction between the adversary and the challenger, $\mathsf{C}^\Pi$ as a shorthand for $\mathsf{C}^{\mathsf{P},\mathsf{K},\mathsf{R},(\mathsf{F}_i)_{i\in[N]}}$, and $\mathbf{G}_{\Pi,\mathsf{A},\mathsf{C}}$ for the random variable corresponding to an execution of game $\mathbf{G}$ with scheme $\Pi$, adversary $\mathsf{A}$, and challenger $\mathsf{C}$.

We say that $\Pi$ is $(t, \epsilon)$-*secure* w.r.t. game $\mathbf{G} = (\mathsf{C}, \gamma)$ if the following holds: For all probabilistic attackers $\mathsf{A}$ running in time $t$ we have

$$\left|\mathbb{P}\left[d = 1 : \; d \overset{\$}{\leftarrow} \langle \mathsf{A}(1^\lambda), \mathsf{C}^{\mathsf{P},\mathsf{K},\mathsf{R},(\mathsf{F}_i)_{i\in[N]}}(1^\lambda)\rangle\right] - \gamma\right| \leq \epsilon.$$

Moreover, whenever for all $t \in \mathsf{poly}(\lambda)$ there exists $\epsilon \in \mathsf{negl}(\lambda)$ such that $\Pi$ is $(t, \epsilon)$-secure w.r.t. game $\mathbf{G}$, we simply say that $\Pi$ is secure w.r.t. game $\mathbf{G}$.

[10] We assume the amount of randomness to generate the public parameters and the keys is the same; a generalization is straightforward.

[11] For instance, if $\Pi$ is a signature scheme, correctness demands that honestly computed signatures (w.r.t. a valid secret key) always verify correctly (w.r.t. the corresponding public key).

**Input-constrained games.** An important distinction will be whether the adversary is allowed or not to choose the inputs for the oracle calls made by the challenger. We call games where the latter is not possible *input-constrained* games.

**Definition 2** (Input-constrained games)**.** Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a cryptographic scheme, and $\mathbf{G} = (\mathsf{C}, \gamma)$ be a security game for $\Pi$. We call $\mathbf{G}$ *input constrained* if the following holds: For each $i \in [N]$, there exists a *public* and *efficiently samplable* distribution $D_i$, such that the challenger chooses the inputs to each oracle $\mathsf{F}_i$ by sampling a fresh and independent value from $D_i$.

In contrast, games where the above property is not met are called *input unconstrained.* Below, we give some examples of cryptographic games that are either input unconstrained or input constrained.

**One-Way Functions:** A one-way function (OWF) is a cryptographic scheme $\Pi = (\mathsf{P}, \mathsf{R}, \mathsf{OWF})$ where $N = 1$, and $\mathsf{OWF} : \mathcal{X} \to \mathcal{Y}$ is a function. Security of $\Pi$ is characterized by a game $\mathbf{G}^{\mathrm{owf}} = (\mathsf{C}_{\mathrm{owf}}, 0)$ defined as follows: (i) $\mathsf{C}_{\mathrm{owf}}$ picks $\rho = \mathsf{P}(1^\lambda; r)$ (for uniform $r \overset{\$}{\leftarrow} \mathsf{R}(1^\lambda)$), samples $x \overset{\$}{\leftarrow} \mathcal{X}$, computes $y = \mathsf{OWF}(1^\lambda, \rho, x)$, and sends $(\rho, y)$ to the adversary; (ii) $\mathsf{A}$ wins iff it returns a values $x' \in \mathcal{X}$ such that $\mathsf{OWF}(1^\lambda, \rho, x') = y$. Notice that $\mathsf{C}_{\mathrm{owf}}$ needs to invoke oracle $\mathsf{OWF}$ upon input $x'$ in order to determine the decision bit $d$, and thus the game is input unconstrained.

**One-Way Permutations:** A one-way permutation (OWP) is a cryptographic scheme $\Pi = (\mathsf{P}, \mathsf{R}, \mathsf{OWP})$ where $N = 1$, and $\mathsf{OWP} : \mathcal{X} \to \mathcal{X}$ is a permutation. Security of $\Pi$ is characterized by a game $\mathbf{G}^{\mathrm{owp}} = (\mathsf{C}_{\mathrm{owp}}, 0)$ defined as follows: (i) $\mathsf{C}_{\mathrm{owp}}$ picks $\rho = \mathsf{P}(1^\lambda; r)$ (for uniform $r \overset{\$}{\leftarrow} \mathsf{R}(1^\lambda)$), samples $x \overset{\$}{\leftarrow} \mathcal{X}$, computes $y = \mathsf{OWP}(1^\lambda, \rho, x)$, and sends $(\rho, y)$ to the adversary; (ii) $\mathsf{A}$ wins iff it returns a value $x' \in \mathcal{X}$ such that $x' = x$. Notice that $\mathsf{C}_{\mathrm{owp}}$ does not need to make any oracle call in order to determine the decision bit $d$, and thus the game is input constrained with public distribution $D$ equal to the uniform distribution over the domain $\mathcal{X}$.

**(Weak) Pseudorandom Functions:** A pseudorandom function (PRF) is a cryptographic scheme $\Pi = (\mathsf{P}, \mathsf{R}, \mathsf{K}, \mathsf{PRF})$ where $N = 1$, and $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a keyed function. Security of $\Pi$ is characterized by a game $\mathbf{G}^{\mathrm{prf}} = (\mathsf{C}_{\mathrm{prf}}, 1/2)$ defined as follows: (i) $\mathsf{C}_{\mathrm{prf}}$ samples a bit $b \overset{\$}{\leftarrow} \{0,1\}$, picks $\rho = \mathsf{P}(1^\lambda; r_1)$ and $\kappa = \mathsf{K}(1^\lambda, \rho; r_2)$ (where $r_1, r_2 \overset{\$}{\leftarrow} \mathsf{R}(1^\lambda)$), and sends $\rho$ to the adversary; (ii) $\mathsf{A}$ can ask queries of the form $x \in \mathcal{X}$, upon which $\mathsf{C}_{\mathrm{prf}}$ either replies with $y = \mathsf{PRF}(\kappa, x)$ (in case $b = 0$) or $y \overset{\$}{\leftarrow} \mathcal{Y}$ (in case $b = 1$); (iii) $\mathsf{A}$ returns a bit $b'$ and wins iff $b = b'$. Notice that $\mathsf{C}_{\mathrm{prf}}$ needs to invoke oracle $\mathsf{PRF}$ upon inputs specified by the adversary, and thus the game is input unconstrained.

For *weak* PRFs the game is changed as follows: In step (ii) the queries made by the adversary are empty, and instead $\mathsf{C}_{\mathrm{prf}}$ samples $x \overset{\$}{\leftarrow} \mathcal{X}$ and returns $(x, y)$, where $y$ is computed as before. Hence, the game is constrained with public distribution equal to the uniform distribution over $\mathcal{X}$.

**Hash Functions:** A cryptographic hash function is a cryptographic scheme $\Pi = (\mathsf{P}, \mathsf{R}, \mathsf{Hash})$ where $N = 1$, and $\mathsf{Hash} : \mathcal{X} \to \mathcal{Y}$ is a (typically compressing) function. Security of $\Pi$ is characterized by a game $\mathbf{G}^{\mathrm{cr}} = (\mathsf{C}_{\mathrm{cr}}, 0)$ defined as follows: (i) $\mathsf{C}_{\mathrm{cr}}$ picks $\rho = \mathsf{P}(1^\lambda; r)$ (for uniform $r \overset{\$}{\leftarrow} \mathsf{R}(1^\lambda)$), and sends $\rho$ to the adversary; (ii) $\mathsf{A}$ wins iff it returns a pair of values $(x, x') \in \mathcal{X}^2$ such that $\mathsf{Hash}(1^\lambda, \rho, x) = \mathsf{Hash}(1^\lambda, \rho, x')$ and $x \neq x'$. Notice that $\mathsf{C}_{\mathrm{cr}}$ needs to invoke oracle $\mathsf{Hash}$ upon input $x, x'$ in order to determine the decision bit $d$, and thus the game is input unconstrained.

**Message Authentication Codes:** A message authentication code (MAC) is a cryptographic scheme $\Pi = (\mathsf{P}, \mathsf{R}, \mathsf{K}, \mathsf{MAC})$ where $N = 1$, and $\mathsf{MAC} : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is a keyed function. Security of $\Pi$ is characterized by a game $\mathbf{G}^{\mathrm{mac}} = (\mathsf{C}_{\mathrm{mac}}, 0)$ defined as follows: (i) $\mathsf{C}_{\mathrm{mac}}$ picks $\rho = \mathsf{P}(1^\lambda; r_1)$ and $\kappa = \mathsf{K}(1^\lambda, \rho; r_2)$ (where $r_1, r_2 \overset{\$}{\leftarrow} \mathsf{R}(1^\lambda)$), and sends $\rho$ to the adversary; (ii) $\mathsf{A}$ can ask queries of the form $m \in \mathcal{M}$, upon which $\mathsf{C}_{\mathrm{mac}}$ replies with $\tau = \mathsf{MAC}(\kappa, m)$; (iii) $\mathsf{A}$ returns a forgery $(m^*, \tau^*)$ and wins iff $\tau^* = \mathsf{MAC}(\kappa, m^*)$ and additionally $m^*$ was never part of a query at step (ii). Notice that $\mathsf{C}_{\mathrm{mac}}$ needs to invoke oracle $\mathsf{MAC}$ upon inputs specified by the adversary, and thus the game is input unconstrained.

**Secret-Key Encryption:** A deterministic secret-key encryption scheme is a cryptographic scheme $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{Enc}, \mathsf{Dec})$ where $N = 2$. The (deterministic) encryption algorithm takes as input the secret key $\kappa \in \mathcal{K}$ and a message $m \in \mathcal{M}$, and outputs a ciphertext $c \in \mathcal{C}$. The (deterministic) decryption algorithm takes as input the secret key $\kappa \in \mathcal{K}$ and a ciphertext $c \in \mathcal{C}$, and outputs a message $m \in \mathcal{M}$ (or an error symbol). Security of a deterministic encryption scheme is characterized, *e.g.*, by a game $\mathbf{G}^{\text{cca-ske}} = (\mathsf{C}_{\text{cca-ske}}, 1/2)$ specified as follows: (i) $\mathsf{C}_{\text{cca-ske}}$ picks $\rho = \mathsf{P}(1^\lambda; r_1)$ and $\kappa = \mathsf{K}(1^\lambda, \rho; r_2)$ (where $r_1, r_2 \overset{\$}{\leftarrow} \mathsf{R}(1^\lambda)$), and sends $\rho$ to the adversary; (ii) $\mathsf{A}$ can specify encryption queries: Upon input a message $m \in \mathcal{M}$, the challenger returns $c = \mathsf{Enc}(1^\lambda, \kappa, m)$; (iii) $\mathsf{A}$ can specify decryption queries: Upon input a ciphertext $c \in \mathcal{C}$, the challenger returns $m = \mathsf{Dec}(1^\lambda, \kappa, c)$; (iv) $\mathsf{A}$ can specify a challenge query: Upon input $(m_0^*, m_1^*) \in \mathcal{M}^2$, the challenger returns $c^* = \mathsf{Enc}(1^\lambda, \kappa, m_b^*)$ where $b \overset{\$}{\leftarrow} \{0, 1\}$ is a hidden bit; (v) $\mathsf{A}$ can continue to specify encryption/decryption queries, with the restriction that $c^*$ cannot be part of a decryption query; (vi) $\mathsf{A}$ returns a bit $b'$ and wins iff $b = b'$. Notice that $\mathsf{C}_{\text{cca-ske}}$ needs to invoke oracles $\mathsf{Enc}, \mathsf{Dec}$ in order to answer encryption/decryption queries, and thus the game is input unconstrained.

**Deterministic Signatures:** A deterministic signature is a cryptographic scheme $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{Sign}, \mathsf{Vrfy})$ where $N = 2$, and $\mathsf{Sign} : \mathcal{SK} \times \mathcal{M} \to \mathcal{S}$, $\mathsf{Vrfy} : \mathcal{VK} \times \mathcal{M} \times \mathcal{S} \to \{0, 1\}$. Security of $\Pi$ is characterized by a game $\mathbf{G}^{\mathrm{sig}} = (\mathsf{C}_{\mathrm{sig}}, 0)$ defined as follows: (i) $\mathsf{C}_{\mathrm{sig}}$ picks $\rho = \mathsf{P}(1^\lambda; r_1)$ and $(vk, sk) = \mathsf{K}(1^\lambda, \rho; r_2)$ (where $r_1, r_2 \overset{\$}{\leftarrow} \mathsf{R}(1^\lambda)$), and sends $\rho, vk$ to the adversary; (ii) $\mathsf{A}$ can ask queries of the form $m \in \mathcal{M}$, upon which $\mathsf{C}_{\mathrm{sig}}$ replies with $\sigma = \mathsf{Sign}(sk, m)$; (iii) $\mathsf{A}$ returns a forgery $(m^*, \sigma^*)$ and wins iff $\mathsf{Vrfy}(vk, m^*, \sigma^*) = 1$ and additionally $m^*$ was never part of a query at step (ii). Notice that $\mathsf{C}_{\mathrm{sig}}$ needs to invoke oracle $\mathsf{Sign}$ upon inputs specified by the adversary, and thus the game is input unconstrained.

**Single-instance games.** As mentioned in the introduction, our results only apply to a subclass of games where the random source $\mathsf{R}$ is sampled only twice, in order to obtain the randomness needed for generating the public parameters and the keys. We call such games *single instance.*

**Definition 3** (Single-instance games)**.** Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a cryptographic scheme, and $\mathbf{G} = (\mathsf{C}, \gamma)$ be a security game for $\Pi$. We call $\mathbf{G}$ *single instance* if during a game execution the challenger invokes the oracle $\mathsf{R}$ twice, in order to obtain coins $r_1, r_2$ that are later fed to oracles $\mathsf{P}, \mathsf{K}$.

## 2.3 Overcoming Weak Expectations

The original specification of a (deterministic) cryptographic scheme leverages a uniformly random source $\mathsf{R}$ that is used for obtaining the public parameters and the keys. We will call this setting the *ideal model.* Motivated by the study of cryptographic applications relying on weak secrets, where essentially secret keys are only guaranteed to have some non-trivial amount of min-entropy, Dodis and Yu [19] considered the more general setting where the source $\mathsf{R}$ is not uniform but satisfies $\mathbb{H}_\infty(R) \geq m-d$, where $R \in \{0,1\}^m$ is the random variable for the outcome of algorithm $\mathsf{R}$, and $d \geq 1$ is called the *entropy deficiency.*

The original result of [19] only applies to the setting where the source $\mathsf{R}$ is used in order to sample the secret key alone, and no public parameters are available (or those are generated using uniform randomness). Nevertheless, below we show that their result already covers the slightly more general setting of deterministic, single-instance, cryptographic primitives (as defined in §2.2). Given a cryptographic scheme $\Pi$, and a single-instance security game $\mathbf{G} = (\mathsf{C}, \gamma)$ for $\Pi$, let us define $\mathbf{Adv}_{\mathsf{A},\mathsf{C}}(r_1, r_2)$ to be the advantage of $\mathsf{A}$ against challenger $\mathsf{C}$ (in an execution of the game), when the coins $r_1, r_2 \in \{0,1\}^m$ to be used as input of algorithms $\mathsf{P}, \mathsf{K}$ are fixed, *i.e.*

$$\mathbf{Adv}_{\mathsf{A},\mathsf{C}}(r_1, r_2) := \mathbb{E}\left[\langle \mathsf{A}(1^\lambda), \mathsf{C}^{\mathsf{P},\mathsf{K},\mathsf{F}_1,\ldots,\mathsf{F}_N}(1^\lambda, r_1, r_2)\rangle\right] - \gamma,$$

where the expectation is taken over the random coin tosses of $\mathsf{A}$, $\mathsf{C}$. Similar to [19], we will refer to $|\mathbb{E}[\mathbf{Adv}_{\mathsf{A},\mathsf{C}}(U_m, U_m)]|$ as the advantage of $\mathsf{A}$ in the ideal model, and to

$$\max_{R_1,R_2} |\mathbb{E}[\mathbf{Adv}_{\mathsf{A},\mathsf{C}}(R_1, R_2)]|,$$

where the maximum is taken over all distributions $R_1, R_2 \equiv R$ with $\mathbb{H}_\infty(R) \geq m - d$, as the advantage of $\mathsf{A}$ in the $(m-d)$-real model. This naturally yields the following definition.

**Definition 4** (Real/Ideal security)**.** Let $\Pi = (\mathsf{P}, \mathsf{R}, \mathsf{K}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a deterministic cryptographic scheme with $\mathcal{R} = \{0,1\}^m$, and consider a single-instance game $\mathbf{G} = (\mathsf{C}, \gamma)$ for $\Pi$. Then, we say that $\Pi$ is $(t, \epsilon)$-secure in the $(m-d)$-real model if for all probabilistic adversaries $\mathsf{A}$ with running time at most $t$ the advantage of $\mathsf{A}$ in the real model is at most $\epsilon$.

Similarly, we say that $\Pi$ is $(t, \epsilon)$-secure in the ideal model if for all probabilistic adversaries $\mathsf{A}$ with running time at most $t$ the advantage of $\mathsf{A}$ in the ideal model is at most $\epsilon$.

It is easy to show that $(t, \epsilon)$-security in the real model as per Definition 4 is equivalent to $(t, \epsilon)$-security as per Definition 1 (when restricted to single-instance games). Now, let us call a game $\mathbf{G} = (\mathsf{C}, \gamma)$ for which $\gamma = 0$ an *unpredictability* game; on the other hand, if $\gamma = 1/2$ let us call $\mathbf{G}$ an *indistinguishability* game. For indistinguishability games it will be useful to recall the following notion of so-called *square*-security.

**Definition 5** (Square security)**.** Let $\Pi = (\mathsf{P}, \mathsf{R}, \mathsf{K}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a deterministic cryptographic scheme with $\mathcal{R} = \{0,1\}^m$, and consider a single-instance game $\mathbf{G} = (\mathsf{C}, \gamma)$ for $\Pi$. Then, we say that $\Pi$ is $(t, \epsilon)$-square-secure in the ideal model if for all probabilistic adversaries $\mathsf{A}$ with running time at most $t$ we have that $\mathbb{E}[(\mathbf{Adv}_{\mathsf{A},\mathsf{C}}(U_m, U_m))^2] \leq \epsilon$.

The theorem below says that there is a natural tradeoff between the entropy deficiency that we want to tolerate in the real model and the security of a given cryptographic primitive in the ideal model. The proof is very similar to that given in [19].

**Theorem 1.** *Let $\Pi = (\mathsf{P}, \mathsf{R}, \mathsf{K}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a deterministic cryptographic scheme, and consider a single-instance game $\mathbf{G} = (\mathsf{C}, \gamma)$ for $\Pi$. Then:*

*(i) In case $\mathbf{G}$ is an unpredictability game, whenever $\Pi$ is $(t, \epsilon)$-secure in the ideal model, it is also $(t, 2^{2d} \cdot \epsilon)$-secure in the $(m-d)$-real model.*

*(ii) In case $\mathbf{G}$ is an indistinguishability game, whenever $\Pi$ is $(t, \epsilon)$-*square-*secure in the ideal model, it is also $(t, \sqrt{2^{2d} \cdot \epsilon})$-secure in the $(m-d)$-real model.*

*Proof.* (i) Let $f : \{0,1\}^m \to \mathbb{R}^+ \cup \{0\}$ be any deterministic real-valued function. For any random variable $R$ such that $\mathbb{H}_\infty(R) \geq m - d$, and for $R_1, R_2 \equiv R$, we can write:

$$\mathbb{E}\left[f(R_1, R_2)\right] = \sum_{r_1, r_2} \mathbb{P}\left[R_1 = r_1, R_2 = r_2\right] \cdot f(r_1, r_2) \leq 2^{2d} \sum_{r_1, r_2} \frac{1}{2^{2m}} f(r_1, r_2).$$

The statement then follows by observing that when $f$ is the advantage during the unpredictability game in the $(m-d)$-real setting, the right-hand side of the above equation is the advantage of the same game in the ideal setting.

(ii) Let $f : \{0,1\}^m \to \mathbb{R}$ be any deterministic real-valued function. For any random variable $R$ such that $\mathbb{H}_\infty(R) \geq m - d$, and for $R_1, R_2 \equiv R$, we can write:

$$\begin{aligned}
\left|\mathbb{E}\left[f(R_1, R_2)\right]\right| &= \left|\sum_{r_1, r_2} \mathbb{P}\left[R_1 = r_1, R_2 = r_2\right] \cdot f(r_1, r_2)\right| \\
&\leq \sqrt{2^{2m} \sum_{r_1, r_2} \mathbb{P}\left[R_1 = r_1, R_2 = r_2\right]^2} \cdot \sqrt{\frac{1}{2^{2m}} \cdot \sum_{r_1, r_2} f(r_1, r_2)^2} \\
&\leq \sqrt{2^{2m} \sum_{r_1} \mathbb{P}\left[R_1 = r_1\right]^2 \cdot \sum_{r_2} \mathbb{P}\left[R_2 = r_2\right]^2} \cdot \sqrt{\frac{1}{2^{2m}} \cdot \sum_{r_1, r_2} f(r_1, r_2)^2} \\
&\leq \sqrt{2^{2d}} \cdot \sqrt{\frac{1}{2^{2m}} \cdot \sum_{r_1, r_2} f(r_1, r_2)^2},
\end{aligned}$$

where the first inequality follows by the Cauchy-Schwartz inequality, the second inequality follows by the fact that $R_1, R_2$ are independent, and the last inequality follows since $R_1, R_2$ have min-entropy (and thus collision entropy) at least $m - d$. The statement then follows by observing that when $f$ is the advantage during the indistinguishability game in the $(m-d)$-real setting, the right-hand side of the above equation is the expectation of the squared advantage of the same game in the ideal setting. □

Many primitives meet square security, including CPA-secure secret-key encryption and weak pseudorandom functions; in contrast, other primitives such as the one-time pad, pseudorandom generators, and pseudorandom functions do not have good square security [6, 19].

## 3 Security Model

In this section we consider a standard-model definition for subversion security, via so-called *immunizers*. An immunizer is a transformation that takes as input a cryptographic scheme (for some task) and transforms it into another scheme (for the same task) that withstands complete subversion; the immunizer is allowed to leverage a single source of public, and possibly untrusted, randomness. Importantly, we seek security in the standard model (*i.e.*, without random oracles) and in a setting where the immunizer itself is subject to subversion.[12]

We first define our model formally, in §3.1, for the case of offline watchdogs. Then, in §3.2, we discuss some definitional choices and compare our definitions with previous work in the area. Finally, in §3.3, we explain how to extend our framework to the case of online watchdogs.

[12] We refer the reader to §8 for the corresponding definitions in the random oracle model.

## 3.1 Subversion-Secure Immunizers

Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a cryptographic scheme (as defined in §2.2), where we assumed that $\mathcal{R} := \{0,1\}^m$ (*i.e.*, the *secret* source $\mathsf{R}$ is a random $m$-bit source). An immunizer for $\Pi$ is a transformation $\Psi^{\mathsf{R},\mathsf{S}}_{k,k'}$ with hard-wired parameters $k, k' \in \mathbb{N}$, and with oracle access to $\mathsf{R}$ and to a *public* random $m'$-bit source $\mathsf{S}$. We write $\Pi^* := \Psi^{\mathsf{R},\mathsf{S}}_{k,k'}(\Pi) := (\mathsf{P}^*, \mathsf{K}^*, \mathsf{R}, \mathsf{F}^*_1, \ldots, \mathsf{F}^*_N)$ for the specification of the immunized cryptosystem, where:

- $\mathsf{P}^*$ and $\mathsf{K}^*$ take as input a seed $s \in \{0,1\}^\ell$, and have $n$-bit random tapes;
- $(\mathsf{F}^*_i)_{i \in N}$ take as input a seed $s \in \{0,1\}^\ell$ plus the same inputs as the corresponding algorithm in $\Pi$.
- The random tapes $r_1, r_2 \in \{0,1\}^n$ for algorithms $\mathsf{P}^*$ and $\mathsf{K}^*$, and the seed $s \in \{0,1\}^\ell$, are obtained, respectively, by selecting and amalgamating[13] $2k$ and $k'$ independent samples from the sources $\mathsf{R}$ and $\mathsf{S}$.

We require an immunizer $\Psi$ to satisfy two properties. The first property is the usual correctness requirement, meaning that the immunized primitive $\Pi^* = \Psi^{\mathsf{R},\mathsf{S}}_{k,k'}(\Pi)$ meets the same correctness condition as that of $\Pi$ (for every possible choice of the seed). The second property is some flavor of security to subversion attacks. More in details, we consider three natural scenarios, as outlined below and depicted in Fig. 1.

- **Semi-Private Immunization:** In this setting, the public $m'$-bit source $\mathsf{S}$ is assumed to be untamperable. The adversary $\mathsf{A}$ knows a description[14] of the immunizer $\Psi$ and of the original primitive $\Pi$, and has to commit to its choice for the subverted immunized cryptosystem $\widetilde{\Pi} = (\widetilde{\mathsf{P}}, \widetilde{\mathsf{K}}, \widetilde{\mathsf{R}}, (\widetilde{\mathsf{F}}_i)_{i \in N})$ before the challenger samples (and later makes public) the actual seed $s = s_1 || \cdots || s_{k'} \in \{0,1\}^\ell$, which is obtained by amalgamating $k'$ independent samples from $\mathsf{S}(1^\lambda)$.

  Next, the adversary (now also given $s$) plays the security game for $\Pi$. Here, the challenger picks $2k$ independent samples $(r^1_i, r^2_i)_{i \in [k]}$ from $\widetilde{\mathsf{R}}$, amalgamates them into strings $r_1 = r^1_1 || \cdots || r^1_k \in \{0,1\}^n$ and $r_2 = r^2_1 || \cdots || r^2_k \in \{0,1\}^n$, and finally interacts with $\mathsf{A}$ given black-box access to $\widetilde{\mathsf{P}}(s, \cdot), \widetilde{\mathsf{K}}(s, \cdot), \widetilde{\mathsf{F}}_i(s, \cdot)$ (*i.e.*, to the subversion specified by the adversary using seed $s \in \{0,1\}^\ell$), where $r_1$ and $r_2$ are used as random tapes for $\widetilde{\mathsf{P}}$ and $\widetilde{\mathsf{K}}$, respectively.
- **Public Immunization:** Here, the adversary $\mathsf{A}$ is allowed to choose $\widetilde{\Pi}$ depending on the actual seed $s \in \{0,1\}^\ell$, which is generated as in the semi-private setting. Afterwards, $\mathsf{A}$ plays the game with the challenger $\mathsf{C}$ exactly as before.
- **Transparent Immunization:** Here, the adversary $\mathsf{A}$ can additionally tamper with the source $\mathsf{S}$ used to setup the immunizer. In particular, $\mathsf{A}$ first specifies a subverted public source $\widetilde{\mathsf{S}}$; then the challenger picks the public seed $s = s_1 || \cdots || s_{k'} \in \{0,1\}^\ell$ by amalgamating $k'$ independent samples from $\widetilde{\mathsf{S}}(1^\lambda)$. Finally, the adversary outputs the subverted immunized cryptosystem $\widetilde{\Pi}$ (which again depends on the seed $s$) and plays the game with $\mathsf{C}$ exactly as before.

For each $\mathtt{type} \in \{\text{spriv}, \text{pub}, \text{trans}\}$, we define the advantage of adversary $\mathsf{A}$ in the subversion game with primitive $\Pi$, immunizer $\Psi$, and challenger $\mathsf{C}$ as:

$$\mathbf{Adv}^{\mathtt{type}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) := \left| \mathbb{P}\left[ \mathbf{G}^{\mathtt{type}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) = 1 \right] - \gamma \right|, \quad (1)$$

[13] In what follows, we restrict ourselves to a specific type of amalgamation (*i.e.*, strings concatenation), since looking ahead this is what all of our immumizers will require.

[14] We often leave the parameters $k, k'$, and the sources $\mathsf{R}, \mathsf{S}$, implicit, in order to simplify notation.

```
Game $\mathbf{G}^{\mathrm{spriv}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda)$
$(\widetilde{\Pi}, \alpha) \xleftarrow{\$} \mathsf{A}_0(1^\lambda, \langle \Pi, \Psi \rangle)$
$s_1, \ldots, s_{k'} \xleftarrow{\$} \mathsf{S}(1^\lambda)$;
$s = s_1 || \cdots || s_{k'}$;
$r^1_1, \ldots, r^1_k, r^2_1, \ldots, r^2_k \xleftarrow{\$} \widetilde{\mathsf{R}}(1^\lambda)$;
$r_1 = r^1_1 || \cdots || r^1_k; r_2 = r^2_1 || \cdots || r^2_k$;
$(d, \widetilde{\tau}) \xleftarrow{\$} (\mathsf{A}_1(1^\lambda, \alpha, s) \leftrightarrows \mathsf{C}^{\widetilde{\mathsf{P}}(s,\cdot),\widetilde{\mathsf{K}}(s,\cdot),(\widetilde{\mathsf{F}}_i(s,\cdot))_{i \in N}}(1^\lambda, r_1, r_2))$
return $d$
```

```
Game $\mathbf{G}^{\mathrm{pub}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda)$
$s_1, \ldots, s_{k'} \xleftarrow{\$} \mathsf{S}(1^\lambda)$;
$s = s_1 || \cdots || s_{k'}$;
$(\widetilde{\Pi}, \alpha) \xleftarrow{\$} \mathsf{A}_0(1^\lambda, s, \langle \Pi, \Psi \rangle)$
$r^1_1, \ldots, r^1_k, r^2_1, \ldots, r^2_k \xleftarrow{\$} \widetilde{\mathsf{R}}(1^\lambda)$;
$r_1 = r^1_1 || \cdots || r^1_k; r_2 = r^2_1 || \cdots || r^2_k$;
$(d, \widetilde{\tau}) \xleftarrow{\$} (\mathsf{A}_1(1^\lambda, \alpha) \leftrightarrows \mathsf{C}^{\widetilde{\mathsf{P}}(s,\cdot),\widetilde{\mathsf{K}}(s,\cdot),(\widetilde{\mathsf{F}}_i(s,\cdot))_{i \in N}}(1^\lambda, r_1, r_2))$
return $d$
```

```
Game $\mathbf{G}^{\mathrm{trans}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda)$
$(\widetilde{\mathsf{S}}, \alpha_0) \xleftarrow{\$} \mathsf{A}_0(1^\lambda, \langle \Pi, \Psi \rangle)$
$s_1, \ldots, s_{k'} \xleftarrow{\$} \widetilde{\mathsf{S}}(1^\lambda)$;
$s = s_1 || \cdots || s_{k'}$;
$(\widetilde{\Pi}, \alpha_1) \xleftarrow{\$} \mathsf{A}_1(1^\lambda, \alpha_0, s)$
$r^1_1, \ldots, r^1_k, r^2_1, \ldots, r^2_k \xleftarrow{\$} \widetilde{\mathsf{R}}(1^\lambda)$;
$r_1 = r^1_1 || \cdots || r^1_k; r_2 = r^2_1 || \cdots || r^2_k$;
$(d, \widetilde{\tau}) \xleftarrow{\$} (\mathsf{A}_2(1^\lambda, \alpha_1) \leftrightarrows \mathsf{C}^{\widetilde{\mathsf{P}}(s,\cdot),\widetilde{\mathsf{K}}(s,\cdot),(\widetilde{\mathsf{F}}_i(s,\cdot))_{i \in N}}(1^\lambda, r_1, r_2))$
return $d$
```

Figure 1: Games defining subversion security of an immunizer $\Psi^{\mathsf{R},\mathsf{S}}_{k,k'}$, in the standard model. We use the notation $\mathsf{C}^{\widetilde{\mathsf{P}}(s,\cdot),\widetilde{\mathsf{K}}(s,\cdot),(\widetilde{\mathsf{F}}_i(s,\cdot))_{i \in N}}(1^\lambda, r_1, r_2)$ to denote a run of the challenger $\mathsf{C}$ with random coins $r_1, r_2$ (that will be used as input of algorithms $\widetilde{\mathsf{P}}, \widetilde{\mathsf{K}}$ during the game).

where the games $\mathbf{G}^{\mathrm{spriv}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda), \mathbf{G}^{\mathrm{pub}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda)$ and $\mathbf{G}^{\mathrm{trans}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda)$ are depicted in Fig. 1, and the probability is taken over the randomness of $\widetilde{\mathsf{S}}, \widetilde{\mathsf{R}}, \mathsf{S}, \mathsf{R}$, and over the coin tosses of $\mathsf{A}$.

Clearly, since the subverted cryptosystem $\widetilde{\Pi}$ specified by the adversary is completely arbitrary, it might be trivial to break security in the above setting.[15] Hence, we need to restrict the adversary in some way. Following previous work, we will consider the adversary to be

[15]For instance, consider $\Pi$ to be a signature scheme and the corresponding subversion to have the signing algorithm return the signing key.

```
Game $\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, b)$
$\mathtt{aux} := \langle\widetilde{\Pi}\rangle$
$\mathtt{aux} := (\langle\widetilde{\Pi}\rangle, s)$
$\mathtt{aux} := (\langle\widetilde{\mathsf{S}}, \widetilde{\Pi}\rangle, s)$
if $b = 0$
    return $\mathsf{W}^{\widetilde{\Pi}}(1^\lambda, \langle\Pi, \Psi\rangle)$
    return $\mathsf{W}^{\widetilde{\Pi}}(1^\lambda, \langle\Pi, \Psi\rangle, s)$
    return $\mathsf{W}^{\widetilde{\mathsf{S}},\widetilde{\Pi}}(1^\lambda, \langle\Pi, \Psi\rangle, s)$
elseif $b = 1$
    $\Pi^* = \Psi(\Pi)$
    return $\mathsf{W}^{\Pi^*}(1^\lambda, \langle\Pi, \Psi\rangle)$
    return $\mathsf{W}^{\Pi^*}(1^\lambda, \langle\Pi, \Psi\rangle, s)$
    return $\mathsf{W}^{\mathsf{S},\Pi^*}(1^\lambda, \langle\Pi, \Psi\rangle, s)$
fi
```

```
$\mathbf{G}^{\mathrm{det\text{-}on}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, b)$
$\mathtt{aux} := (\langle\widetilde{\Pi}\rangle, \widetilde{\tau})$
$\mathtt{aux} := (\langle\widetilde{\Pi}\rangle, s, \widetilde{\tau})$
$\mathtt{aux} := (\langle\widetilde{\mathsf{S}}, \widetilde{\Pi}\rangle, s, \widetilde{\tau})$
if $b = 0$
    return $\mathsf{W}^{\widetilde{\Pi}}(1^\lambda, \langle\Pi, \Psi\rangle, \widetilde{\tau})$
    return $\mathsf{W}^{\widetilde{\Pi}}(1^\lambda, \langle\Pi, \Psi\rangle, s, \widetilde{\tau})$
    return $\mathsf{W}^{\widetilde{\mathsf{S}},\widetilde{\Pi}}(1^\lambda, \langle\Pi, \Psi\rangle, s, \widetilde{\tau})$
elseif $b = 1$
    $\Pi^* = \Psi(\Pi)$
    return $\mathsf{W}^{\Pi^*}(1^\lambda, \langle\Pi, \Psi\rangle, \widetilde{\tau})$
    return $\mathsf{W}^{\Pi^*}(1^\lambda, \langle\Pi, \Psi\rangle, s, \widetilde{\tau})$
    return $\mathsf{W}^{\mathsf{S},\Pi^*}(1^\lambda, \langle\Pi, \Psi\rangle, s, \widetilde{\tau})$
fi
```

Figure 2: Description of the detection game for an immunizer $\Psi^{\mathsf{R},\mathsf{S}}_{k,k'}$ with offline (left) and online (right) watchdogs, in the standard model. The auxiliary information $\mathtt{aux}$ is taken from the subversion game (cf. Fig. 1). The dashed boxed code refers to the security definition for the public model (*i.e.*, the watchdog receives as input the seed from the subversion game), the colored boxed code refers to the transparent model (*i.e.*, the watchdog receives as input the seed and has access to the source distribution).

"malicious-but-proud" in the sense that in order to be successful a subversion attack should also be *undetectable* by the honest user. The latter is formalized by a detection game featuring an efficient algorithm, called the watchdog, whose goal is to detect whether a subversion took place. In particular, given a description of the immunizer and of the original scheme, the watchdog has to distinguish the immunized cryptosystem $\Pi^*$ from the subversion $\widetilde{\Pi}$ used by the adversary in the subversion game; importantly, since in the transparent model the public source $\mathsf{S}$ can be subverted by the adversary, the watchdog in the detect game gets black-box access either to $\mathsf{S}$ or to its subversion $\widetilde{\mathsf{S}}$. (In the semi-private and public models, instead, $\widetilde{\mathsf{S}} \equiv \mathsf{S}$ as the public source is untamperable.) The detect advantage of watchdog $\mathsf{W}$ is defined as:[16]

$$\mathbf{Adv}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda) := \left|\mathbb{P}\left[\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, 0) = 1\right] - \mathbb{P}\left[\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, 1) = 1\right]\right|, \tag{2}$$

where the game $\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, b)$ is depicted in Fig. 2, and the probability is taken over the randomness of $\widetilde{\mathsf{S}}, \widetilde{\mathsf{R}}, \mathsf{S}, \mathsf{R}$, and over the coin tosses of $\mathsf{W}$; the values in the auxiliary information $\mathtt{aux}$ are taken from $\mathbf{G}^{\mathsf{type}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda)$. Similarly to previous work, we assume that $\mathsf{W}$ has rewinding black-box access to its oracles, a feature required in order to detect stateful subversion [29, Remark 2.5].

[16] Of course, we could also treat the detection game as an indistinguishability game $\mathbf{G} = (\mathsf{C}, \gamma)$, and thus define the detection advantage as a function of $\gamma = 1/2$. However, we prefer the above formulation in order to be consistent with previous work [29, 30].

We are now ready to define subversion security of an immunizer for the offline watchdog.

**Definition 6** (Subversion-resistant immunizer)**.** Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a cryptographic scheme, $\mathbf{G} = (\mathsf{C}, \gamma)$ be a security game for $\Pi$, $\mathsf{S}$ be a random $m'$-bit source, and $k, k' \in \mathbb{N}$ be parameters. For a constant $c^* \geq 1$, and for $\mathtt{type} \in \{\text{spriv}, \text{pub}, \text{trans}\}$, we say that an immunizer $\Psi_{k,k'}^{\mathsf{R},\mathsf{S}}$ is $(t_\mathsf{A}, t_\mathsf{W}, c^*, \epsilon^*)$-subversion-resistant for the $\mathtt{type}$-model with an *offline* watchdog if the following holds: There exists a watchdog $\mathsf{W}$ with running time $t_\mathsf{W}$ such that for all adversaries $\mathsf{A}$ with running time $t_\mathsf{A}$ for which $\mathbf{Adv}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}^{\mathtt{type}}(\lambda) > \epsilon^*$, we have

$$\mathbf{Adv}_{\Pi,\Psi,\mathsf{W}}^{\mathrm{det}}(\lambda) \geq \frac{1}{c^*} \cdot \mathbf{Adv}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}^{\mathtt{type}}(\lambda).$$

Moreover, for all $s \in \{0,1\}^\ell$, we require that the immunized cryptosystem with seed $s$ meets the same correctness requirement as that of $\Pi$.

**Remark 1** (On subverting the immunizer)**.** We stress that the subversion $\widetilde{\Pi}$ should be thought of as the subversion of the *immunized* cryptosystem $\Pi^* = \Psi(\Pi)$. In particular, since the subversion is completely arbitrary, the latter means that the adversary can tamper with the immunizer itself.

**Remark 2** (On trusted amalgamation)**.** The only component of the immunizer that cannot be subverted by the attacker is the sampling of the random tapes $r_1, r_2$ and of the seed $s$, which are obtained by concatenating, respectively, $2k$ and $k'$ independent samples from the (possibly subverted) random sources $\widetilde{\mathsf{R}}, \widetilde{\mathsf{S}}$.

Looking ahead, trusted amalgamation is needed in the security analysis of our immunizers for the transparent model, in order to argue that both $r_1, r_2$ and $s$ have a sufficient amount of min-entropy even when the public/secret random source is subverted. On the other hand, trusted amalgamation of the seed is not needed in the semi-private and public model (as the source $\mathsf{S}$ is assumed to be untamperable), where we can directly replace $\mathsf{S}$ with $\mathsf{S}' = \mathsf{S}^{k'}$.

**Remark 3** (On including the seed in the auxiliary information)**.** Note that the seed $s$ sampled during the subversion game is part of the auxiliary information $\mathtt{aux}$, and later given as additional input to the watchdog in the detection game.

It is easy to see that the latter is necessary. Consider, for instance, a deterministic signature scheme $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{Sign}, \mathsf{Vrfy})$ (cf. §2.2), and let $\Pi^* = (\mathsf{P}^*, \mathsf{K}^*, \mathsf{R}, \mathsf{Sign}^*, \mathsf{Vrfy}^*) = \Psi(\Pi)$ be the immunized version of $\Pi$. Since the subversion $\widetilde{\Pi}$ is allowed to depend on the seed $s$, the adversary could instruct $\widetilde{\mathsf{K}}$ to output a fixed verification/signature key pair $(\overline{vk}, \overline{sk})$, known to the adversary, whenever $\widetilde{\mathsf{K}}$ is run upon input $s$. Now, if the watchdog $\mathsf{W}$ would not be given as input the actual seed $s$, the above attack would be undetectable, as $\mathsf{W}$ has only a negligible chance of hitting the seed $s$ while sampling the source $\mathsf{S}$.

## 3.2 Discussion

On rough terms, Definition 6 says the following. There exists a universal (efficient) watchdog algorithm such that for any adversary that has advantage at least $\epsilon^*$ in the subversion game (cf. Eq. (1)), the probability that the watchdog detects the subversion (cf. Eq. (2)) is at least equal to the advantage of the adversary in the subversion game divided by some positive constant $c^* \geq 1$. We observe that there could be a substantial gap between the value of $\epsilon^*$ and the actual advantage of an adversary in the subversion game. In practice, we would like to obtain Definition 6 for small $\epsilon^*, c^*$, such that either the advantage in the subversion game is smaller than $\epsilon^*$, or the advantage in the detection game has a similar magnitude as that in the subversion game (which might be much larger than $\epsilon^*$).

Looking ahead, the choice to state security of immunizers in the style of concrete security will allow us to lower bound the level of unpredictability in the subverted random source $\widetilde{\mathsf{R}}$ with a concrete (rather than asymptotic) value, a feature that will be exploited by our immunizer. One might wonder why Definition 6 considers only a single parameter $\epsilon^*$, instead of having two distinct parameters (*i.e.*, one parameter, say $\epsilon^*$, for the advantage of $\mathsf{A}$ in breaking the scheme, and another parameter, say $\delta^*$, for the advantage of $\mathsf{W}$ in detecting a subversion). While this might seem like a natural way of phrasing concrete security, it is problematic due to the fact that such a definition conveys information about a single point over the range of values $\epsilon^*, \delta^* \in [0, 1]$. A similar issue was already observed in [14], who also suggested the approach of relating the advantage in the two games.

Lastly, we stress that the security threshold $\epsilon^*$ of the immunizer $\Psi$ and the security threshold $\epsilon$ of the underlying primitive $\Pi$ (according to the original security game $\mathbf{G} = (\mathsf{C}, \gamma)$) are implicitly related in a way that depends on the actual immunizer $\Psi$. In fact, note that the optimal value of $\epsilon^*$ would be equal to $\epsilon$ as it can be seen by considering the degenerate subversion that does not modify $\Pi$ at all. Looking ahead, our immunizers will achieve sligthly sub-optimal $\epsilon^* > \epsilon$ (see Theorems 2–4 for the actual relation between $\epsilon$ and $\epsilon^*$ in the semi-private, public, and transparent models).

## 3.3 Offline versus Online Watchdogs

As pointed out in [29], different watchdog types provide different levels of security. Definition 6 considers a so-called *offline watchdog*, meaning that the watchdog is not allowed to see the messages exchanged between the challenger and the adversary during a subversion attack. On the other hand, an *online* watchdog is additionally allowed to check the transcript $\widetilde{\tau}$ containing all messages exchanged between the adversary and the challenger during an execution of a subversion attack, to make sure that such a transcript is compliant with the specification of the underlying cryptographic scheme. The main motivation behind the online watchdog model stems from the fact that, for many natural schemes $\Pi$, there are simple subversion attacks that completely break security, and yet are undetectable by an offline watchdog.

**Input-triggered attacks.** Consider for instance a (deterministic) signature scheme $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{Sign}, \mathsf{Vrfy})$ (cf. §2.2). An easy way to subvert $\Pi$ is to change the behaviour of algorithm $\mathsf{Vrfy}$, in such a way that it always returns 1 upon input a special trigger pair $(\overline{m}, \overline{\sigma})$. Such input-triggered attacks—analyzed for the first time in [14]—are easily seen to be undetectable by an offline watchdog, as the probability of hitting the hidden pair $(\overline{m}, \overline{\sigma})$ is negligible (as long as the message space is large enough). However, an online watchdog monitoring the subversion game, can easily detect such an attack by noticing that the behaviour of the verification algorithm upon input the pair $(\overline{m}, \overline{\sigma})$ is not compliant to the specification of the signature scheme.[17]

**The online watchdog model.** In order to formalize the above intuition, [29] proposes to modify the detect game as follows. In case $b = 0$ (*i.e.*, subversion takes place), the watchdog $\mathsf{W}$ is additionally provided as input the transcript $\widetilde{\tau}$ containing all messages exchanged by $\mathsf{A}$ and $\mathsf{C}$ in the subversion game. (The transcript can be appended to the auxiliary information $\mathtt{aux}$ that is passed from the subversion game to the detect game.) On the other hand, if $b = 1$ (*i.e.*, no subversion takes place), the watchdog $\mathsf{W}$ is instead given the transcript $\widehat{\tau}$ that is obtained by having $\mathsf{C}$ answering $\mathsf{A}$'s queries with oracle access to the original (non-subverted) algorithms

[17]Similar attacks are possible by subverting the signing algorithm instead of the verification algorithm; *e.g.*, we might require that $\mathsf{Sign}$ returns the secret key upon input a secret trigger message $\overline{m}$.

in $\Pi$ (see, *e.g.*, [29, Definition 4.1]). Russell *et al.* explicitly suggest that the transcripts should contain the decision bit of the challenger [29, Footnote 2].

Unfortunately, as we argue below, the online watchdog model of [29] seems to be too strong, as any subversion attack would be trivially detected by an online watchdog. In particular, we claim that for any scheme $\Pi$ that is $(t, \epsilon)$-secure w.r.t. a cryptographic game $\mathbf{G}$, the "identity immunizer" $\Psi_{\mathrm{id}}$ (*i.e.*, the immunizer that returns an identical copy of $\Pi$) is already, *e.g.*, $(\mathsf{poly}(\lambda), O(1), 1, 2\epsilon)$-subversion-secure in the online watchdog model.[18] To see this, consider the universal watchdog $\mathsf{W}$ that given the transcript $\tau \in \{\widetilde{\tau}, \widehat{\tau}\}$ outputs the decision bit $d$ contained in the transcript. Let now $\mathsf{A}$ be an adversary with advantage at least $2\epsilon$ in the subversion game. This implies that, with probability greater than $2\epsilon$, the decision bit contained in the transcript $\widetilde{\tau}$ (given to $\mathsf{W}$ when $b = 0$) will be 1, and thus

$$\mathbb{P}\left[\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi_{\mathrm{id}},\mathsf{W}}(\lambda, \mathtt{aux}, 0) = 1\right] > 2\epsilon.$$

On the other hand, the decision bit contained in $\widehat{\tau}$ (given to $\mathsf{W}$ when $b = 1$) must equal zero with probability at least $1 - \epsilon$, else we could break the security of the original scheme $\Pi$ with advantage $\epsilon$ by simply using $\mathsf{A}$'s queries in the original game $\mathbf{G}$. (Recall that the only difference between the original game and the subversion game is in the oracles used by the challenger $\mathsf{C}$ to answer $\mathsf{A}$'s queries.) Hence,

$$\mathbb{P}\left[\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi_{\mathrm{id}},\mathsf{W}}(\lambda, \mathtt{aux}, 1) = 1\right] \leq \epsilon.$$

Combining the above, we obtain a detect advantage of at least $2\epsilon - \epsilon = \epsilon$, which implies subversion-security of $\Pi^* = \Psi_{\mathrm{id}}(\Pi) = \Pi$ by setting $c^* = 2$. Note that this argument is independent of the type of game considered (*i.e.*, unpredictability or indistinguishability) and of the subversion model being semi-private, public, or transparent (as the public source $\mathsf{S}$ plays no role for the identity immunizer). Remarkably, the above discussion applies also to the definition considered in [29], *e.g.*, for the case of (possibly randomized) signatures and symmetric encryption schemes. Unfortunately, this is not consistent with known impossibility results showing that, when the randomness is large enough, such primitives are susceptible to subversion attacks that cannot be detected even by an online watchdog [9, 3, 8].

**The right definition.** A natural question is whether there is a different formulation that still allows to rule out input-triggered attacks, while at the same time giving less power to the online watchdog and, when opportunely extended to the case of randomized primitives, being consistent with known impossibility results. We propose such a variant below.[19]

Briefly, we require that the watchdog $\mathsf{W}$ be always given the same transcript during an execution of the detect game (both when $b = 0$ and $b = 1$). In other words, we remove the transcript $\widehat{\tau}$ and always give the watchdog the transcript $\widetilde{\tau}$ generated during the execution of the subversion game with the adversary $\mathsf{A}$.[20] We refer the reader to Fig. 2 for the formal specification of the detect game with online watchdogs; the definition of subversion security in the online watchdog model is then identical to Definition 6, except that we replace $\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, b)$ with $\mathbf{G}^{\text{det-on}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, b)$ in Eq.(2).

[18]A similar statement holds for the asymptotic setting, and even in the random oracle model. Put differently, any scheme that is secure is also secure under subversion in the online watchdog model of [29].

[19]In retrospect, this is probably also the definition envisioned by [29]; in fact, in proving their positive result in the online watchdog model, the watchdog does not need the transcript $\widehat{\tau}$.

[20]As for the case of offline watchdogs, in the public and transparent models the online watchdog needs to additionally take as input the actual seed sampled in the subversion game (cf. Remark 3); in fact, the seed is sampled before the interaction with the challenger starts, and thus is not part of the game transcript.

Now, it is easy to see that there are schemes $\Pi$ that are secure w.r.t. some game $\mathbf{G}$, but for which the "identity immunizer" is not subversion secure in our version of the online watchdog model. Let $\Pi$ be a deterministic signature scheme and consider a contrived modification of the scheme where the random source outputs a long random string $r = r_1||r_2$, so that the key generation algorithm, given $r_1||r_2$, outputs $r_2$ on the outside, and uses $r_1$ in order to generate a pair of keys $(pk, sk)$ for the underlying scheme $\Pi$. Let $\mathbf{G}$ be the usual UF-CMA security game for signatures (cf. §2.2). Consider the subversion that replaces the random source with the algorithm $\widetilde{\mathsf{R}}$ that hard-wires a key $\kappa$ for a pseudo-random function $\mathsf{PRF}$, and a key $\kappa'$ for a CPA-secure symmetric encryption scheme with encryption algorithm $\mathsf{Enc}$; hence, upon each invocation, $\widetilde{\mathsf{R}}$ samples random coins $r$, evaluates the PRF by computing $\widetilde{r}_1 := \mathsf{PRF}(\kappa, r)$ and encrypts $r$ by computing $\widetilde{r}_2 \xleftarrow{\$} \mathsf{Enc}(\kappa', r)$. The output of $\widetilde{\mathsf{R}}$ is defined as $\widetilde{r} := \widetilde{r}_1||\widetilde{r}_2$, while all other algorithms are left unchanged. It is easy to see that the adversary always wins the UF-CMA game when $\Pi$ is subverted, as it can simply use $\kappa, \kappa'$ in order to decrypt $\widetilde{r}_2$ (that is included as part of $\widetilde{pk}$), and thus obtain the value $r$ that allows to recover the random coins $\widetilde{r}_1$ that the challenger uses in order to generate the public/secret key; on the other hand, a standard reduction to the security of the PRF and the CPA security of the symmetric encryption scheme shows that no watchdog, even an online one, can detect the subversion with non-negligible probability. The intuition is that the online watchdog, even in possession of the transcript $\widetilde{\tau}$, will not be able to distinguish $\widetilde{r}$ from a random value (except with negligible probability), which renders the detection of the subversion practically impossible.

## 4 Semi-Private Immunizer

In this section we give a construction of a *semi-private* immunizer. The construction is described in §4.2 and is based on strong computational randomness extractors, which we define in §4.1. Finally, in §4.3, we prove security.

### 4.1 Ingredients: Strong Computational Extractors

An extractor takes as input values from a min-entropy source and outputs values that are indistinguishable from random, by making use of a short uniformly random seed. Whenever indistinguishability holds for all computationally bounded distinguishers, we speak of computational extractors. For cryptographic purposes it is often important that the seed can be made public, *i.e.* indistinguishability should hold even given the seed in the clear; such extractors are usually called *strong* extractors.

Below we give a formal definition, where we model extractors as a family of (efficiently computable) hash functions, indexed by a seed.

**Definition 7** (Computational extractors)**.** A family of hash functions $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^\ell}$ is a family of $(n, m, k, t, \epsilon)$-computational strong extractors if for all random variables $X \in \{0,1\}^n$ with min-entropy $k$, and for all distinguishers $\mathsf{D}$ running in time $t$, it holds that

$$\Delta_{\mathsf{D}}((S, h_S(X)); U) \leq \epsilon,$$

where $S$ and $U$ are uniform, respectively, over $\{0,1\}^\ell$ and $\{0,1\}^{\ell+m}$.

In case the above definition holds for the statistical (instead of computational) distance, or equivalently for all unbounded distinguishers with $t = \infty$, we call $\mathcal{H}$ a family of $(n, m, k, \epsilon)$-statistical strong extractors. Guruswami *et al.* [24] gave a construction of strong statistical extractor with near optimal parameters.

**Subversion-Resistant Immunizer $\Psi^{\mathsf{R},\mathsf{S}}_{k,k'}[\mathcal{H}]$:**

Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a cryptographic scheme, $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^\ell}$ be a family of hash functions, and $\mathsf{R}, \mathsf{S}$ be random sources over $m$-bit and $m'$-bit strings (respectively). For parameters $k := n/m$ and $k' := \ell/m'$, consider the immunizer $\Psi^{\mathsf{R},\mathsf{S}}_{k,k'}(\Pi) = \Pi^* := (\mathsf{P}^*, \mathsf{K}^*, \mathsf{R}, \mathsf{F}^*_1, \ldots, \mathsf{F}^*_N)$ specified as follows.

- Algorithm $\mathsf{P}^*$: Upon input $(1^\lambda, s_1, r_1)$, return $\rho = \mathsf{P}(1^\lambda; h_{s_1}(r_1))$.
- Algorithm $\mathsf{R}^*$: Upon input $1^\lambda$, return $r$ such that $r \overset{\$}{\leftarrow} \mathsf{R}(1^\lambda)$.
- Algorithm $\mathsf{K}^*$: Upon input $(1^\lambda, s_2, \rho, r_2)$, return $(pk, sk) = \mathsf{K}(1^\lambda, \rho; h_{s_2}(r_2))$.
- Algorithm $\mathsf{F}^*_i$ (for $i \in [N]$): Upon input $(1^\lambda, \rho, (pk, sk), x)$, return $y = \mathsf{F}_i(1^\lambda, \rho, (pk, sk), x)$.

Figure 3: Description of our subversion-resistant immunizer; the seed $s = (s_1, s_2) \in \{0,1\}^{2\ell}$ is generated by concatenating $2k' = 2\ell/m'$ independent samples from the public source $\mathsf{S}(1^\lambda)$. Similarly, the random tapes $r_1, r_2$ are obtained by concatenating $2k = 2n/m$ independent samples from the secret source $\mathsf{R}(1^\lambda)$.

**Lemma 1** ([24, Theorem 1.5])**.** *For every constant $c > 0$, all $\epsilon_{\text{ext}} > 0$, and all positive integers $n$, $k$, there is a polynomial-time computable strong $(n, m, k, \epsilon_{\text{ext}})$-statistical extractor with $\ell = O(\log(n) + \log(1/\epsilon_{\text{ext}}))$ and $m \geq (1 - c)k$.*

## 4.2 Immunizer Description

We refer the reader to Fig. 3 for a formal description of our immunizer, where we assumed that $\mathcal{R} := \{0,1\}^m$. Essentially, the immunizer sanitizes the random coins used to generate the public parameters $\rho$ and the public/secret keys $(pk, sk)$ by first sampling $(r^1_i, r^2_i)_{i\in[k]} \overset{\$}{\leftarrow} \mathsf{R}(1^\lambda)$ and amalgamating $r_1 = r^1_1||\cdots||r^1_k$ and $r_2 = r^2_1||\cdots||r^2_k$, and then using, respectively, $h_{s_1}(r_1)$ and $h_{s_2}(r_2)$ as random coins for $\mathsf{P}$ and $\mathsf{K}$, where seeds $s_1 = s^1_1||\cdots||s^1_{k'} \in \{0,1\}^\ell$ and $s_2 = s^2_1||\cdots||s^2_{k'} \in \{0,1\}^\ell$ are generated by concatenating $2k'$ independent samples from the public source $\mathsf{S}$ (*i.e.*, $(s^1_i, s^2_i)_{i\in[k']} \overset{\$}{\leftarrow} \mathsf{S}(1^\lambda)$). All other algorithms are unchanged.[21]

## 4.3 Security Analysis (Semi-Private Model)

We now analyze the security of the immunizer from Fig. 3. Intuitively, since in the semi-private model the public source $\mathsf{S}$ is assumed to be untamperable, we can focus on the special case where $m' = \ell$ (*i.e.*, the output size of the public source is the same as the size of the seed for the hash family), and thus $k' = 1$ (see also Remark 2).

We distinguish between input-constrained and input-unconstrained games. In the former case, we establish the following result; an analogous statement holds for input-unconstrained games in the online watchdog model (cf. §7.1).

**Theorem 2.** *Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a deterministic cryptographic scheme with $\mathcal{R} = \{0,1\}^m$, and $\mathsf{S}$ be uniform over $\{0,1\}^\ell$. Consider any* input-constrained, single-instance *game $\mathbf{G} = (\mathsf{C}, \gamma)$ for $\Pi$. Then, for any $c^* > 4$, the immunizer $\Psi^{\mathsf{R},\mathsf{S}}_{k,1}[\mathcal{H}]$ of Fig. 3 is $(t_\mathsf{A}, t_\mathsf{W}, c^*, \epsilon^*)$-subversion-resistant for the* spriv*-model with an* offline *watchdog, as long as $\mathcal{H} := \{h_s : \{0,1\}^n \to$*

[21] With a slight abuse of notation, we denote with $\ell$ the seed length for the hash family, whereas the final seed for the immunizer is $s := (s_1, s_2)$ of length $2\ell$ bits.

$\{0,1\}^m\}_{s\in\{0,1\}^\ell}$ *is a family of strong* $(n, m, k, t_{\mathrm{ext}}, \epsilon_{\mathrm{ext}})$*-computational extractors and* $\Pi$ *is* $(t, \epsilon)$*-secure w.r.t. game* $\mathbf{G}$ *for parameters* $t_{\mathrm{ext}}, t_{\mathsf{W}}, t \approx t_{\mathsf{A}}$*, and*

$$\epsilon \leq \frac{c^* - 1}{c^*} \cdot \epsilon^* - 2\epsilon_{\mathrm{ext}}.$$

*Proof.* By contradiction, assume that the immunizer is not subversion resistant, *i.e.* there exists an adversary $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ with running time $t_{\mathsf{A}}$ such that for all watchdogs $\mathsf{W}$ with running time $t_{\mathsf{W}}$ we have

$$\mathbf{Adv}^{\mathrm{spriv}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) = \left|\mathbb{P}\left[\mathbf{G}^{\mathrm{spriv}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) = 1\right] - \gamma\right| > \epsilon^* \tag{3}$$

$$\mathbf{Adv}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda) < \frac{1}{c^*} \cdot \mathbf{Adv}^{\mathrm{spriv}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) := \delta^*. \tag{4}$$

We introduce a sequence of intermediate hybrid experiments that are modified in an incremental manner. The games are informally described below; we refer the reader to Fig. 4 for a description in pseudocode.

```
$\mathbf{G}_1(\lambda)$
$(\widetilde{\Pi}, \alpha) \xleftarrow{\$} \mathsf{A}_0(1^\lambda, \langle \Pi, \Psi \rangle)$
$s_1, s_2 \xleftarrow{\$} \mathsf{S}(1^\lambda)$
$r^1_1, \ldots, r^1_k, r^2_1, \ldots, r^2_k \xleftarrow{\$} \widetilde{\mathsf{R}}(1^\lambda)$
$r_1 = r^1_1 || \ldots || r^1_k; r_2 = r^2_1 || \ldots || r^2_k$
$d \xleftarrow{\$} \langle \mathsf{A}_1(1^\lambda, \alpha, s_1, s_2), \mathsf{C}^{\widetilde{\mathsf{P}}(1^\lambda, s_1, \cdot), \widetilde{\mathsf{K}}(1^\lambda, s_2, \cdot, \cdot), \widetilde{\mathsf{F}}_1, \ldots, \widetilde{\mathsf{F}}_N}(1^\lambda, r_1, r_2) \rangle$
return $d$

$\mathbf{G}_2(\lambda)$
$(\widetilde{\Pi}, \alpha) \xleftarrow{\$} \mathsf{A}_0(1^\lambda, \langle \Pi, \Psi \rangle)$
$s_1, s_2 \xleftarrow{\$} \mathsf{S}(1^\lambda)$
$r^1_1, \ldots, r^1_k, r^2_1, \ldots, r^2_k \xleftarrow{\$} \widetilde{\mathsf{R}}(1^\lambda)$
$r_1 = r^1_1 || \ldots || r^1_k; r_2 = r^2_1 || \ldots || r^2_k$
$d \xleftarrow{\$} \langle \mathsf{A}_1(1^\lambda, \alpha, s_1, s_2), \mathsf{C}^{\mathsf{P}^*(1^\lambda, s_1, \cdot), \mathsf{K}^*(1^\lambda, s_2, \cdot, \cdot), \mathsf{F}_1, \ldots, \mathsf{F}_N}(1^\lambda, r_1, r_2) \rangle$
return $d$

$\mathbf{G}_3(\lambda)$
$(\widetilde{\Pi}, \alpha) \xleftarrow{\$} \mathsf{A}_0(1^\lambda, \langle \Pi, \Psi \rangle)$
$s_1, s_2 \xleftarrow{\$} \mathsf{S}(1^\lambda)$
$r_1, r_2 \xleftarrow{\$} \{0,1\}^m;$
$d \xleftarrow{\$} \langle \mathsf{A}_1(1^\lambda, \alpha, s_1, s_2), \mathsf{C}^{\mathsf{P}, \mathsf{K}, \mathsf{F}_1, \ldots, \mathsf{F}_N}(1^\lambda, r_1, r_2) \rangle$
return $d$
```

Figure 4: Game hops in the proof of Theorem 2.

**$\mathbf{G}_1(\lambda)$:** This is identical to the game $\mathbf{G}^{\text{spriv}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda)$, defining subversion security of an immunizer (cf. Fig. 1). Here, the adversary $\mathsf{A}$ specifies the modified implementation $\widetilde{\Pi}$ of the immunized scheme $\Pi^*$ before receiving the seeds $s_1, s_2 \in \{0,1\}^\ell$. (Recall that $s_1, s_2$ are generated by sampling $\mathsf{S}$ only once, since $\mathsf{S}$ is defined over $\{0,1\}^\ell$, *i.e.* $m' = \ell$.) Hence, the interaction between $\mathsf{A}$ and $\mathsf{C}$ goes like in the original game $\mathbf{G}$, except that $\mathsf{C}$ is given oracle access to $\widetilde{\mathsf{P}}(1^\lambda, s_1, \cdot)$, $\widetilde{\mathsf{K}}(1^\lambda, s_2, \cdot)$, and $(\widetilde{\mathsf{F}}_i)_{i\in N}$, where $\widetilde{\mathsf{P}}$ and $\widetilde{\mathsf{K}}$ take as random coins, respectively, $r_1 = r_1^1||\dots,||r_k^1$ and $r_2 = r_1^2||\dots,||r_k^2$, for $r_i^j \xleftarrow{\$} \widetilde{\mathsf{R}}(1^\lambda)$ and $(i,j) \in [k]\times[2]$.

Eq. (4)

**$\mathbf{G}_2(\lambda)$:** Identical to the previous game, except that we replace the adversarial implementation of the algorithm for generating the public parameters (*i.e.*, algorithm $\widetilde{\mathsf{P}}$), for key generation (*i.e.*, algorithm $\widetilde{\mathsf{K}}$), and each $\widetilde{\mathsf{F}}_i$ (for $i \in [N]$) with the corresponding immunized algorithm, *i.e.*, respectively, $\mathsf{P}^*(1^\lambda, s_1; r_1) := \mathsf{P}(1^\lambda; h_{s_1}(r_1))$, $\mathsf{K}^*(1^\lambda, s_2, \rho; r_2) := \mathsf{K}(1^\lambda, \rho; h_{s_2}(r_2))$, and $\mathsf{F}^*_i(1^\lambda, \rho, (pk, sk), x) := \mathsf{F}_i(1^\lambda, \rho, (pk, sk), x)$ where $r_1, r_2 \in \{0,1\}^n$ are as before.

Rand. extr.

**$\mathbf{G}_3(\lambda)$:** Identical to the previous game, except that we generate the strings $r_1, r_2$ differently. Namely, we now run the challenger upon input $r_1, r_2$ uniformly sampled from $\{0,1\}^m$. Additionally, the public parameters and the keys are now generated using $\mathsf{P}(1^\lambda; r_1)$ and $\mathsf{K}(1^\lambda; r_2)$, where $r_1, r_2 \xleftarrow{\$} \{0,1\}^m$.

We proceed to show that the above games are all computationally indistinguishable, and thus $\mathsf{A}$'s advantage does not vanish across the different games.

**Lemma 2.** $|\,\mathbb{P}[\mathbf{G}_1(\lambda) = 1] - \mathbb{P}[\mathbf{G}_2(\lambda) = 1]\,| \leq \delta^*$.

*Proof.* Consider the following events, which are defined over the probability space of game $\mathbf{G}_1$.

**Event $E_{\text{pub}}$:** The event becomes true whenever $\widetilde{\mathsf{P}}(1^\lambda, s_1; r_1) \neq \mathsf{P}^*(1^\lambda, s_1; r_1)$, where $r_1 = r_1^1||\dots||r_k^1$, for $r_i^1 \xleftarrow{\$} \widetilde{\mathsf{R}}(1^\lambda)$, and $s_1 \xleftarrow{\$} \mathsf{S}(1^\lambda)$.

**Event $E_{\text{kgen}}$:** The event becomes true whenever $\widetilde{\mathsf{K}}(1^\lambda, s_2, \rho; r_2) \neq \mathsf{K}^*(1^\lambda, s_2, \rho; r_2)$, where $\rho = \widetilde{\mathsf{P}}(1^\lambda, s_1; r_1)$ for $r_1$ as in the previous event, $r_2 = r_1^2||\dots||r_k^2$ for $r_i^2 \xleftarrow{\$} \widetilde{\mathsf{R}}(1^\lambda)$, and $s_1, s_2 \xleftarrow{\$} \mathsf{S}(1^\lambda)$.

**Event $E_{\text{func}}$:** The event becomes true whenever $\exists i \in [N], j \in [q], x_j \in \mathcal{X}_i$ such that $\widetilde{\mathsf{F}}_i(1^\lambda, \rho, (pk, sk), x_j) \neq \mathsf{F}^*_i(1^\lambda, \rho, (pk, sk), x_j)$, where $\rho = \widetilde{\mathsf{P}}(1^\lambda, s_1; r_1)$, $(pk, sk) = \widetilde{\mathsf{K}}(1^\lambda, s_2, \rho; r_2)$, the coins $r_1, r_2$ are as in the previous event, $s_1, s_2 \xleftarrow{\$} \mathsf{S}(1^\lambda)$, $x_j \xleftarrow{\$} D_i$ (recall that the game is input constrained), and $q \in \mathsf{poly}(\lambda)$ is an upper bound on the total number of queries asked by the adversary to each of the oracles $\widetilde{\mathsf{F}}_1, \dots, \widetilde{\mathsf{F}}_N$.

Define $E_{\text{bad}} := E_{\text{pub}} \vee E_{\text{kgen}} \vee E_{\text{func}}$. Clearly, if $E_{\text{bad}}$ does not happen, we have that $\mathbf{G}_1(\lambda)$ and $\mathbf{G}_2(\lambda)$ are identically distributed, and thus it suffices to prove that event $E_{\text{bad}}$ only happens with probability smaller than $\delta^*$. By contradiction, assume that $E_{\text{bad}}$ is provoked in the game execution with probability at least $\delta^*$. We construct an efficient offline watchdog $\mathsf{W}^*$ that distinguishes between $\mathbf{G}^{\text{det}}_{\Pi,\Psi,\mathsf{W}^*}(\lambda, \mathsf{aux}, 0)$ and $\mathbf{G}^{\text{det}}_{\Pi,\Psi,\mathsf{W}^*}(\lambda, \mathsf{aux}, 1)$ with the same probability, which contradicts Eq. (4). A description of $\mathsf{W}^*$ follows.

1. Let us denote by $\mathsf{O}_{\text{pub}}$, $\mathsf{O}_{\text{rand}}$, $\mathsf{O}_{\text{kgen}}$, and $\mathsf{O}^i_{\text{func}}$ (for $i \in [N]$) the target oracles which $\mathsf{W}^*$ can access, corresponding to the subverted/immunized public parameters generator, private randomness generator, key generator, and each of the underlying functional algorithms. Note that:

$$(\mathsf{O}_{\text{pub}}, \mathsf{O}_{\text{rand}}, \mathsf{O}_{\text{kgen}}, (\mathsf{O}^i_{\text{func}})_{i\in[N]}) \equiv \begin{cases} (\widetilde{\mathsf{P}}, \widetilde{\mathsf{R}}, \widetilde{\mathsf{K}}, (\widetilde{\mathsf{F}}_i)_{i\in[N]}) & \text{for } b = 0 \\ (\mathsf{P}^*, \mathsf{R}, \mathsf{K}^*, (\mathsf{F}^*_i)_{i\in[N]}) & \text{for } b = 1 \end{cases}$$

where $b$ is the hidden bit in the detection game, and $\Pi^* := (\mathsf{P}^*, \mathsf{R}, \mathsf{K}^*, (\mathsf{F}_i^*)_{i\in[N]}) = \Psi(\Pi)$ and $\widetilde{\Pi} := (\widetilde{\mathsf{P}}, \widetilde{\mathsf{R}}, \widetilde{\mathsf{K}}, (\widetilde{\mathsf{F}}_i)_{i\in[N]})$ are, respectively, the specification of the immunized cryptosystem and of the subversion output by the adversary in the subversion game.

2. Sample $s_1, s_2 \overset{\$}{\leftarrow} \mathsf{S}(1^\lambda)$, and set $r_1 = r_1^1||\ldots||r_k^1$, $r_2 = r_1^2||\ldots||r_k^2$, where $r_i^j \overset{\$}{\leftarrow} \mathsf{O}_{\mathrm{rand}}(1^\lambda)$ for each $(i,j) \in [k] \times [2]$.
3. Invoke $\mathsf{O}_{\mathrm{pub}}(1^\lambda, \cdot, \cdot)$ upon input $(1^\lambda, s_1, r_1)$, obtaining a value $\rho$; in case $\rho$ does not equal $\mathsf{P}^*(1^\lambda, s_1, r_1) = \mathsf{P}(1^\lambda, h_{s_1}(r_1))$ output 1 and stop.
4. Invoke $\mathsf{O}_{\mathrm{kgen}}$ upon input $(1^\lambda, s_2, \rho, r_2)$, obtaining a pair of keys $(pk, sk)$; in case $(pk, sk)$ does not equal $\mathsf{K}^*(1^\lambda, s_2, \rho, r_2) = \mathsf{K}(1^\lambda, \rho, h_{s_2}(r_2))$ output 1 and stop.
5. For each $i \in [N]$ and $j \in [q]$, invoke $\mathsf{O}_{\mathrm{func}}^i$ upon input $(1^\lambda, \rho, (pk, sk), x_j)$, where $x_j \overset{\$}{\leftarrow} D_i$, obtaining some output $y_j$; in case there exists $i \in [N]$ and $j \in [q]$ such that the value $y_j$ does not equal $\mathsf{F}_i^*(1^\lambda, \rho, (pk, sk), x_j) = \mathsf{F}_i(1^\lambda, \rho, (pk, sk), x_j)$ output 1 and stop.
6. Output 0.

Recall that the watchdog $\mathsf{W}^*$ is always given the original specification of $\langle\Psi, \Pi\rangle$, and thus it can indeed perform the comparisons described in steps 3–5; additionally the inputs $x_j$ can be sampled by the watchdog in an offline fashion (*i.e.*, without talking to the adversary), since the game is input constrained. By definition of the event $E_{\mathrm{bad}}$, the probability of the watchdog outputting 1 when $b = 0$ is at least the probability of $E_{\mathrm{bad}}$; on the other hand, when $b = 1$, it is easy to see that the watchdog never outputs 1. Thus:

$$\mathbb{P}\left[\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}^*}(\lambda, \mathtt{aux}, 1) = 1\right] \geq \mathbb{P}[E_{\mathrm{bad}}]$$
$$\mathbb{P}\left[\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}^*}(\lambda, \mathtt{aux}, 0) = 1\right] = 0$$

and hence we have obtained

$$\mathbf{Adv}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}^*}(\lambda) = \left|\mathbb{P}\left[\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}^*}(\lambda, \mathtt{aux}, 0) = 1\right] - \mathbb{P}\left[\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}^*}(\lambda, \mathtt{aux}, 1) = 1\right]\right| \geq \delta^*,$$

which concludes the proof of the lemma. □

**Lemma 3.** $|\,\mathbb{P}[\mathbf{G}_2(\lambda) = 1] - \mathbb{P}[\mathbf{G}_3(\lambda) = 1]\,| \leq 2\epsilon_{\mathrm{ext}}$.

*Proof.* Let $\widetilde{R}$ be the random variable associated to the subverted source $\widetilde{\mathsf{R}}$. We start by showing that the min-entropy of $\widetilde{R}$ is at least 1 bit. By contradiction, assume that $\mathbb{H}_\infty(\widetilde{R}) < 1$. Consider the watchdog $\mathsf{W}_{\mathrm{rand}}$ that samples $r, r' \overset{\$}{\leftarrow} \mathsf{O}_{\mathrm{rand}}(1^\lambda)$, and outputs 1 if and only if $r = r'$; here, $\mathsf{O}_{\mathrm{rand}} \in \{\mathsf{R}, \widetilde{\mathsf{R}}\}$ is the target oracle corresponding to the subverted/immunized randomness generator in the detection game. Hence, we can write

$$\begin{aligned}\mathbf{Adv}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}_{\mathrm{rand}}}(\lambda) &= \left|\mathbb{P}\left[\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}_{\mathrm{rand}}}(\lambda, \mathtt{aux}, 0)\right] - \mathbb{P}\left[\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}_{\mathrm{rand}}}(\lambda, \mathtt{aux}, 1)\right]\right| \\ &= 2^{-\mathbb{H}_2(\widetilde{R})} - 2^{-\mathbb{H}_2(R)} \qquad (5) \\ &> 2^{-2} - 2^{-n} \qquad (6) \\ &> 2^{-2k''} - 2^{-n} = \frac{1}{c^*}, \qquad (7)\end{aligned}$$

where $k'' := -1/2 \log(1/c^* + 2^{-n})$. In the above derivation, Eq. (5) follows by definition of collision entropy, Eq. (6) follows because $R$ is uniform and moreover $\mathbb{H}_2(\widetilde{R}) \leq 2\mathbb{H}_\infty(\widetilde{R}) < 2$, and

Eq. (7) is due to the fact that $k' > 1$ whenever $n, c^* \geq 5$ (as in the statement of the theorem). Note that Eq. (7) contradicts Eq. (4), since the latter implies $\mathbf{Adv}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}_{\mathrm{rand}}}(\lambda) < \frac{1}{c^*}$.

Next, we use the fact that $\mathbb{H}_\infty(\widetilde{R}) \geq 1$ in order to show that game $\mathbf{G}_2(\lambda)$ and $\mathbf{G}_3(\lambda)$ are computationally indistinguishable. To this end, consider the intermediate experiment $\mathbf{H}(\lambda)$ that is identical to $\mathbf{G}_2(\lambda)$, except that the challenger now generates $\rho$ by running $\mathsf{P}$ on uniformly random coins $r_1 \overset{\$}{\leftarrow} \{0,1\}^m$. We first show that $\mathbf{G}_2(\lambda)$ is computationally indistinguishable from $\mathbf{H}(\lambda)$. By contradiction, assume that there is a distinguisher $\mathsf{D}$ with running time $t_{\mathsf{D}} = t_{\mathsf{A}}$ such that $\Delta_{\mathsf{D}}(\mathbf{G}_2(\lambda); \mathbf{H}(\lambda)) > \epsilon_{\mathrm{ext}}$. We construct a distinguisher $\mathsf{D}'$ breaking security of the hash family $\mathcal{H}$. Distinguisher $\mathsf{D}'$ is given as input a pair $(z_1, z_2)$ and has to determine whether $z_2 = h_{z_1}(r)$ (for uniform $z_1 \overset{\$}{\leftarrow} \{0,1\}^\ell$ and $r \overset{\$}{\leftarrow} \widetilde{R}^k$, where $\widetilde{R}^k$ is the concatenation of $k$ identical copies of $\widetilde{\mathsf{R}}$) or $z_1 || z_2 \overset{\$}{\leftarrow} \{0,1\}^{\ell+m}$.

Hence, $\mathsf{D}'$ proceeds as follows:

1. Receive (a description of) $\widetilde{\Pi} = (\widetilde{\mathsf{P}}, \widetilde{\mathsf{K}}, \widetilde{\mathsf{R}}, \widetilde{\mathsf{F}}_1, \ldots, \widetilde{\mathsf{F}}_N)$.
2. Sample $s_2 \overset{\$}{\leftarrow} \{0,1\}^\ell$ and forward $z_1 || s_2$ to $\mathsf{D}$.
3. Emulate the challenger by answering $\mathsf{D}$'s queries exactly as described in game $\mathbf{G}_2(\lambda)$, except that the public parameters are set by default to be equal to $\rho := \mathsf{P}(1^\lambda; z_2)$.
4. Output the same guess as that of $\mathsf{D}$.

Observe that $\mathsf{D}'$ runs in time $t_{\mathrm{ext}} \approx t_{\mathsf{A}}$, and moreover, as we argue below, the simulation is perfect. In fact, the only difference between the two games is in how the public parameters are set by the challenger:

- In case $(z_1, z_2)$ are uniformly random, *i.e.* $(z_1, z_2) \overset{\$}{\leftarrow} \{0,1\}^{\ell+m}$, the public parameters are computed as $\rho := \mathsf{P}(1^\lambda; r)$ for uniform $z_2 = r$; this is exactly what happens in game $\mathbf{H}(\lambda)$.
- In case $(z_1, z_2)$ are defined through the extractor, *i.e.* $(z_1, z_2)$ are such that $z_2 = h_{z_1}(r)$ for $r \overset{\$}{\leftarrow} \widetilde{\mathsf{R}}^k$, the public parameters are computed as $\rho := \mathsf{P}(1^\lambda; h_{z_1}(r))$; this is exactly what happens in game $\mathbf{G}_2(\lambda)$.

Finally, we claim that the random variable $\widetilde{R}^k$ has $k$ bits of min-entropy. The latter can be seen as follows:

$$2^{-\mathbb{H}_\infty(\widetilde{R}^k)} = \max_{r_1,\ldots,r_k} \mathbb{P}\left[\widetilde{R}_1 = r_1, \ldots, \widetilde{R}_k = r_k\right] = \max_{r_1,\ldots,r_k} \prod_{i=1}^{k} \mathbb{P}\left[\widetilde{R}_i = r_i\right] \leq 2^{-k}, \tag{8}$$

where the sources $(\widetilde{R}_i)_{i\in[k]}$ are identical copies of $\widetilde{R}$, and where we used the fact that $\widetilde{R}$ is efficiently samplable with $\mathbb{H}_\infty(\widetilde{R}) \geq 1$, and furthermore can wlog. be assumed to be stateless [29, Remark 2.5].[22]

Hence, $\mathsf{D}'$ retains the same advantage as that of $\mathsf{D}$, which contradicts security of the family of extractors $\mathcal{H}$. An analogous[23] argument shows that game $\mathbf{H}(\lambda)$ and $\mathbf{G}_3(\lambda)$ are within distance $\epsilon_{\mathrm{ext}}$, which concludes the proof. □

[22]Recall that the watchdog is given rewinding black-box access to its oracles.

[23]The only difference is that $\mathsf{D}'$ forwards $(s_1, z_1)$ to $\mathsf{D}$, where $s_1 \overset{\$}{\leftarrow} \{0,1\}^\ell$, and moreover it defines $(pk, sk) := \mathsf{K}(1^\lambda, \rho; z_2)$, with $\rho = \mathsf{P}(1^\lambda; r)$ for uniform $r \overset{\$}{\leftarrow} \{0,1\}^m$.

Finally, note that game $\mathbf{G}_3(\lambda)$ is identically distributed to the security game $\mathbf{G}$. Hence, by combining the above lemmas with Eq. (3), and by plugging the expression for the detect advantage $\delta^*$, we obtain

$$\epsilon \geq |\mathbb{P}[\mathbf{G}_3(\lambda) = 1] - \gamma| \geq \mathbf{Adv}^{\mathrm{spriv}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) - \delta^* - 2\epsilon_{\mathrm{ext}} > \frac{c^* - 1}{c^*} \cdot \epsilon^* - 2\epsilon_{\mathrm{ext}},$$

which contradicts the condition on $\epsilon$ in the theorem statement. This finishes the proof. □

**Remark 4** (On single-instance games)**.** The fact that our immunizer samples $2k$ times from the source $\mathsf{R}$ does *not* contradict the assumption that $\mathbf{G}$ is a single-instance game, as the latter condition only concerns the game $\mathbf{G}$ for the original primitive $\Pi$.

**Instantiating the immunizer.** Using Lemma 1, we proceed to a corollary that shows the parameters our semi-private immunizer achieves.

**Corollary 1.** *Let* $\mathsf{R}$ *and* $\mathsf{S}$ *be, respectively, uniformly random* $m$*-bit and* $m'$*-bit sources. For any cryptographic primitive that is* $(\mathsf{poly}(\lambda), \mathsf{negl}(\lambda))$*-secure w.r.t. input-constrained game* $\mathbf{G}$*, there exists an immunizer* $\Psi^{\mathsf{R},\mathsf{S}}_{k,1}$ *for* $\Pi$ *that is* $(\mathsf{poly}(\lambda), \mathsf{poly}(\lambda), 5, \mathsf{negl}(\lambda))$*-subversion-resistant for the* spriv*-model with an offline watchdog, with parameters* $k, \ell, m \in \omega(\log(\lambda))$*.*

*Proof.* The statement follows by plugging any statistical extractor with $k, \ell \in \omega(\log(\lambda))$ and $\epsilon_{\mathrm{ext}} \in \mathsf{negl}(\lambda)$, as given, *e.g.*, by Lemma 1. □

**Remark 5** (On the parameter $m$)**.** Using statistical extractors, the best we can hope for is to apply the immunizer to a cryptographic primitive with randomness length $m = (1 - c)k$ for any constant $c > 0$ (as guaranteed by Lemma 1). Alternatively, we can assume polynomially-secure one-way functions and obtain $\ell \in \omega(\log(\lambda))$ and $m \in \mathsf{poly}(\lambda)$ by using the classical "extract-then-prg" approach, or more sophisticated computational extractors [13].

## 5 Public Immunizer

In this section we provide a different instantiation for the immunizer from Fig. 3, and show that it achieves security in the *public* model. The instantiation relies on seed-dependent condensers, which we define in §5.1. The security analysis appears in §5.2.

### 5.1 Ingredients: Seed-Dependent Randomness Condensers

We recall the notion of seed-dependent randomness condenser [18]. Intuitively, this corresponds to a family of hash functions indexed by an $\ell$-bit seed, and mapping $n$ bits into $m$ bits. The security guarantee is that when the seed $s$ is uniform, and the input $x$ comes from an adversarial (efficiently sampleable) source which might depend on $s$, and with min-entropy at least $k$, the output of the hash function has at least $m - d$ bits of min-entropy, for deficiency parameter $d \geq 1$.

**Definition 8** (Seed-dependent condenser)**.** Let $\mathcal{G} := \{g_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^\ell}$ be a family of efficiently computable functions. We say that $\mathcal{G}$ is a family of $(\frac{k}{n} \to \frac{m-d}{m}, t, \epsilon)$-seed-dependent condensers if for all probabilistic adversaries $\mathsf{A}$ running in time $t$ who take a seed $s \xleftarrow{\$} \{0,1\}^\ell$ and output (using more coins) a distribution $X \xleftarrow{\$} \mathsf{A}(s)$ of entropy $\widetilde{\mathbb{H}}_\infty(X|S) \geq k$, the joint distribution $(S, g_S(X))$ is $\epsilon$-close to some $(S, Y)$, where $\widetilde{\mathbb{H}}_\infty(Y|S) \geq m - d$ and $S$ is uniform over $\{0,1\}^\ell$.

As we recall below, Dodis *et al.* [18, 19] have shown that any sufficiently strong collision-resistant family of hash functions directly yields a family of seed-dependent condensers.

**Definition 9** (Collision-resistant hash function)**.** Let $\mathcal{G} := \{g_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^\ell}$ be a family of efficiently computable functions. We say that $\mathcal{G}$ is a family of $(t,\epsilon)$-collision-resistant hash functions if for all probabilistic adversaries $\mathsf{B}$ running in time $t$ that output $(x_1,x_2) \xleftarrow{\boxdot} \mathsf{B}(s)$ for $s \xleftarrow{\boxdot} \{0,1\}^\ell$, we have that $\Pr[g_s(x_1) = g_s(x_2) \wedge x_1 \neq x_2] \leq \epsilon$.

**Lemma 4** ([18])**.** *Let $\mathcal{G} := \{g_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^\ell}$ be a family of $(2t, \mathsf{poly}(\lambda)/2^m)$-collision-resistant hash functions. Then, $\mathcal{G}$ is a family of $(\frac{k}{n} \to \frac{m-d}{m}, t, 0)$-seed-dependent condensers for $d = O(\log(\lambda))$ and $k \leq n$.*

## 5.2 Security Analysis (Public Model)

The theorem below says that when we instantiate the immunizer of Fig. 3 with a family of seed-dependent condensers, we achieve subversion security in the public model. Analogously to the instantiation in the semi-private model, since the public source $\mathsf{S}$ is assumed to be untamperable, we can focus on the special case where $m' = \ell$ (*i.e.*, the output size of the public source is the same as the size of the seed for the hash family), and thus $k' = 1$ (see also Remark 2).

For input-constrained games, we obtain the following result. An analogous statement holds for input-unconstrained games, in the online watchdog model (cf. §7.2).

**Theorem 3.** *Let $\Pi = (\mathsf{P},\mathsf{K},\mathsf{R},\mathsf{F}_1,\dots,\mathsf{F}_N)$ be a deterministic cryptographic scheme with $\mathcal{R} = \{0,1\}^m$, and $\mathsf{S}$ be uniform over $\{0,1\}^\ell$. Consider any* input-constrained, single-instance *game $\mathbf{G} = (\mathsf{C},\gamma)$ for $\Pi$. Then, for any $c^* > 4$, the immunizer $\Psi^{\mathsf{R},\mathsf{S}}_{k,1}[\mathcal{G}]$ of Fig. 3 is $(t_\mathsf{A}, t_\mathsf{W}, c^*, \epsilon^*)$-subversion-resistant for the* pub*-model with an* offline *watchdog, as long as $\mathcal{G} := \{g_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^\ell}$ is a family of $(\frac{k}{n} \to \frac{m-d}{m}, t_{\text{cond}}, \epsilon_{\text{cond}})$-seed-dependent condensers and $\Pi$ is either $(t,\epsilon)$-secure w.r.t. game $\mathbf{G}$ (in case of unpredictability games) or $(t,\epsilon)$-*square-*secure w.r.t. game $\mathbf{G}$ (in case of indistinguishability games), for parameters $t_{\text{cond}}, t, t_\mathsf{W} \approx t_\mathsf{A}$, and*

$$\epsilon \leq \begin{cases} \frac{c^*-1}{c^*} \cdot \frac{\epsilon^*}{2^{2d}} - \frac{2\epsilon_{\text{cond}}}{2^{2d}} & \textit{if } \mathbf{G} \textit{ is an unpredictability game} \\ \left(\frac{c^*-1}{c^*} \cdot \epsilon^* - 2\epsilon_{\text{cond}}\right)^2 \cdot \frac{1}{2^{2d}} & \textit{if } \mathbf{G} \textit{ is an indistinguishability game.} \end{cases}$$

*Proof.* We prove Theorem 3 by using an identical proof strategy to that of Theorem 2. The main difference is that now subversion of the source $\mathsf{R}$ depends on the seed sampled from the public source $\mathsf{S}$ which justifies the need for seed-dependent randomness condensers. We include the full proof for self-containment. By contradiction, assume that the immunizer is not subversion resistant, *i.e.* there exists an adversary $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ with running time $t_\mathsf{A}$ such that for all watchdogs $\mathsf{W}$ with running time $t_\mathsf{W}$ we have

$$\mathbf{Adv}^{\text{pub}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) = \left|\mathbb{P}\left[\mathbf{G}^{\text{pub}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) = 1\right] - \gamma\right| > \epsilon^* \tag{9}$$

$$\mathbf{Adv}^{\text{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda) < \frac{1}{c^*} \cdot \mathbf{Adv}^{\text{pub}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) := \delta^*. \tag{10}$$

We introduce a sequence of intermediate hybrid experiments that are modified in an incremental manner. The games are informally described below; we refer the reader to Fig. 5 for a description in pseudocode.

**$\mathbf{G}_1(\lambda)$:** This is identical to the game $\mathbf{G}^{\text{pub}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda)$, defining subversion security of an immunizer (cf. Fig. 1). Here, the adversary $\mathsf{A}$ specifies the modified implementation $\widetilde{\Pi}$ of the immunized scheme $\Pi^*$ as a function of the seeds $s_1, s_2 \in \{0,1\}^\ell$. (Recall that $s_1, s_2$ are generated by sampling $\mathsf{S}$ only once, since $\mathsf{S}$ is defined over $\{0,1\}^\ell$, *i.e.* $m' = \ell$.) Hence, the interaction between $\mathsf{A}$ and $\mathsf{C}$ goes like in the original game $\mathbf{G}$, except that $\mathsf{C}$ is given oracle access to $\widetilde{\mathsf{P}}(1^\lambda, s_1, \cdot)$, $\widetilde{\mathsf{K}}(1^\lambda, s_2, \cdot)$, and $(\widetilde{\mathsf{F}}_i)_{i \in N}$, where $\widetilde{\mathsf{P}}$ and $\widetilde{\mathsf{K}}$ take as random coins, respectively $r_1 = r_1^1 || \dots, || r_k^1$ and $r_2 = r_1^2 || \dots, || r_k^2$, for $r_i^j \overset{\boxdot}{\leftarrow} \widetilde{\mathsf{R}}(1^\lambda)$ and $(i,j) \in [k] \times [2]$.

$\downarrow$ Eq. (10)

$\mathbf{G}_2(\lambda)$**:** Identical to the previous game, except that we replace the adversarial implementation of the algorithm for generating the public parameters (*i.e.*, algorithm $\widetilde{\mathsf{P}}$), for key generation (*i.e.*, algorithm $\widetilde{\mathsf{K}}$), and each $\widetilde{\mathsf{F}}_i$ (for $i \in [N]$) with the corresponding immunized algorithm, *i.e.*, respectively, $\mathsf{P}^*(1^\lambda, s_1; r_1) := \mathsf{P}(1^\lambda; h_{s_1}(r_1))$, $\mathsf{K}^*(1^\lambda, s_2, \rho; r_2) := \mathsf{K}(1^\lambda, \rho; h_{s_2}(r_2))$, and $\mathsf{F}_i^*(1^\lambda, \rho, (pk, sk), x) := \mathsf{F}_i(1^\lambda, \rho, (pk, sk), x)$ where $r_1, r_2 \in \{0,1\}^n$ are as before.

$\downarrow$ Rand. cond.

$\mathbf{G}_3(\lambda)$**:** Identical to the previous game, except that we generate the strings $r_1, r_2$ differently. Namely, we now give to the challenger directly $y_1, y_2$ sampled from the efficiently sampleable source $\mathsf{Y}$ guaranteed by the security of the family of condensers $\mathcal{G}$. Additionally, the public parameters and the keys are now generated using algorithms $\mathsf{P}(1^\lambda; y_1)$ and $\mathsf{K}(1^\lambda; y_2)$ where $y_1, y_2 \overset{\boxdot}{\leftarrow} \mathsf{Y}(1^\lambda)$.

$\mathbf{G}_1(\lambda)$

```
$s_1, s_2 \overset{\boxdot}{\leftarrow} \mathsf{S}(1^\lambda)$
$(\widetilde{\Pi}, \alpha) \overset{\boxdot}{\leftarrow} \mathsf{A}_0(1^\lambda, s_1, s_2, \langle \Pi, \Psi \rangle)$
$r_1^1, \dots, r_k^1, r_1^2, \dots, r_k^2 \overset{\boxdot}{\leftarrow} \widetilde{\mathsf{R}}(1^\lambda)$
$r_1 = r_1^1 || \dots || r_k^1; r_2 = r_1^2 || \dots || r_k^2$
$d \overset{\boxdot}{\leftarrow} \langle \mathsf{A}_1(1^\lambda, \alpha), \mathsf{C}^{\widetilde{\mathsf{P}}(1^\lambda, s_1, \cdot), \widetilde{\mathsf{K}}(1^\lambda, s_2, \cdot, \cdot), \widetilde{\mathsf{F}}_1, \dots, \widetilde{\mathsf{F}}_N}(1^\lambda, r_1, r_2) \rangle$
return $d$
```

$\mathbf{G}_2(\lambda)$

```
$s_1, s_2 \overset{\boxdot}{\leftarrow} \mathsf{S}(1^\lambda)$
$(\widetilde{\Pi}, \alpha) \overset{\boxdot}{\leftarrow} \mathsf{A}_0(1^\lambda, s_1, s_2, \langle \Pi, \Psi \rangle)$
$r_1^1, \dots, r_k^1, r_1^2, \dots, r_k^2 \overset{\boxdot}{\leftarrow} \widetilde{\mathsf{R}}(1^\lambda)$
$r_1 = r_1^1 || \dots || r_k^1; r_2 = r_1^2 || \dots || r_k^2$
$d \overset{\boxdot}{\leftarrow} \langle \mathsf{A}_1(1^\lambda, \alpha), \mathsf{C}^{\mathsf{P}^*(1^\lambda, s_1, \cdot), \mathsf{K}^*(1^\lambda, s_2, \cdot, \cdot), \mathsf{F}_1, \dots, \mathsf{F}_N}(1^\lambda, r_1, r_2) \rangle$
return $d$
```

$\mathbf{G}_3(\lambda)$

```
$s_1, s_2 \overset{\boxdot}{\leftarrow} \mathsf{S}(1^\lambda)$
$(\widetilde{\Pi}, \alpha) \overset{\boxdot}{\leftarrow} \mathsf{A}_0(1^\lambda, s_1, s_2, \langle \Pi, \Psi \rangle)$
$y_1, y_2 \overset{\boxdot}{\leftarrow} \mathsf{Y}(1^\lambda)$;
$d \overset{\boxdot}{\leftarrow} \langle \mathsf{A}_1(1^\lambda, \alpha), \mathsf{C}^{\mathsf{P}, \mathsf{K}, \mathsf{F}_1, \dots, \mathsf{F}_N}(1^\lambda, y_1, y_2) \rangle$
return $d$
```

Figure 5: Game hops in the proof of Theorem 3.

We proceed to show that the above games are all computationally indistinguishable, and thus $\mathsf{A}$'s advantage does not vanish across the different games.

**Lemma 5.** $|\,\mathbb{P}[\mathbf{G}_1(\lambda) = 1] - \mathbb{P}[\mathbf{G}_2(\lambda) = 1]\,| \leq \delta^*$.

*Proof.* The proof is identical to that of Lemma 2. The only difference is that the watchdog $\mathsf{W}^*$ uses the seeds $s_1, s_2$ that now are part of its own input, instead of sampling the seeds from $\mathsf{S}$, in order to check the implementations of $\mathsf{P}^*, \mathsf{K}^*$, and $(\mathsf{F}^*_i)_{i\in[N]}$. □

**Lemma 6.** $|\,\mathbb{P}[\mathbf{G}_2(\lambda) = 1] - \mathbb{P}[\mathbf{G}_3(\lambda) = 1]\,| \leq 2\epsilon_{\text{cond}}$.

*Proof.* Let $\widetilde{R}$ be the random variable associated to the subverted source $\widetilde{\mathsf{R}}$, let $S$ be the random variable corresponding to the public source $\mathsf{S}$, and let $S_1, S_2$ be identical copies of $S$. We start by showing that for all $s_1, s_2 \in \{0,1\}^\ell$ the min-entropy of $\widetilde{R}$ conditioned on $s_1, s_2$ is at least 1.

By contradiction, assume that $\mathbb{H}_\infty(\widetilde{R}|S_1 = s_1, S_2 = s_2) < 1$. Consider the watchdog $\mathsf{W}_{\text{rand}}$ with input $s_1, s_2$ that samples $r, r' \overset{\$}{\leftarrow} \mathsf{O}_{\text{rand}}(1^\lambda)$, and outputs 1 if and only if $r = r'$; here, $\mathsf{O}_{\text{rand}} \in \{\mathsf{R}, \widetilde{\mathsf{R}}\}$ is the target oracle corresponding to the subverted/immunized randomness generator in the detection game. Hence, we can write

$$\begin{aligned}
\mathbf{Adv}^{\text{det}}_{\Pi,\Psi,\mathsf{W}_{\text{rand}}}(\lambda) &= \left|\mathbb{P}\left[\mathbf{G}^{\text{det}}_{\Pi,\Psi,\mathsf{W}_{\text{rand}}}(\lambda, \mathtt{aux}, 0)\right] - \mathbb{P}\left[\mathbf{G}^{\text{det}}_{\Pi,\Psi,\mathsf{W}_{\text{rand}}}(\lambda, \mathtt{aux}, 1)\right]\right| \\
&= 2^{-\mathbb{H}_2(\widetilde{R}|S_1=s_1,S_2=s_2)} - 2^{-\mathbb{H}_2(R|S_1=s_1,S_2=s_2)} && (11)\\
&> 2^{-2} - 2^{-n} && (12)\\
&> 2^{-2k''} - 2^{-n} = \frac{1}{c^*}, && (13)
\end{aligned}$$

where $k'' := -1/2\log(1/c^* + 2^{-n})$. In the above derivation, Eq. (11) follows by definition of conditional collision entropy, Eq. (12) follows because $R$ is uniform and moreover $\mathbb{H}_2(\widetilde{R}|S_1 = s_1, S_2 = s_2) \leq 2\mathbb{H}_\infty(\widetilde{R}|S_1 = s_1, S_2 = s_2) < 2$, and Eq. (13) is due to the fact that $k'' > 1$ whenever $n, c^* \geq 5$ (as in the statement of the theorem). Note that Eq. (13) contradicts Eq. (10), since the latter implies $\mathbf{Adv}^{\text{det}}_{\Pi,\Psi,\mathsf{W}_{\text{rand}}}(\lambda) < \frac{1}{c^*}$.

Finally, we can compute the conditional average min-entropy of $\widetilde{R}$ conditioned on $S_1, S_2$ as:

$$2^{-\widetilde{\mathbb{H}}_\infty(\widetilde{R}|S_1,S_2)} = \mathbb{E}_{s_1,s_2\leftarrow\mathsf{S}}\left[2^{-\mathbb{H}_\infty(\widetilde{R}|S_1=s_1,S_2=s_2)}\right] \leq \mathbb{E}_{s_1,s_2\leftarrow\mathsf{S}}\left[2^{-1}\right] = \frac{1}{2},$$

and thus, $\widetilde{\mathbb{H}}_\infty(\widetilde{R}|S_1, S_2) \geq 1$.

Next, we use the fact that $\widetilde{\mathbb{H}}_\infty(\widetilde{R}|S_1, S_2) \geq 1$ in order to show that game $\mathbf{G}_2(\lambda)$ and $\mathbf{G}_3(\lambda)$ are computationally indistinguishable. To this end, consider the intermediate experiment $\mathbf{H}(\lambda)$ that is identical to $\mathbf{G}_2(\lambda)$, except that the challenger now generates $\rho$ by running $\mathsf{P}$ on uniformly random coins $y_1 \overset{\$}{\leftarrow} \mathsf{Y}(1^\lambda)$, where $\mathsf{Y}$ outputs strings of length $m$ bits according to the distribution $Y$ guaranteed by Definition 8. We first show that $\mathbf{G}_2(\lambda)$ is computationally indistinguishable from $\mathbf{H}(\lambda)$. By contradiction, assume that there is a distinguisher $\mathsf{D}$ with running time $t_\mathsf{D} = t_\mathsf{A}$ such that $\Delta_\mathsf{D}(\mathbf{G}_2(\lambda); \mathbf{H}(\lambda)) > \epsilon_{\text{cond}}$. We construct a distinguisher $\mathsf{D}'$ breaking security of the hash family $\mathcal{G}$. Distinguisher $\mathsf{D}'$ proceeds as follows:

1. Receive seed $s_1 := s$ from the challenger, sample $s_2 \overset{\$}{\leftarrow} \{0,1\}^\ell$, and run $\mathsf{D}$ upon input $(s_1, s_2)$ obtaining (a description of) $\widetilde{\Pi} = (\widetilde{\mathsf{P}}, \widetilde{\mathsf{K}}, \widetilde{\mathsf{R}}, \widetilde{\mathsf{F}}_1, \ldots, \widetilde{\mathsf{F}}_N)$.

2. Forward $X := \widetilde{R}^k$ to the challenger, where $\widetilde{R}^k$ is the concatenation of $k$ identical copies of $\widetilde{R}$.

3. Emulate the challenger by answering $\mathsf{D}$'s queries exactly as described in game $\mathbf{G}_2(\lambda)$, except that the public parameters are set by default to be equal to $\rho := \mathsf{P}(1^\lambda; y_1)$, where $y_1 := y \in \{0,1\}^m$ is the challenge in the security game of the condenser.
4. Output the same guess as that of $\mathsf{D}$.

Observe that $\mathsf{D}'$ runs in time $t_{\text{cond}} \approx t_{\mathsf{A}}$, and moreover, as we argue below, the simulation is perfect. In fact, the only difference between the two games is in how the public parameters are set by the challenger:

- In case $y_1 \xleftarrow{\$} Y$, the public parameters are computed as $\rho := \mathsf{P}(1^\lambda; y_1)$ as defined in game $\mathbf{H}(\lambda)$.
- In case $y_1$ is defined through the condenser, the public parameters are computed as $\rho := \mathsf{P}(1^\lambda; g_{s_1}(r_1))$ for $r_1 \xleftarrow{\$} \widetilde{R}^k$; this is exactly what happens in game $\mathbf{G}_2(\lambda)$.

Finally, we claim that the random variable $\widetilde{R}^k$ has $k$ bits of conditional average min-entropy (conditioned on $S_1, S_2$). The latter can be seen as follows:

$$\begin{aligned}
2^{-\widetilde{\mathbb{H}}_\infty(\widetilde{R}^k|S_1,S_2)} &= \mathbb{E}_{s_1,s_2\leftarrow\mathsf{S}}\left[2^{-\mathbb{H}_\infty(\widetilde{R}^k|S_1=s_1,S_2=s_2)}\right] \\
&= \mathbb{E}_{s_1,s_2\leftarrow\mathsf{S}}\left[\max_{r_1,\ldots,r_k} \mathbb{P}\left[\widetilde{R}_1 = r_1,\ldots,\widetilde{R}_k = r_k | S_1 = s_1, S_2 = s_2\right]\right] \\
&= \mathbb{E}_{s_1,s_2\leftarrow\mathsf{S}}\left[\max_{r_1,\ldots,r_k} \prod_{i=1}^{k} \mathbb{P}\left[\widetilde{R}_i = r_i | S_1 = s_1, S_2 = s_2\right]\right] \\
&\le \mathbb{E}_{s_1,s_2\leftarrow\mathsf{S}}\left[2^{-k}\right] = 2^{-k},
\end{aligned} \tag{14}$$

where the sources $(\widetilde{R}_i)_{i\in[k]}$ are identical copies of $\widetilde{R}$, and where we used the fact that $\widetilde{R}$ is efficiently samplable with $\widetilde{\mathbb{H}}_\infty(\widetilde{R}|S_1,S_2) \ge 1$, and furthermore can wlog. be assumed to be stateless [29, Remark 2.5].

Hence, $\mathsf{D}'$ retains the same advantage as that of $\mathsf{D}$, which contradicts security of the family of condensers $\mathcal{G}$. An analogous[24] argument shows that game $\mathbf{H}(\lambda)$ and $\mathbf{G}_3(\lambda)$ are within distance $\epsilon_{\text{cond}}$, which concludes the proof. □

Finally, note that game $\mathbf{G}_3(\lambda)$ is identically distributed to the security game $\mathbf{G}$, except that the random coins that are fed as input to, respectively, $\mathsf{P}$ and $\mathsf{K}$ come from a min-entropy source. This corresponds to the $(m-d)$-real setting defined in §2.3. Hence, by combining the above lemmas with Eq. (9), after plugging the expression for the detect advantage $\delta^*$, and applying Theorem 1, we obtain:

(i) If $\mathbf{G}$ is an unpredictability game

$$\begin{aligned}
2^{2d}\cdot\epsilon &\ge |\mathbb{P}[\mathbf{G}_3(\lambda) = 1] - \gamma| \\
&\ge \mathbf{Adv}^{\text{pub}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) - \delta^* - 2\epsilon_{\text{cond}} > \frac{c^*-1}{c^*}\cdot\epsilon^* - 2\epsilon_{\text{cond}};
\end{aligned}$$

(ii) If $\mathbf{G}$ is an indistinguishability game

$$\begin{aligned}
\sqrt{2^{2d}\cdot\epsilon} &\ge |\mathbb{P}[\mathbf{G}_3(\lambda) = 1] - \gamma| \\
&\ge \mathbf{Adv}^{\text{pub}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) - \delta^* - 2\epsilon_{\text{cond}} > \frac{c^*-1}{c^*}\cdot\epsilon^* - 2\epsilon_{\text{cond}}.
\end{aligned}$$

[24]The only difference is that $\mathsf{D}'$ forwards $(s_1, s)$ to $\mathsf{D}$, where $s_1 \xleftarrow{\$} \{0,1\}^\ell$, and moreover it defines $(pk, sk) := \mathsf{K}(1^\lambda, \rho; y)$, with $\rho = \mathsf{P}(1^\lambda; y_1)$ for $y_1 \xleftarrow{\$} \mathsf{Y}(1^\lambda)$.

In both cases we reach a contradiction on the expression for $\epsilon$ in the theorem statement. This finishes the proof. $\square$

**Remark 6** (On square security)**.** The reason for which Theorem 3 does not work for all deterministic primitives is that its proof crucially relies on the "overcoming weak expectations" framework (cf. Theorem 1 in §2.3). In particular, for single-instance indistinguishability games, this theorem requires square security, and it is well known that some primitives such as pseudorandom generators and pseudorandom functions do not have good square security [6, 19].

One can also show that the limitation of Remark 6 is inherent, in the sense that our immunizer is in general insecure for primitives that are not square friendly. Take, for instance, any PRG $\Pi = (\mathsf{R}, \mathsf{K}, \mathsf{PRG})$, where $\mathsf{K}(1^\lambda; r) = r$ outputs directly a seed sampled from the secret source $\mathsf{R}$, and $\mathsf{PRG}(1^\lambda, r)$ stretches the seed to a pseudorandom output. Let $\Pi^* = (\mathsf{K}^*, \mathsf{R}, \mathsf{PRG}^*) = \Psi(\Pi)$ be the immunized version of $\Pi$. Now, consider the attacker $\mathsf{A}(s)$ that plays the subversion game by specifying the subversion $\widetilde{\Pi}$ where:

- $\widetilde{\mathsf{K}}$ and $\widetilde{\mathsf{PRG}}$ are unchanged (*i.e.*, $\widetilde{\mathsf{K}} \equiv \mathsf{K}^*$, and $\widetilde{\mathsf{PRG}} \equiv \mathsf{PRG}^*$);
- $\widetilde{\mathsf{R}}$ embeds a key $\kappa$ for a pseudorandom function $\mathsf{PRF}$ with one-bit output, and performs the following rejection-sampling procedure:
  - Sample a random $r$;
  - If $\mathsf{PRF}(1^\lambda, \kappa, y) = 1$, where $\mathsf{PRG}(h_s(r)) = y$, return $r$;
  - Else, sample a fresh $r$ and start again.

Intuitively, the above subversion allows $\mathsf{A}$ to win the subversion game by simply checking whether $\mathsf{PRF}(1^\lambda, \kappa, y) = 1$, where $y$ is the challenge. Moreover, this attack is undetectable as a watchdog not knowing the key $\kappa$ has a negligible advantage in distinguishing $\widetilde{\mathsf{R}}$ from $\mathsf{R}$ (by security of the pseudorandom function). Note that the above attack requires the adversary to choose the subversion depending on the seed.

**Instantiating the immunizer.** When instantiating seed-dependent randomness condensers with state-of-the-art constructions [18, 19], we obtain the following parameters.

**Corollary 2.** *Let* $\mathsf{R}$ *and* $\mathsf{S}$ *be, respectively, uniformly random* $m$*-bit and* $m'$*-bit sources. Assuming the existence of* $(\mathsf{poly}(\lambda), \mathsf{poly}(\lambda)/2^m)$*-collision-resistant families of hash functions, for any cryptographic primitive* $\Pi$ *that is either* $(\mathsf{poly}(\lambda), \mathsf{negl}(\lambda))$*-secure (in case of unpredictability games) or* $(\mathsf{poly}(\lambda), \mathsf{negl}(\lambda))$-square-*secure (in case of indistinguishability games) w.r.t. an* input-constrained, single-instance *game* $\mathbf{G}$*, there exists an immunizer* $\Psi_{k,1}^{\mathsf{R},\mathsf{S}}$ *for* $\Pi$ *that is* $(\mathsf{poly}(\lambda), \mathsf{poly}(\lambda), 5, \mathsf{negl}(\lambda))$*-subversion-resistant for the* pub*-model with an offline watchdog, with parameters* $k, \ell, m \in \omega(\log(\lambda))$*.*

*Proof.* Setting $k, \ell, m \in \omega(\log(\lambda))$ in Lemma 4, we can see that any $(\mathsf{poly}(\lambda), \mathsf{poly}(\lambda)/2^m)$-collision-resistant family of hash functions yields a family of condensers that achieves $t_{\text{cond}} \in \mathsf{poly}(\lambda)$, $\epsilon_{\text{cond}} = 0$, and entropy deficiency $d \in O(\log(\lambda))$. Note that this is enough to tolerate the loss of $2^{2d}$ in the security of the immunizer, starting with any $\Pi$ with polynomial security. Hence, the statement follows directly by plugging the above parameters in Theorem 3. $\square$

## 6 Transparent Immunizer

In this section we provide a different instantiation for the immunizer from Fig. 3, and show that it achieves security in the *transparent* model. The instantiation relies on a new notion of seed-dependent condensers with weak seeds, which we define and construct in §6.1. The security analysis appears in §6.2.

### 6.1 Ingredients: Condensers with Weak Seeds

Intuitively, a randomness condenser with weak seeds is a condenser working with seeds that only have min-entropy (instead of being uniform). We state a definition below for the seed-dependent case, where the source is allowed to further depend on the seed.

**Definition 10** (Seed-dependent condenser with weak seeds)**.** Let $\mathcal{G} := \{g_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^\ell}$ be a family of efficiently computable functions. We say that $\mathcal{G}$ is a family of $(\frac{k}{n} \to \frac{m-d}{m}, k', t, \epsilon)$-seed-dependent condensers with weak seeds if for all probabilistic adversaries $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ running in time $t$ such that $\mathsf{A}_0$ outputs a distribution $S$ for which $\mathbb{H}_\infty(S) \geq k'$, and $\mathsf{A}_1$ takes as input a seed $s \xleftarrow{\$} S$ and outputs a distribution $X$ for which $\widetilde{\mathbb{H}}_\infty(X|S) \geq k$, the joint distribution $(S, g_S(X))$ is $\epsilon$-close to some $(S, Y)$ with $\widetilde{\mathbb{H}}_\infty(Y|S) \geq m - d$.

Notice that the attacker has the power to influence both the distribution of the seed $S$ and of the random source $X$. However, it must first commit to the distribution of the seed, and later can choose the distribution of the source depending on the seed. We stress that this notion is different than the notion of seed-dependent condensers for leaky sources studied by [18].[25] In fact, we establish that the above notion is already satisfied by any collision-resistant hash function that tolerates weak seeds of min-entropy $k'$.

**Lemma 7.** *Let $\mathcal{G}$ be a family of $(2t, \epsilon)$-collision-resistant hash functions that is secure in the $k'$-real model. Then $\mathcal{G}$ is a family of $(\frac{k}{n} \to \frac{m-d}{m}, k', t, 0)$-seed-dependent condensers with weak seeds, for $2^d = 2^{m-k} + 2^m\epsilon$.*

*Proof.* Let $X_1, X_2 \xleftarrow{\$} \mathsf{A}_1(S)$ for $S \xleftarrow{\$} \mathsf{A}_0(1^\lambda)$. We can estimate the collision probability as follows:

$$\begin{aligned}
\mathbb{P}[g_S(X_1) = g_S(X_2)] &\leq \mathbb{P}[X_1 = X_2] + \mathbb{P}[g_S(X_1) = g_S(X_2) \wedge X_1 \neq X_2] \\
&\leq \mathbb{P}[X_1 = X_2] + \epsilon && (15)\\
&\leq 2^{-k} + \epsilon && (16)\\
&= 2^{-m}\left(2^{m-k} + 2^m\epsilon\right) := 2^{d-m}, && (17)
\end{aligned}$$

where Eq. (17) follows by the expression of $d$ in the theorem statement, Eq. (16) follows by the fact that $\mathbb{H}_2(X) \geq \widetilde{\mathbb{H}}_2(X|S) \geq \widetilde{\mathbb{H}}_\infty(X|S) \geq k$, while Eq. (15) follows by the fact that $\mathcal{G}$ is collision-resistant in the $k'$-real model. The latter can be seen as follows. Assume that $\mathbb{P}[g_S(X_1) = g_S(X_2) \wedge X_1 \neq X_2] > \epsilon$, then we can define an adversary $\mathsf{B}$ attacking the hash function w.r.t. seed distribution $S \xleftarrow{\$} \mathsf{A}_0(1^\lambda)$:

- Given a seed $s \xleftarrow{\$} S$, run $\mathsf{A}_1(s)$ twice obtaining distributions $X_1, X_2$.
- Sample $x_1 \xleftarrow{\$} X_1$ and $x_2 \xleftarrow{\$} X_2$, and output $(x_1, x_2)$.

[25]The difference is that in [18] the adversary outputting the seed-dependent distribution $X$ can also pass some *bounded leakage* $Z$ on $X$ to the distinguisher, which makes their definition much trickier to achieve.

Adversary $\mathsf{B}$ runs in time $t_{\mathsf{B}} \approx 2t$ and succeeds with probability at least $\epsilon$; since by assumption $\mathbb{H}_\infty(S) \geq k'$, this contradicts the fact that $\mathcal{G}$ is $(2t, \epsilon)$-collision-resistant in the $k'$-real model. □

By Lemma 7 and Theorem 1 we derive the following corollary.

**Corollary 3.** *Let $\mathcal{G} := \{g_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^\ell}$ be a family of $(2t, 2^{k'}\epsilon)$-collision-resistant hash functions in the ideal model. Then $\mathcal{G}$ is a $(\frac{k}{n} \to \frac{m-d}{m}, k', t, 0)$-seed-dependent condenser with weak seeds, for $d = m - k + \log(1 + 2^{k'}\epsilon)$, , $k \leq n$, and $k' \leq \ell$.*

## 6.2 Security Analysis (Transparent Model)

The theorem below says that when we instantiate the immunizer of Fig. 3 with a family of seed-dependent condensers with weak seeds, we achieve subversion security in the transparent model. However, in this case, since the public source $\mathsf{S}$ can be subverted, it will be crucial to exploit the fact that the final seed $s = (s_1, s_2)$ is obtained by amalgamating $k' > 1$ independent samples from $\mathsf{S}$, in a trusted way (see also Remark 2).

For input-constrained games, we obtain the following result. An analogous statement holds for input-unconstrained games, in the online watchdog model (cf. §7.3).

**Theorem 4.** *Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a deterministic cryptographic scheme with $\mathcal{R} = \{0,1\}^m$, and $\mathsf{S}$ be uniform over $\{0,1\}^{m'}$. Consider any* input-constrained, single-instance *game $\mathbf{G} = (\mathsf{C}, \gamma)$ for $\Pi$. Then, for any $c^* > 4$, the immunizer $\Psi^{\mathsf{R},\mathsf{S}}_{k,k'}[\mathcal{G}]$ of Fig. 3 is $(t_{\mathsf{A}}, t_{\mathsf{W}}, c^*, \epsilon^*)$-subversion-resistant for the* trans-*model with an* offline *watchdog, as long as $\mathcal{G} := \{g_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^\ell}$ is a family of $(\frac{k}{n} \to \frac{m-d}{m}, k', t_{\text{cond}}, \epsilon_{\text{cond}})$-seed-dependent condensers with weak seeds and $\Pi$ is either $(t, \epsilon)$-secure w.r.t. game $\mathbf{G}$ (in case of unpredictability games) or $(t, \epsilon)$-square-secure w.r.t. game $\mathbf{G}$ (in case of indistinguishability games), for parameters $t_{\text{cond}}, t, t_{\mathsf{W}} \approx t_{\mathsf{A}}$, and*

$$\epsilon \leq \begin{cases} \frac{c^*-1}{c^*} \cdot \frac{\epsilon^*}{2^{2d}} - \frac{2\epsilon_{\text{cond}}}{2^{2d}} & \textit{if } \mathbf{G} \textit{ is an unpredictability game} \\ \left(\frac{c^*-1}{c^*} \cdot \epsilon^* - 2\epsilon_{\text{cond}}\right)^2 \cdot \frac{1}{2^{2d}} & \textit{if } \mathbf{G} \textit{ is an indistinguishability game.} \end{cases}$$

*Proof.* We prove Theorem 4 by using an identical proof strategy to that of Theorem 3. The only difference is that now the public source $\mathsf{S}$ can also be tampered, which justifies the need for seed-dependent randomness condensers with weak seeds. We include the full proof for self-containment. By contradiction, assume that the immunizer is not subversion resistant, *i.e.* there exists an adversary $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ with running time $t_{\mathsf{A}}$ such that for all watchdogs $\mathsf{W}$ with running time $t_{\mathsf{W}}$ we have

$$\mathbf{Adv}^{\text{trans}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) = \left|\mathbb{P}\left[\mathbf{G}^{\text{trans}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) = 1\right] - \gamma\right| > \epsilon^* \tag{18}$$

$$\mathbf{Adv}^{\text{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda) < \frac{1}{c^*} \cdot \mathbf{Adv}^{\text{trans}}_{\Pi,\Psi,\mathsf{A},\mathsf{C}}(\lambda) := \delta^*. \tag{19}$$

We introduce a sequence of intermediate hybrid experiments that are modified in an incremental manner. The games are defined exactly as in the proof of Theorem 3, except that the seeds $s_1, s_2$ are generated by sampling $k'$ times from $\widetilde{\mathsf{S}}$. More in details, we now have $s_1 = s^1_1 || \cdots || s^1_{k'}, s_2 = s^2_1 || \cdots || s^2_{k'}$ for $k' = \ell/m'$, $s^j_i \xleftarrow{\$} \widetilde{\mathsf{S}}(1^\lambda)$ and $(i,j) \in [k'] \times [2]$. The games are formally described in Fig. 6. We proceed to show that the games are all computationally indistinguishable, and thus $\mathsf{A}$'s advantage does not vanish across the different games.

**Lemma 8.** $|\,\mathbb{P}[\mathbf{G}_1(\lambda) = 1] - \mathbb{P}[\mathbf{G}_2(\lambda) = 1]\,| \leq \delta^*$.

```
$\mathbf{G}_1(\lambda)$
$(\widetilde{\mathsf{S}}, \alpha_0) \overset{\$}{\leftarrow} \mathsf{A}_0(1^\lambda, \langle \Pi, \Psi \rangle)$
$s_1^1, \ldots, s_{k'}^1, s_1^2, \ldots, s_{k'}^2 \overset{\$}{\leftarrow} \widetilde{\mathsf{S}}(1^\lambda)$;
$s_1 = s_1^1 || \cdots || s_{k'}^1; s_2 = s_1^2 || \cdots || s_{k'}^2$;
$(\widetilde{\Pi}, \alpha_1) \overset{\$}{\leftarrow} \mathsf{A}_1(1^\lambda, \alpha_0, s_1, s_2)$
$r_1^1, \ldots, r_k^1, r_1^2, \ldots, r_k^2 \overset{\$}{\leftarrow} \widetilde{\mathsf{R}}(1^\lambda)$;
$r_1 = r_1^1 || \cdots || r_k^1; r_2 = r_1^2 || \cdots || r_k^2$;
$d \overset{\$}{\leftarrow} \langle \mathsf{A}_2(1^\lambda, \alpha_1), \mathsf{C}^{\widetilde{\mathsf{P}}(1^\lambda, s_1, \cdot), \widetilde{\mathsf{K}}(1^\lambda, s_2, \cdot, \cdot), \widetilde{\mathsf{F}}_1, \ldots, \widetilde{\mathsf{F}}_N}(1^\lambda, r_1, r_2) \rangle$
return $d$
```

```
$\mathbf{G}_2(\lambda)$
$(\widetilde{\mathsf{S}}, \alpha_0) \overset{\$}{\leftarrow} \mathsf{A}_0(1^\lambda, \langle \Pi, \Psi \rangle)$
$s_1^1, \ldots, s_{k'}^1, s_1^2, \ldots, s_{k'}^2 \overset{\$}{\leftarrow} \widetilde{\mathsf{S}}(1^\lambda)$;
$s_1 = s_1^1 || \cdots || s_{k'}^1; s_2 = s_1^2 || \cdots || s_{k'}^2$;
$(\widetilde{\Pi}, \alpha_1) \overset{\$}{\leftarrow} \mathsf{A}_1(1^\lambda, \alpha_0, s_1, s_2)$
$r_1^1, \ldots, r_k^1, r_1^2, \ldots, r_k^2 \overset{\$}{\leftarrow} \widetilde{\mathsf{R}}(1^\lambda)$;
$r_1 = r_1^1 || \cdots || r_k^1; r_2 = r_1^2 || \cdots || r_k^2$;
$d \overset{\$}{\leftarrow} \langle \mathsf{A}_2(1^\lambda, \alpha_1), \mathsf{C}^{\mathsf{P}^*(1^\lambda, s_1, \cdot), \mathsf{K}^*(1^\lambda, s_2, \cdot, \cdot), \mathsf{F}_1, \ldots, \mathsf{F}_N}(1^\lambda, r_1, r_2) \rangle$
return $d$
```

```
$\mathbf{G}_3(\lambda)$
$(\widetilde{\mathsf{S}}, \alpha_0) \overset{\$}{\leftarrow} \mathsf{A}_0(1^\lambda, \langle \Pi, \Psi \rangle)$
$s_1^1, \ldots, s_{k'}^1, s_1^2, \ldots, s_{k'}^2 \overset{\$}{\leftarrow} \widetilde{\mathsf{S}}(1^\lambda)$;
$s_1 = s_1^1 || \cdots || s_{k'}^1; s_2 = s_1^2 || \cdots || s_{k'}^2$;
$(\widetilde{\Pi}, \alpha_1) \overset{\$}{\leftarrow} \mathsf{A}_1(1^\lambda, \alpha_0, s_1, s_2)$
$y_1, y_2 \overset{\$}{\leftarrow} \mathsf{Y}(1^\lambda)$;
$d \overset{\$}{\leftarrow} \langle \mathsf{A}_2(1^\lambda, \alpha_1), \mathsf{C}^{\mathsf{P}, \mathsf{K}, \mathsf{F}_1, \ldots, \mathsf{F}_N}(1^\lambda, y_1, y_2) \rangle$
return $d$
```

Figure 6: Game hops in the proof of Theorem 4.

*Proof.* The proof is identical to that of Lemma 2. The only difference is that the watchdog $\mathsf{W}^*$ uses the seeds $s_1, s_2$ that now are part of its own input, instead of sampling the seeds from $\mathsf{S}^*$, in order to check the implementations of $\mathsf{P}^*, \mathsf{K}^*$, and $(\mathsf{F}_i^*)_{i \in [N]}$. $\square$

**Lemma 9.** $|\,\mathbb{P}[\mathbf{G}_2(\lambda) = 1] - \mathbb{P}[\mathbf{G}_3(\lambda) = 1]\,| \leq 2\epsilon_{\text{cond}}$.

*Proof.* Let $\widetilde{R}$ be the random variable associated to the subverted source $\widetilde{\mathsf{R}}$ output by $\mathsf{A}_1$, let $\widetilde{S}$ be the random variable corresponding to the subverted source $\widetilde{\mathsf{S}}$ output by $\mathsf{A}_0$, and let $\widetilde{S}_1, \widetilde{S}_2$ be identical copies of $\widetilde{S}$. An argument identical to the one given at the beginning of the proof of

Lemma 3 and Lemma 6 shows that $\mathbb{H}_\infty(\widetilde{S}) \geq 1$ and $\widetilde{\mathbb{H}}_\infty(\widetilde{R}|\widetilde{S}_1, \widetilde{S}_2) \geq 1$. Next, we use this fact in order to show that game $\mathbf{G}_2(\lambda)$ and $\mathbf{G}_3(\lambda)$ are computationally indistinguishable. To this end, consider the intermediate experiment $\mathbf{H}(\lambda)$ that is identical to $\mathbf{G}_2(\lambda)$, except that the challenger now generates $\rho$ by running $\mathsf{P}$ on uniformly random coins $y_1 \overset{\$}{\leftarrow} \mathsf{Y}(1^\lambda)$, where $\mathsf{Y}$ outputs strings of length $m$ bits according to the distribution $Y$ guaranteed by Definition 10. We first show that $\mathbf{G}_2(\lambda)$ is computationally indistinguishable from $\mathbf{H}(\lambda)$. By contradiction, assume that there is a distinguisher $\mathsf{D}$ with running time $t_\mathsf{D} = t_\mathsf{A}$ such that $\Delta_\mathsf{D}(\mathbf{G}_2(\lambda); \mathbf{H}(\lambda)) > \epsilon_{\mathsf{cond}}$. We construct a distinguisher $\mathsf{D}'$ breaking security of the hash family $\mathcal{G}$. Distinguisher $\mathsf{D}'$ proceeds as follows:

1. Run $\mathsf{D}$, obtaining a distribution $\widetilde{S}$, and forward $S := \widetilde{S}^{k'}$ to the challenger where $\widetilde{S}^{k'}$ is the concatenation of $k'$ identical copies of $\widetilde{S}$.
2. Receive the seed $s_1 := s_1^1 || \cdots || s_{k'}^1 := s$ from the challenger, and define $s_2 = s_1^2 || \cdots || s_{k'}^2$ for $s_i^j \overset{\$}{\leftarrow} \widetilde{S}$ and $(i, j) \in [k'] \times [2]$.
3. Run $\mathsf{D}$ upon input $(s_1, s_2)$ obtaining (a description of) $\widetilde{\Pi} = (\widetilde{\mathsf{P}}, \widetilde{\mathsf{K}}, \widetilde{\mathsf{R}}, \widetilde{\mathsf{F}}_1, \ldots, \widetilde{\mathsf{F}}_N)$.
4. Forward $X := \widetilde{R}^k$ to the challenger, where $\widetilde{R}^k$ is the concatenation of $k$ identical copies of $\widetilde{R}$.
5. Emulate the challenger by answering $\mathsf{D}$'s queries exactly as described in game $\mathbf{G}_2(\lambda)$, except that the public parameters are set by default to be equal to $\rho := \mathsf{P}(1^\lambda; y_1)$, where $y_1 := y \in \{0,1\}^m$ is the challenge in the security game of the condenser.
6. Output the same guess as that of $\mathsf{D}$.

Observe that $\mathsf{D}'$ runs in time $t_{\mathsf{cond}} \approx t_\mathsf{A}$, and moreover the simulation is perfect. Notice, in fact, that the only difference between the two games is in how the public parameters are set by the challenger. In particular:

- In case $y_1 \overset{\$}{\leftarrow} Y$, the public parameters are computed as $\rho := \mathsf{P}(1^\lambda; y_1)$ as defined in game $\mathbf{H}(\lambda)$.
- In case $y_1$ is defined through the condenser, the public parameters are computed as $\rho := \mathsf{P}(1^\lambda; g_{s_1}(r_1))$ for $r_1 \overset{\$}{\leftarrow} \widetilde{R}^k$; this is exactly what happens in game $\mathbf{G}_2(\lambda)$.

Finally, an argument similar to those used to derive Eq. (8) (in the proof of Lemma 2) and Eq. (14) (in the proof of Lemma 6) shows that $\mathbb{H}_\infty(S) = \mathbb{H}_\infty(\widetilde{S}^{k'}) \geq k'$ and $\widetilde{\mathbb{H}}_\infty(X|S) = \widetilde{\mathbb{H}}_\infty(\widetilde{R}^k|S) \geq k$. Hence, $\mathsf{D}'$ retains the same advantage as that of $\mathsf{D}$, which contradicts security of the family of condensers $\mathcal{G}$. $\square$

As in the proof of Theorem 3, the statement now follows by observing that game $\mathbf{G}_3(\lambda)$ is identically distributed to the security game $\mathbf{G}$, except that the random coins $y_1, y_2$ that are fed as input to, respectively, $\mathsf{P}$ and $\mathsf{K}$ come from a min-entropy source. $\square$

**Remark 7** (On square security)**.** As for the public model, Theorem 4 does not cover all deterministic primitives, but only those for which we can invoke Theorem 1. See also Remark 6.

When instantiating seed-dependent randomness condensers with weak seeds using sufficiently strong collision-resistant hash families, we obtain the following parameters.

**Corollary 4.** *Let* $\mathsf{R}$ *and* $\mathsf{S}$ *be, respectively, uniformly random* $m$*-bit and* $m'$*-bit sources. Assuming the existence of* $(\mathsf{poly}(\lambda), \mathsf{poly}(\lambda)/2^m)$*-collision-resistant families of hash functions, for any* $\alpha < 1$*, and for any cryptographic primitive* $\Pi$ *that is either* $(2^{\lambda^\alpha}, 2^{-\lambda^\alpha})$*-secure (in case of unpredictability games) or* $(2^{\lambda^\alpha}, 2^{-\lambda^\alpha})$-square-*secure (in case of indistinguishability games) w.r.t.* input-constrained, single-instance *game* $\mathbf{G}$*, there exists an immunizer* $\Psi_{k,k'}^{\mathsf{R},\mathsf{S}}$ *for* $\Pi$ *that is* $(\mathsf{poly}(\lambda), \mathsf{poly}(\lambda), 5, \mathsf{negl}(\lambda))$*-subversion-resistant for the* trans-*model with an offline watchdog, with parameters* $k, k', m, m' \in \omega(\log(\lambda))$.

*Proof.* Setting $k, k', m \in \omega(\log(\lambda))$ in Corollary 3, we can see that any $(\mathsf{poly}(\lambda), \mathsf{poly}(\lambda)/2^m)$-collision-resistant family of hash functions yields a family $\mathcal{G}$ of condensers with weak seeds that achieves $t_{\text{cond}} \in \mathsf{poly}(\lambda)$, $\epsilon_{\text{cond}} = 0$, and entropy deficiency $d \in \omega(\log(\lambda))$. Note that this is enough to tolerate the loss of $2^{2d}$ in the security of the immunizer, starting with any $\Pi$ with sub-exponential security. Hence, the statement follows directly by plugging the above parameters in Theorem 4. □

## 7 Security Statements in the Online Watchdog Model

In this section, we revisit the security statements for our immunizer in the semi-private, public, and transparent model in the case of input unconstrained games. As we show, for such games we can only obtain positive results in the online watchdog model. Intuitively, this is the case because the challenger is required to query the subversion game oracles on inputs chosen by the adversary (*i.e.*, the game is input unconstrained), and thus it might be *impossible* for the watchdog to detect subversion without being given a transcript of the interaction between the challenger and the adversary (*e.g.*, input-triggered attacks are always possible).

### 7.1 Semi-Private Immunization

The statement below is the equivalent of Theorem 2 for input-unconstrained games.

**Theorem 5.** *Let* $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ *be a deterministic cryptographic scheme with* $\mathcal{R} = \{0,1\}^m$*, and* $\mathsf{S}$ *be uniform over* $\{0,1\}^\ell$*. Consider any* input-unconstrained, single-instance *game* $\mathbf{G} = (\mathsf{C}, \gamma)$ *for* $\Pi$*. Then, for any* $c^* > 4$*, the immunizer* $\Psi_{k,1}^{\mathsf{R},\mathsf{S}}[\mathcal{H}]$ *of Fig. 3 is* $(t_{\mathsf{A}}, t_{\mathsf{W}}, c^*, \epsilon^*)$*-subversion-resistant for the* spriv*-model with an* online *watchdog, as long as* $\mathcal{H} := \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^\ell}$ *is a family of strong* $(n, m, k, t_{\text{ext}}, \epsilon_{\text{ext}})$*-computational extractors and* $\Pi$ *is* $(t, \epsilon)$*-secure w.r.t. game* $\mathbf{G}$ *for parameters* $t_{\text{ext}}, t_{\mathsf{W}}, t \approx t_{\mathsf{A}}$*, and*

$$\epsilon \leq \frac{c^* - 1}{c^*} \cdot \epsilon^* - 2\epsilon_{\text{ext}}.$$

*Proof sketch.* The proof follows closely that of Theorem 2, and thus we only emphasize the main differences. We consider exactly the same hybrid experiments $\mathbf{G}_1(\lambda)$–$\mathbf{G}_3(\lambda)$. The only step of the proof that changes is the one where we analyze the transition from $\mathbf{G}_1(\lambda)$ to $\mathbf{G}_2(\lambda)$. Hence, in what follows, we focus only on this particular step of the proof.

**Lemma 10.** $|\mathbb{P}[\mathbf{G}_1(\lambda) = 1] - \mathbb{P}[\mathbf{G}_2(\lambda) = 1]| \leq \delta^*$.

*Proof.* Consider the same events $E_{\text{pub}}$, $E_{\text{kgen}}$, and $E_{\text{func}}$, as in the proof of Lemma 2, except that event $E_{\text{func}}$ is now defined as follows.

**Event** $E_{\text{func}}$**:** The event becomes true whenever $\exists i \in [N], j \in [q], x_j \in \widetilde{\tau}$ s.t. $\widetilde{\mathsf{F}}_i(1^\lambda, \rho, (pk, sk), x_j) \neq \mathsf{F}_i^*(1^\lambda, \rho, (pk, sk), x_j)$, where $\rho = \widetilde{\mathsf{P}}(1^\lambda, s_1; r_1)$, $(pk, sk) = \widetilde{\mathsf{K}}(1^\lambda, s_2, \rho; r_2)$, $r_1 = r_1^1 || \ldots || r_k^1$ for $r_i^1 \overset{\$}{\leftarrow} \widetilde{\mathsf{R}}(1^\lambda)$, $r_2 = r_1^2 || \ldots || r_k^2$ for $r_i^2 \overset{\$}{\leftarrow} \widetilde{\mathsf{R}}(1^\lambda)$, $s_1, s_2 \overset{\$}{\leftarrow} \mathsf{S}(1^\lambda)$, $\widetilde{\tau}$ is the transcript of the messages exchanged between the adversary and the challenger in the subversion game (recall that the game is input unconstrained), and $q(\lambda) \in \mathsf{poly}(\lambda)$ is an upper bound on the total number of queries asked by the adversary to each of the oracles $\widetilde{\mathsf{F}}_1, \ldots, \widetilde{\mathsf{F}}_N$.

Define $E_{\text{bad}} := E_{\text{pub}} \vee E_{\text{kgen}} \vee E_{\text{func}}$. Clearly, if $E_{\text{bad}}$ does not happen, we have that $\mathbf{G}_1(\lambda)$ and $\mathbf{G}_2(\lambda)$ are identically distributed, and thus it suffices to prove that event $E_{\text{bad}}$ only happens with probability smaller than $\delta^*$. By contradiction, assume that $\mathsf{A}$ provokes event $E_{\text{bad}}$ with probability at least $\delta^*$. We construct an efficient online watchdog $\mathsf{W}^*$ that distinguishes between $\mathbf{G}^{\text{det-on}}_{\Pi,\Psi,\mathsf{W}^*}(\lambda, \mathtt{aux}, 0)$ and $\mathbf{G}^{\text{det-on}}_{\Pi,\Psi,\mathsf{W}^*}(\lambda, \mathtt{aux}, 1)$ with the same probability, which contradicts Eq. (4).

The watchdog $\mathsf{W}^*$ is identical to the one constructed in the proof of Lemma 2, except that step 5 is changed as follows:

5' For each $i \in [N]$, $j \in [q]$, and $x_j \in \widetilde{\tau}$, invoke $\mathsf{O}^i_{\text{func}}$ upon input $(1^\lambda, \rho, (pk, sk), x_j)$, obtaining some output $y_j$; in case there exists $i \in [N]$ and $j \in [q]$ such that the value $y_j$ does not equal $\mathsf{F}^*_i(1^\lambda, \rho, (pk, sk), x_j) = \mathsf{F}_i(1^\lambda, \rho, (pk, sk), x_j)$ output 1 and stop.

Recall that the watchdog $\mathsf{W}^*$ is always given the original specification of $\langle \Psi, \Pi \rangle$, and additionally, since the watchdog is online, it also receives the transcript $\widetilde{\tau}$ resulting from the execution of the subversion game with the adversary $\mathsf{A}$. Hence, $\mathsf{W}^*$ can indeed perform the above check, even if the game is input unconstrained. □

The rest of the proof is as before. □

## 7.2 Public Immunization

The statement below is the equivalent of Theorem 3 for input-unconstrained games.

**Theorem 6.** *Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a deterministic cryptographic scheme with $\mathcal{R} = \{0,1\}^m$, and $\mathsf{S}$ be uniform over $\{0,1\}^\ell$. Consider any* input-unconstrained, single-instance *game $\mathbf{G} = (\mathsf{C}, \gamma)$ for $\Pi$. Then, for any $c^* > 4$, the immunizer $\Psi^{\mathsf{R},\mathsf{S}}_{k,1}[\mathcal{G}]$ of Fig. 3 is $(t_\mathsf{A}, t_\mathsf{W}, c^*, \epsilon^*)$-subversion-resistant for the* pub-*model with an* online *watchdog, as long as $\mathcal{G} := \{g_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^\ell}$ is a family of $(\frac{k}{n} \to \frac{m-d}{m}, t_{\text{cond}}, \epsilon_{\text{cond}})$-seed-dependent condensers and $\Pi$ is either $(t, \epsilon)$-secure (in case of unpredictability games) or $(t, \epsilon)$-*square-*secure (in case of indistinguishability games) w.r.t. game $\mathbf{G}$, for parameters $t_{\text{cond}}, t \approx t_\mathsf{A}$, $t_\mathsf{W} \approx t_\mathsf{A}$, and*

$$\epsilon \le \begin{cases} \frac{c^*-1}{c^*} \cdot \frac{\epsilon^*}{2^{2d}} - \frac{2\epsilon_{\text{cond}}}{2^{2d}} & \text{if } \mathbf{G} \text{ is an unpredictability game} \\ \left(\frac{c^*-1}{c^*} \cdot \epsilon^* - 2\epsilon_{\text{cond}}\right)^2 \cdot \frac{1}{2^{2d}} & \text{if } \mathbf{G} \text{ is an indistinguishability game.} \end{cases}$$

## 7.3 Transparent Immunization

The statement below is the equivalent of Theorem 4 for input-unconstrained games.

**Theorem 7.** *Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a deterministic cryptographic scheme with $\mathcal{R} = \{0,1\}^m$, and $\mathsf{S}$ be uniform over $\{0,1\}^{m'}$. Consider any* input-unconstrained, single-instance *game $\mathbf{G} = (\mathsf{C}, \gamma)$ for $\Pi$. Then, for any $c^* > 4$, the immunizer $\Psi^{\mathsf{R},\mathsf{S}}_{k,k'}[\mathcal{G}]$ of Fig. 3 is $(t_\mathsf{A}, t_\mathsf{W}, c^*, \epsilon^*)$-subversion-resistant for the* trans-*model with an* online *watchdog, as long as $\mathcal{G} := \{g_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^\ell}$ is a family of $(\frac{k}{n} \to \frac{m-d}{m}, k', t_{\text{cond}}, \epsilon_{\text{cond}})$-seed-dependent condensers with weak seeds and $\Pi$ is either $(t, \epsilon)$-secure w.r.t. game $\mathbf{G}$ (in case of unpredictability games) or $(t, \epsilon)$-square-secure w.r.t. game $\mathbf{G}$ (in case of indistinguishability games), for parameters $t_{\text{cond}}, t, t_\mathsf{W} \approx t_\mathsf{A}$, and*

$$\epsilon \le \begin{cases} \frac{c^*-1}{c^*} \cdot \frac{\epsilon^*}{2^{2d}} - \frac{2\epsilon_{\text{cond}}}{2^{2d}} & \text{if } \mathbf{G} \text{ is an unpredictability game} \\ \left(\frac{c^*-1}{c^*} \cdot \epsilon^* - 2\epsilon_{\text{cond}}\right)^2 \cdot \frac{1}{2^{2d}} & \text{if } \mathbf{G} \text{ is an indistinguishability game.} \end{cases}$$

## 8 Immunization in the Random Oracle Model

In this section we show how to use random oracles to build immunizers for any deterministic cryptographic primitive. As we will show, the random oracle assumption allows to immunize any polynomially-secure deterministic cryptoscheme (w.r.t. both unpredictability and indistinguishability games, and without the need for square security), even in case the random oracle itself is subject to subversion.

We start by defining a notion of subversion-resilient immunizers that is specifically tailored for the ROM, in §8.2. In §8.3 we describe our generic immunizer, and we analyze its security in §8.4.

### 8.1 Random Oracle Model

In the ROM, a cryptographic scheme might be parametrized by a family $\mathcal{H} := \{h : \{0,1\}^* \to \{0,1\}^*\}$ of hash functions. At the beginning of the security game for $\Pi$, a random hash function $h \xleftarrow{\$} \mathcal{H}$ is sampled and all algorithms (including the adversary) are allowed to query the random oracle. Additionally, all probabilities are taken over the choice of the random oracle.

As usual, the only way for the adversary $\mathsf{A}$ to learn the output of $h$ upon some input value $x \in \{0,1\}^*$ is to query the random oracle on $x$.

### 8.2 Subversion-Secure Immunization in the ROM

Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a deterministic cryptographic scheme. An immunizer for $\Pi$ in the ROM is a transformation $\Psi[\mathcal{H}]$, parametrized by a family of hash functions $\mathcal{H} = \{h : \{0,1\}^* \to \{0,1\}^*\}$. Given a particular hash function $h \in \mathcal{H}$, we write $\widehat{\Pi}^h := \Psi^h(\Pi) := (\widehat{\mathsf{K}}^h, \widehat{\mathsf{R}}^h, \widehat{\mathsf{F}}_1^h, \ldots, \widehat{\mathsf{F}}_N^h)$ for the immunized version of the scheme $\Pi$ using random oracle $h$.

We require an immunizer $\Psi$ to satisfy the two properties defined next. Intuitively, the first property says that the immunized scheme $\widehat{\Pi}^h$ meets the same correctness condition as $\Pi$. The second property, instead, establish the security of the immunized scheme against a powerful, "malicious but proud", adversary that attempts to provide its own specification of the immunized algorithms in an undetectable manner.

**Definition 11** (Subversion security in the ROM)**.** Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a cryptographic scheme, and $\mathbf{G} = (\mathsf{C}, \gamma)$ be a security game for $\Pi$. For a set of hash functions $\mathcal{H} = \{h : \{0,1\}^* \to \{0,1\}^*\}$, we say that an immunizer $\Psi[\mathcal{H}]$ is subversion-resistant in the ROM if the following holds: There exists a PPT watchdog $\mathsf{W}$ such that for all PPT adversaries $\mathsf{A} := (\mathsf{A}_0, \mathsf{A}_1)$ there is a negligible function $\nu_{\text{sub}} : \mathbb{N} \to [0,1]$ and a polynomial $p_{\text{det}}(\lambda)$, for which *at least* one of the following conditions hold:

$$\left|\mathbb{P}\left[\mathbf{G}^{\text{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, 0) = 1\right] - \mathbb{P}\left[\mathbf{G}^{\text{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, 1) = 1\right]\right| \geq 1/p_{\text{det}}(\lambda), \tag{20}$$

$$\left|\mathbb{P}\left[\mathbf{G}^{\text{sub}}_{\mathsf{A},\mathsf{C}}(\lambda) = 1\right] - \gamma\right| \leq \nu_{\text{sub}}(\lambda), \tag{21}$$

where the games $\mathbf{G}^{\text{sub}}_{\mathsf{A},\mathsf{C}}(\lambda)$ and $\mathbf{G}^{\text{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, b)$ are depicted in Fig. 7, and the probability is taken over the randomness of $\Pi$, over the coin tosses of $\mathsf{A}$ and $\mathsf{W}$, and over the choice of the random oracle $h \xleftarrow{\$} \mathcal{H}$. The values in the auxiliary information $\mathtt{aux} := (\widetilde{\rho}, \widetilde{\Pi}^h, \widetilde{\mathsf{H}}^h)$ are taken from $\mathbf{G}^{\text{sub}}_{\mathsf{A},\mathsf{C}}(\lambda)$. Moreover, for all $h \in \mathcal{H}$, we require that $\widehat{\Pi}^h = \Psi^h(\Pi)$ meets the same correctness requirement as that of $\Pi$.

The above definition is a slight variant of a similar definition considered in [29]; the difference is that our version is specifically tailored for the ROM, and that we consider security of

```
Game $\mathbf{G}^{\mathrm{sub}}_{\mathsf{A},\mathsf{C}}(\lambda)$
$(\widetilde{\rho}, \widetilde{\Pi}^h, \widetilde{\mathsf{H}}^h, \alpha) \overset{\$}{\leftarrow} \mathsf{A}^h_0(1^\lambda)$
$(d, \widetilde{\tau}) \overset{\$}{\leftarrow} (\mathsf{A}^h_1(1^\lambda, \alpha) \leftrightarrows \mathsf{C}^{\widetilde{\mathsf{H}}^h, \widetilde{\Pi}^h}(1^\lambda, \widetilde{\rho}))$
return $d$
```

```
Game $\mathbf{G}^{\mathrm{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, b)$   $\mathbf{G}^{\mathrm{det\text{-}on}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, b)$
$\mathtt{aux} := (\widetilde{\rho}, \widetilde{\Pi}^h, \widetilde{\mathsf{H}}^h)$
$\mathtt{aux} := (\mathcal{Q}_h, \widetilde{\tau}, \widetilde{\rho}, \widetilde{\Pi}^h, \widetilde{\mathsf{H}}^h)$
if $b = 0$
  return $\mathsf{W}^{\widetilde{\Pi}^h, \widetilde{\mathsf{H}}^h}(1^\lambda, \widetilde{\rho}, \langle \Pi, \Psi \rangle)$
  $\mathsf{W}^{\widetilde{\Pi}^h, \widetilde{\mathsf{H}}^h}(1^\lambda, \widetilde{\rho}, \langle \Pi, \Psi \rangle, \widetilde{\tau}, \mathcal{Q}_h)$
elseif $b = 1$
  $\widehat{\Pi}^h = \Psi^h(\Pi)$
  return $\mathsf{W}^{\widehat{\Pi}^h, h}(1^\lambda, \widetilde{\rho}, \langle \Pi, \Psi \rangle)$
  $\mathsf{W}^{\widehat{\Pi}^h, h}(1^\lambda, \widetilde{\rho}, \langle \Pi, \Psi \rangle, \widetilde{\tau}, \mathcal{Q}_h)$
fi
```

Figure 7: Games defining subversion security of an immunizer $\Psi[\mathcal{H}]$ in the ROM, for both the offline and the online watchdog model.

immunizers and not of primitives directly. Note that, since the adversary is allowed to specify its own implementation of all algorithms, the subversion game is parametrized only by the challenger $\mathsf{C}$ underlying the original game $\mathbf{G}$ for $\Pi$ (and by the adversary $\mathsf{A}$).

Eq. (21) requires that no PPT adversary can break security of $\widehat{\Pi}^h = \Psi^h(\Pi)$ with more than a negligible probability, even if it is able to replace all algorithms in $\widehat{\Pi}^h$ (including the random oracle) with its own malicious specification. We also note that, in contrast to the subversion model of §3, we allow for the adversary to "cook up" the public parameters for $\widehat{\Pi}^h$; this is motivated by the fact that, in some cases, specific values of the public parameters may be decided a priori and included, *e.g.*, in cryptographic standards. (This was the case, for instance, for the celebrated DUAL_EC PRG incident [11].)

On the other hand, Eq. (20) says that the watchdog can, by black-box testing, distinguish the immunized primitive from the implementation specified by the adversary with some non-negligible probability. Note that the watchdog is always given the (malicious) public parameters specified by the adversary in the subversion game; moreover, it always posseses (a description of) the original specification of the cryptographic scheme $\Pi$ and of the compiler $\Psi$.

**Offline versus online watchdogs.** The discussion of §3.3 also applies in the ROM, and we refer the reader to Fig. 7 for a formal description of the games describing subversion-secure immunization in the ROM with an online watchdog. We write $\mathcal{Q}_h$ for the list of random oracle queries asked by the adversary during the subversion game; note that this list is passed to the watchdog in the detection game.

## 8.3 Immunizer Description

We refer the reader to Fig. 8 for a formal description of our immunizer in the ROM. On a high level, the output of the public parameters generator $\mathsf{P}$ is sanitized via a hash function $h_1 \in \mathcal{H}_1$; similarly, the output of the randomness generator $\mathsf{R}$ is sanitized via another hash function $h_2 \in \mathcal{H}_2$. Note that in the ROM both hash functions can be generated by a single random oracle $h \xleftarrow{\$} \mathcal{H}$, by letting $h_1(\cdot) := h(0||\cdot)$ and $h_2(\cdot) := h(1||\cdot)$.

## 8.4 Security Analysis

Next, we analyze the security of the immunizer from Fig. 8 in the ROM, both for the online and offline watchdog models. Our analysis encompasses all deterministic cryptographic schemes.

### 8.4.1 Online Watchdog Model

The theorem below states the security of the immunizer from Fig. 8 in the ROM (cf. Fig. 7).

**Theorem 8.** *Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a deterministic cryptographic scheme, and consider any game $\mathbf{G} = (\mathsf{C}, \gamma)$ such that $\Pi$ is secure w.r.t. game $\mathbf{G}$. Then, the immunizer $\Psi[\mathcal{H}_1, \mathcal{H}_2]$ of Fig. 8 is subversion-resistant in the ROM, for the* online *watchdog model.*

*Proof.* By contradiction, assume that the immunizer $\Psi[\mathcal{H}_1, \mathcal{H}_2]$ is not subversion resistant in the ROM, *i.e.* there exists a PPT adversary $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ such that for all PPT watchdogs $\mathsf{W}$ there exist a negligible function $\nu_{\text{det}} : \mathbb{N} \to [0, 1]$ and a polynomial $p_{\text{sub}}(\lambda)$ for which

$$\left|\mathbb{P}\left[\mathbf{G}^{\text{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, 0) = 1\right] - \mathbb{P}\left[\mathbf{G}^{\text{det}}_{\Pi,\Psi,\mathsf{W}}(\lambda, \mathtt{aux}, 1) = 1\right]\right| \leq \nu_{\text{det}}(\lambda) \tag{22}$$

$$\text{and} \qquad \left|\mathbb{P}\left[\mathbf{G}^{\text{sub}}_{\mathsf{A},\mathsf{C}}(\lambda) = 1\right] - \gamma\right| \geq 1/p_{\text{sub}}(\lambda). \tag{23}$$

The proof goes by introducing the following hybrid games. We refer the reader to Fig. 9 for a description of the games in pseudocode.

**$\mathbf{G}_1(\lambda)$:** This is identical to the game $\mathbf{G}^{\text{sub}}_{\mathsf{A},\mathsf{C}}(\lambda)$, defining subversion security of an immunizer (cf. Fig. 7). Here, the adversary $\mathsf{A}$ specifies the modified public parameters $\widetilde{\rho}$, as well as its own implementation of all other immunized algorithms as in $\widetilde{\Pi}^{h_1,h_2} = (\widetilde{\mathsf{K}}^{h_1,h_2}, \widetilde{\mathsf{R}}^{h_1,h_2}, \widetilde{\mathsf{F}}_1^{h_1,h_2}, \ldots, \widetilde{\mathsf{F}}_N^{h_1,h_2})$, and of the random oracles (via algorithms $\widetilde{\mathsf{H}}_1^{h_1,h_2}$ and $\widetilde{\mathsf{H}}_2^{h_1,h_2}$). Except for these differences, the interaction between $\mathsf{A}$ and $\mathsf{C}$ is identical to that in the original game $\mathbf{G}$.

**$\mathbf{G}_2(\lambda)$:** Identical to the previous game, except that we replace the adversarial implementation of the random oracle (*i.e.*, the algorithms $\widetilde{\mathsf{H}}_1^{h_1,h_2}$ and $\widetilde{\mathsf{H}}_2^{h_1,h_2}$), with the random functions $h_1, h_2$; we also replace the algorithms $\widetilde{\mathsf{K}}^{h_1,h_2}, \widetilde{\mathsf{F}}_1^{h_1,h_2}, \ldots, \widetilde{\mathsf{F}}_N^{h_1,h_2}$ with their immunized counterparts $\widehat{\mathsf{K}}^{h_1,h_2}, \widehat{\mathsf{F}}_1^{h_1,h_2}, \ldots, \widehat{\mathsf{F}}_N^{h_1,h_2}$.

**Subversion-Resistant Immunizer $\Psi^{h_1,h_2}(\cdot)$:**

Let $\Pi = (\mathsf{P}, \mathsf{K}, \mathsf{R}, \mathsf{F}_1, \ldots, \mathsf{F}_N)$ be a cryptographic scheme, and define $\widehat{\Pi}^{h_1,h_2} := \Psi^{h_1,h_2}(\Pi) := (\widehat{\mathsf{K}}^{h_1,h_2}, \widehat{\mathsf{R}}, \widehat{\mathsf{F}}_1^{h_2}, \ldots, \widehat{\mathsf{F}}_N^{h_2})$ as follows:

- Algorithm $\widehat{\mathsf{R}}$: Upon input $1^\lambda$, return $r$ such that $r \xleftarrow{\$} \mathsf{R}(1^\lambda)$.
- Algorithm $\widehat{\mathsf{K}}^{h_1,h_2}$: Upon input $(1^\lambda, \widetilde{\rho}, r)$, return $(pk, sk) = \mathsf{K}(1^\lambda, h_2(\widetilde{\rho}); h_1(r))$.
- Algorithm $\widehat{\mathsf{F}}_i^{h_2}$ (for $i \in [N]$): Upon input $(1^\lambda, \widetilde{\rho}, (pk, sk), x)$, return $y = \mathsf{F}_i(1^\lambda, h_2(\widetilde{\rho}), (pk, sk), x)$.

Figure 8: Description of our subversion resistant immunizer in the ROM, parametrized by hash functions $h_1 \in \mathcal{H}_1$ and $h_2 \in \mathcal{H}_2$.

```
$\mathbf{G}_1(\lambda)$
$h_1, h_2 \xleftarrow{\$} \mathcal{H}_1 \times \mathcal{H}_2$
$(\widetilde{\rho}, \widetilde{\Pi}^{h_1,h_2}, \widetilde{\mathsf{H}}_1^{h_1}, \widetilde{\mathsf{H}}_2^{h_2}, \alpha) \xleftarrow{\$} \mathsf{A}_0^{h_1,h_2}(1^\lambda)$
$d \xleftarrow{\$} \langle \mathsf{A}_1^{h_1,h_2}(1^\lambda, \alpha), \mathsf{C}^{\widetilde{\Pi}^{h_1,h_2}, \widetilde{\mathsf{H}}_1^{h_1,h_2}, \widetilde{\mathsf{H}}_2^{h_1,h_2}}(1^\lambda, \widetilde{\rho})\rangle$
return $d$
```

```
$\mathbf{G}_2(\lambda)$
$h_1, h_2 \xleftarrow{\$} \mathcal{H}_1 \times \mathcal{H}_2$
$(\widetilde{\rho}, \widetilde{\Pi}^{h_1,h_2}, \widetilde{\mathsf{H}}_1^{h_1}, \widetilde{\mathsf{H}}_2^{h_2}, \alpha) \xleftarrow{\$} \mathsf{A}_0^{h_1,h_2}(1^\lambda)$
$d \xleftarrow{\$} \langle \mathsf{A}_1^{h_1,h_2}(1^\lambda, \alpha), \mathsf{C}^{\widehat{\mathsf{K}}^{h_1,h_2}, \widetilde{\mathsf{R}}^{h_1,h_2}, \widehat{\mathsf{F}}_1^{h_2}, \ldots, \widehat{\mathsf{F}}_N^{h_2}, h_1, h_2}(1^\lambda, \widetilde{\rho})\rangle$
return $d$
```

Figure 9: Game hops in the proof of Theorem 8.

We proceed to show that the above games are all computationally indistinguishable, and thus $\mathsf{A}$'s advantage does not vanishes across the different games.

**Lemma 11.** $|\mathbb{P}[\mathbf{G}_1(\lambda) = 1] - \mathbb{P}[\mathbf{G}_2(\lambda) = 1]| \leq \delta^*$.

*Proof.* The proof is similar to that of Lemma 10, hence we only sketch the main differences. Consider the same events $E_{\text{pub}}$, $E_{\text{kgen}}$, and $E_{\text{func}}$ as defined in the proof of Lemma 10. Additionally define the following events:

$$\text{Event } E^1_{\text{ro}}\text{: } \exists j \in [q_1] \text{ s.t. } \widetilde{\mathsf{H}}_1^{h_1,h_2}(u^1_j) = \widetilde{v}^1_j \neq v^1_j = h_1(u^1_j)$$
$$\text{Event } E^2_{\text{ro}}\text{: } \exists j \in [q_2] \text{ s.t. } \widetilde{\mathsf{H}}_2^{h_1,h_2}(u^2_j) = \widetilde{v}^2_j \neq v^2_j = h_2(u^2_j),$$

where $q_1, q_2 \in \mathsf{poly}(\lambda)$ are, respectively, upper bounds on the number of queries asked by $\mathsf{A}$ to its own malicious specification of the first and second random oracle, and $(u^1_j, \widetilde{v}^1_j), (u^2_j, \widetilde{v}^2_j) \in \mathcal{Q}_{h_1,h_2}$.

Define $E_{\text{bad}} := E_{\text{pub}} \vee E_{\text{kgen}} \vee E_{\text{func}} \vee E^1_{\text{ro}} \vee E^2_{\text{ro}}$. Clearly, if $E_{\text{bad}}$ does not happen, we have that $\mathbf{G}_1(\lambda)$ and $\mathbf{G}_2(\lambda)$ are identically distributed, and thus it suffices to prove that event $E_{\text{bad}}$ only happens with negligible probability. By contradiction, assume that $\mathsf{A}$ provokes event $E_{\text{bad}}$ with non-negligible probability. We construct an efficient online watchdog $\mathsf{W}^*$ that distinguishes between $\mathbf{G}^{\text{det-on}}_{\Pi,\Psi,\mathsf{W}^*}(\lambda, \mathtt{aux}, 0)$ and $\mathbf{G}^{\text{det-on}}_{\Pi,\Psi,\mathsf{W}^*}(\lambda, \mathtt{aux}, 1)$ with the same probability, which contradicts Eq. (22).

The watchdog $\mathsf{W}^*$ is identical to the one constructed in the proof of Lemma 10, except that the watchdog now additionally looks at the list of random oracle queries $\mathcal{Q}_{h_1,h_2}$ asked by the adversary $\mathsf{A}$ in game $\mathbf{G}^{\text{sub}}_{\mathsf{A},\mathsf{C}}(\lambda)$, it finds an index $j \in [q_1]$ such that $\mathsf{O}^1_{\text{ro}}(u^1_j) \neq \widetilde{v}^1_j$ or $\mathsf{O}^2_{\text{ro}}(u^2_j) \neq \widetilde{v}^2_j$, and outputs 1 in case such an index is found; here, $\mathsf{O}^1_{\text{ro}}(\cdot)$ and $\mathsf{O}^2_{\text{ro}}(\cdot)$ are the target oracles corresponding to the first and second random oracle, *i.e.* either $\mathsf{O}^1_{\text{ro}} \equiv \widetilde{\mathsf{H}}_1^{h_1,h_2}$ and $\mathsf{O}^2_{\text{ro}} \equiv \widetilde{\mathsf{H}}_2^{h_1,h_2}$ (if $b = 0$), or $\mathsf{O}^1_{\text{ro}} \equiv h_1$ and $\mathsf{O}^2_{\text{ro}} \equiv h_2$ (if $b = 1$). Recall that in the ROM the watchdog is provided with the list $\mathcal{Q}_{h_1,h_2}$ of random oracle queries asked by the adversary in the subversion game, and thus it can indeed run the above comparison. The rest of the proof is unchanged. $\square$

**Lemma 12.** *There exists a polynomial $p_2(\lambda)$ such that*

$$\frac{1}{p_2(\lambda)} |\mathbb{P}[\mathbf{G}_2(\lambda) = 1] - \gamma| \leq |\mathbb{P}[\mathbf{G}_{\Pi,\mathsf{A},\mathsf{C}}(\lambda) = 1] - \gamma| .$$

*Proof.* We build an attacker $\mathsf{A}^*$ that breaks security of the original game $\mathbf{G} = (\mathsf{C}, \gamma)$ for $\Pi$, by making black-box usage of an adversary $\mathsf{A}$ playing game $\mathbf{G}_2(\lambda)$. An important observation here is that the original security game $\mathbf{G}_{\Pi,\mathsf{A},\mathsf{C}}(\lambda)$ and the game for subversion security $\mathbf{G}^{\mathrm{sub}}_{\mathsf{A},\mathsf{C}}(\lambda)$ have the same structure, meaning that $\mathsf{A}$'s queries are syntactically valid in both games (except for random oracle queries, that are not allowed in the original game).[26] A description of $\mathsf{A}^*$ follows.

Adversary $\mathsf{A}^*$:

1. At the beginning, receive the target public parameters $\rho^*$ sampled by the challenger $\mathsf{C}$ in an execution of game $\mathbf{G}$. Hence, initialize $\mathsf{A}(1^\lambda)$.
2. Sample a random bit $\beta \xleftarrow{\$} \{0,1\}$ and a random index $j^* \xleftarrow{\$} [q_2]$.
3. Upon input a query $u^1_j$ to the first random oracle, first check whether $u^1_j$ was already asked; if that is the case, return the corresponding value $v^1_j$, and else return a fresh value $v^1_j \xleftarrow{\$} \{0,1\}^*$.
4. Upon input a query $u^2_j$ to the second random oracle, first check if the corresponding output $v^2_j$ is already defined, and in case it is, return such a value. Else:
   - If $\beta = 0$ always return a fresh value $v^2_j \xleftarrow{\$} \{0,1\}^*$, except for the case $j = j^*$ in which case the random oracle is programmed as in $h_2(u^2_{j^*}) := \rho^*$, and the value $\rho^*$ is returned.
   - If $\beta = 1$, always return a fresh value $v^2_j \xleftarrow{\$} \{0,1\}^*$.
5. Whenever $\mathsf{A}$ outputs $(\widetilde{\rho}, \widetilde{\mathsf{K}}^{h_1,h_2}, \widetilde{\mathsf{R}}^{h_1,h_2}, \widetilde{\mathsf{F}}^{h_1,h_2}_1, \ldots, \widetilde{\mathsf{F}}^{h_1,h_2}_N)$, adversary $\mathsf{A}^*$ acts as follows:
   - If $\beta = 1$, define $h_2(\widetilde{\rho}) := \rho^*$;
   - If $\beta = 0$, define $h_2(\widetilde{\rho}) := \rho'$ for a random $\rho' \xleftarrow{\$} \{0,1\}^*$.
6. Any other query from $\mathsf{A}$ is forwarded to $\mathsf{C}$, and the corresponding answer is sent back, until the game terminates. While doing this, random oracle queries are answered as explained above.
7. At the end of the simulation, check the following conditions:
   - If $r \in \mathcal{Q}_{h_1}$, where $r \xleftarrow{\$} \widetilde{\mathsf{R}}^{h_1,h_2}(1^\lambda)$, output $\perp_1$.
   - If $\beta = 0$ and $\widetilde{\rho} \neq v^2_{j^*}$, output $\perp_{2,0}$.
   - If $\beta = 1$ and $\widetilde{\rho} \notin \mathcal{Q}_{h_2}$, output $\perp_{2,1}$.

Let $E_{\mathrm{bad}} := E_1 \vee E_{2,0} \vee E_{2,1}$ be the event that the reduction aborts, where $E_1$, $E_{2,0}$ and $E_{2,1}$ denote, respectively, the events that the reduction outputs $\perp_1$, $\perp_{2,0}$, and $\perp_{2,1}$; note that this happens whenever $\mathsf{A}^*$ fails either to match the target public parameters $\rho^*$ with $\widetilde{\rho}$, or because $\mathsf{A}$ predicts the output of the subverted randomness source. Below, we bound the probability of each of these events happening.

**Event** $E_1$**:** We claim that $\mathbb{P}[E_1] \in \mathsf{negl}(\lambda)$. Otherwise, there is a simple online watchdog $\mathsf{W}_{\mathrm{rgen}}$ that distinguishes between $\mathbf{G}^{\text{det-on}}_{\Pi,\Psi,\mathsf{W}_{\mathrm{rgen}}}(\lambda, \mathsf{aux}, 0)$ and $\mathbf{G}^{\text{det-on}}_{\Pi,\Psi,\mathsf{W}_{\mathrm{rgen}}}(\lambda, \mathsf{aux}, 1)$ by comparing the outcome of its target oracle for the random source $\mathsf{O}_{\mathrm{rgen}}$ with the input values in $\mathcal{Q}_{h_1}$; such a watchdog is successful with overwhelming probability (since the probability of a collision with the values in $\mathcal{Q}_{h_1}$ is negligible in case $\mathsf{O}_{\mathrm{rgen}} \equiv \mathsf{R}$), which contradicts Eq. (22).

[26]For simplicity we assume that the original scheme is secure in the standard model (*i.e.*, without random oracles); yet, it is easy to adapt the proof to cover also schemes $\Pi$ that are secure in the ROM to start with.

**Event** $E_{2,0}$**:** Define $\mu := \mathbb{P}[\widetilde{\rho} \in \mathcal{Q}_{h_2}]$. We claim that $\mathbb{P}[E_{2,0}] = \frac{1}{2}(1 - \mu/q_2)$, where $q_2 \in \mathsf{poly}(\lambda)$ is an upper bound on the number of queries to the second random oracle. In fact,

$$\begin{aligned}\mathbb{P}[E_{2,0}] &= \mathbb{P}\left[\beta = 0 \wedge \widetilde{\rho} \neq u^2_{j^*}\right] = \mathbb{P}[\beta = 0] \cdot \mathbb{P}\left[\widetilde{\rho} \neq u^2_{j^*}\right] \\ &= \frac{1}{2}\left(\mathbb{P}\left[\widetilde{\rho} \neq u^2_{j^*} \mid \widetilde{\rho} \in \mathcal{Q}_{h_2}\right] \cdot \mathbb{P}[\widetilde{\rho} \in \mathcal{Q}_{h_1}] + \mathbb{P}\left[\widetilde{\rho} \neq u^2_{j^*} \mid \widetilde{\rho} \notin \mathcal{Q}_{h_2}\right] \cdot \mathbb{P}[\widetilde{\rho} \notin \mathcal{Q}_{h_2}]\right) \\ &= \frac{1}{2}\left(\left(1 - \frac{1}{q_2}\right) \cdot \mu + (1 - \mu)\right) = \frac{1}{2} \cdot \left(1 - \frac{\mu}{q_2}\right).\end{aligned}$$

**Event** $E_{2,1}$**:** By definition, we have:

$$\Pr[E_{2,1}] = \mathbb{P}[\beta = 1 \wedge \widetilde{\rho} \in \mathcal{Q}_{h_2}] = \mathbb{P}[\beta = 1] \cdot \mathbb{P}[\widetilde{\rho} \in \mathcal{Q}_{h_2}] = \mu/2.$$

By the union bound, there exists a negligible function $\nu_2 : \mathbb{N} \to [0, 1]$ such that:

$$\begin{aligned}\mathbb{P}[E_{\text{bad}}] \leq \mathbb{P}[E_1] + \mathbb{P}[E_{2,0}] + \mathbb{P}[E_{2,1}] &= \frac{1}{2}\left(1 - \frac{\mu}{q_2} + \mu\right) + \mathbb{P}[E_{2,1}] \\ &= \frac{1}{2}\left(1 + \mu\left(1 - \frac{1}{q_2}\right)\right) + \mathbb{P}[E_{2,1}] \leq 1 - \frac{1}{2q_2} + \nu_2(\lambda).\end{aligned}$$

Finally, note that $\mathbb{P}[\mathbf{G}_{\Pi,\mathsf{A}^*,\mathsf{C}}(\lambda) = 1] = \mathbb{P}[\mathbf{G}_{\Pi,\mathsf{A}^*,\mathsf{C}}(\lambda) = 1 \mid \neg E_{\text{bad}}] \cdot \mathbb{P}[\neg E_{\text{bad}}]$, and by the above calculation $\mathbb{P}[\neg E_{\text{bad}}] \geq 1/p_2(\lambda)$ for some polynomial $p_2(\lambda)$. It remains to analyze $\mathbb{P}[\mathbf{G}_{\Pi,\mathsf{A}^*,\mathsf{C}}(\lambda) = 1 \mid \neg E_{\text{bad}}]$. Let $r^* \overset{⚄}{\leftarrow} \mathsf{R}(1^\lambda)$ and $(pk^*, sk^*) := \mathsf{K}(1^\lambda, \rho^*; r^*)$ be the randomness and the keys generated by the challenger $\mathsf{C}$. Conditioned on event $E_{\text{bad}}$ not happening, we have that $h_2(\widetilde{\rho}) = \rho^*$, and moreover the distribution of $h_2(r)$, where $r \overset{⚄}{\leftarrow} \widetilde{\mathsf{R}}^{h_1,h_2}(1^\lambda)$, is identical to that of $r^*$ (since $r$ is never queried to the random oracle). Moreover, the view of adversary $\mathsf{A}$ is perfectly simulated, as

$$\begin{aligned}\forall r \overset{⚄}{\leftarrow} \widetilde{\mathsf{R}}^{h_1,h_2}(1^\lambda) :\ & \widehat{\mathsf{K}}^{h_1,h_2}(1^\lambda, \widetilde{\rho}; r) = \mathsf{K}(1^\lambda, h_2(\widetilde{\rho}); h_1(r)) = \mathsf{K}(1^\lambda, \rho^*; r^*) = (pk^*, sk^*) \\ \forall i \in [N], x \in \widetilde{\tau} :\ & \widehat{\mathsf{F}}_i^{h_2}(1^\lambda, \widetilde{\rho}, (pk^*, sk^*), x) = \mathsf{F}_i(1^\lambda, h_2(\widetilde{\rho}), (pk^*, sk^*), x) = \mathsf{F}_i(1^\lambda, \rho^*, (pk^*, sk^*), x),\end{aligned}$$

and thus $\mathbb{P}[\mathbf{G}_{\Pi,\mathsf{A},\mathsf{C}}(\lambda) = 1 \mid \neg E_{\text{bad}}] = \mathbb{P}[\mathbf{G}_2(\lambda) = 1]$ as desired.

□

Considering the previous lemmas, and applying the triangle inequality, we obtain that

$$|\mathbb{P}[\mathbf{G}_0(\lambda) = 1] - \gamma| \geq |\mathbb{P}[\mathbf{G}_2(\lambda) = 1] - \gamma| - \nu_1(\lambda).$$

Together with Eq. (23), this yields

$$|\mathbb{P}[\mathbf{G}_{\Pi,\mathsf{A},\mathsf{C}}(\lambda) = 1] - \gamma| \geq \frac{1}{p_2(\lambda)} \cdot \frac{1}{p_{\text{sub}}(\lambda)} - \nu(\lambda),$$

for some negligible function $\nu : \mathbb{N} \to [0, 1]$. As the above quantity is non negligible, this contradicts the security of the original scheme $\Pi$ w.r.t. game $\mathbf{G}$, and thus finishes the proof of the theorem. □

### 8.4.2 Offline Watchdog Model

We do not know how to make the above proof work for input constrained games in the offline watchdog model. The reason is that, in order to program the random oracles as described in the last step of the proof, we must reach a hybrid where the subverted implementation of the random oracles $\widetilde{\mathsf{H}}_1^{h_1,h_2}$ and $\widetilde{\mathsf{H}}_2^{h_1,h_2}$ is replaced with the real random oracles $h_1$ and $h_2$. However, it seems hard to make this transition without requiring that the watchdog is given the list of random oracle queries $\mathcal{Q}_{h_1,h_2}$ asked by the adversary in the subversion game. Unfortunately, this list should not be available in the offline watchdog model.

We note that the above observation does not seem to effect the proof given in [29, Theorem 3.5], as their argument does not explicitly take into account subversion of the random oracles.

## 9 Conclusions

We have shown how to immunize a large class of *deterministic* cryptographic primitives against complete subversion, meaning that the adversary is allowed to tamper with all the underlying algorithms, and with the immunizer itself. In the random oracle model, there is a simple immunizer that relies on a single secret, but tamperable, source of randomness [29, 30]. In the standard model, instead, we need to assume an additional independent public, and in some case untamperable, random source.

Open problems include, *e.g.*, finding better immunizers, both in terms of computational assumptions and/or the number of assumed trusted random sources. Also, exploring alternative approaches in order to achieve subversion security in the plain model for larger classes of cryptographic schemes (*e.g.*, indistinguishability applications without square security, or randomized ones), while still relying on $O(1)$ independent random sources, is an interesting direction for future research.

## Acknowledgements

The third author was supported by the Concordium Blockchain Research Center, Aarhus University, Denmark.

## References

[1] Giuseppe Ateniese, Danilo Francati, Bernardo Magri, and Daniele Venturi. Public immunization against complete subversion without random oracles. In *ACNS*, pages 465–485, 2019.

[2] Giuseppe Ateniese, Aggelos Kiayias, Bernardo Magri, Yiannis Tselekounis, and Daniele Venturi. Secure outsourcing of cryptographic circuits manufacturing. In Joonsang Baek, Willy Susilo, and Jongkil Kim, editors, *Provable Security - 12th International Conference, ProvSec 2018*. Springer, 2018.

[3] Giuseppe Ateniese, Bernardo Magri, and Daniele Venturi. Subversion-resilient signature schemes. In *CCS*, pages 364–375, 2015.

[4] Benedikt Auerbach, Mihir Bellare, and Eike Kiltz. Public-key encryption resistant to parameter subversion and its realization from efficiently-embeddable groups. In *PKC*, pages 348–377, 2018.

[5] James Ball, Julian Borger, and Glenn Greenwald. Revealed: How US and UK spy agencies defeat internet privacy and security. *Guardian Weekly*, September 2013.

[6] Boaz Barak, Yevgeniy Dodis, Hugo Krawczyk, Olivier Pereira, Krzysztof Pietrzak, François-Xavier Standaert, and Yu Yu. Leftover hash lemma, revisited. In *CRYPTO*, pages 1–20, 2011.

[7] Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. NIZKs with an untrusted CRS: Security in the face of parameter subversion. In *ASIACRYPT*, pages 777–804, 2016.

[8] Mihir Bellare, Joseph Jaeger, and Daniel Kane. Mass-surveillance without the state: Strongly undetectable algorithm-substitution attacks. In *CCS*, pages 1431–1440, 2015.

[9] Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. Security of symmetric encryption against mass surveillance. In *CRYPTO*, pages 1–19, 2014.

[10] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS*, pages 62–73, 1993.

[11] Stephen Checkoway, Ruben Niederhagen, Adam Everspaugh, Matthew Green, Tanja Lange, Thomas Ristenpart, Daniel J. Bernstein, Jake Maskiewicz, Hovav Shacham, and Matthew Fredrikson. On the practical exploitability of dual EC in TLS implementations. In *USENIX Security Symposium*, pages 319–335, 2014.

[12] Sherman S. M. Chow, Alexander Russell, Qiang Tang, Moti Yung, Yongjun Zhao, and Hong-Sheng Zhou. Let a non-barking watchdog bite: Cliptographic signatures with an offline watchdog. In *PKC*, pages 221–251, 2019.

[13] Dana Dachman-Soled, Rosario Gennaro, Hugo Krawczyk, and Tal Malkin. Computational extractors and pseudorandomness. In *TCC*, pages 383–403, 2012.

[14] Jean Paul Degabriele, Pooya Farshim, and Bertram Poettering. A more cautious approach to security against mass surveillance. In *FSE*, pages 579–598, 2015.

[15] Jean Paul Degabriele, Kenneth G. Paterson, Jacob C. N. Schuldt, and Joanne Woodage. Backdoors in pseudorandom number generators: Possibility and impossibility results. In *CRYPTO*, pages 403–432, 2016.

[16] Yevgeniy Dodis, Chaya Ganesh, Alexander Golovnev, Ari Juels, and Thomas Ristenpart. A formal treatment of backdoored pseudorandom generators. In *EUROCRYPT*, pages 101–126, 2015.

[17] Yevgeniy Dodis, Ilya Mironov, and Noah Stephens-Davidowitz. Message transmission with reverse firewalls - Secure communication on corrupted machines. In *CRYPTO*, pages 341–372, 2016.

[18] Yevgeniy Dodis, Thomas Ristenpart, and Salil P. Vadhan. Randomness condensers for efficiently samplable, seed-dependent sources. In *TCC*, pages 618–635, 2012.

[19] Yevgeniy Dodis and Yu Yu. Overcoming weak expectations. In *TCC*, pages 1–22, 2013.

[20] Marc Fischlin, Christian Janson, and Sogol Mazaheri. Backdoored hash functions: Immunizing HMAC and HKDF. In *IEEE Computer Security Foundations Symposium*, pages 105–118, 2018.

[21] Marc Fischlin and Sogol Mazaheri. Self-guarding cryptographic protocols against algorithm substitution attacks. In *IEEE Computer Security Foundations Symposium*, pages 76–90, 2018.

[22] Chaya Ganesh, Bernardo Magri, and Daniele Venturi. Cryptographic reverse firewalls for interactive proof systems. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[23] Glenn Greenwald. No place to hide: Edward Snowden, the NSA, and the U.S. surveillance state. *Metropolitan Books*, May 2014.

[24] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *Journal of the ACM (JACM)*, 56(4):20, 2009.

[25] Nicholas J. Hopper, Luis von Ahn, and John Langford. Provably secure steganography. *IEEE Trans. Computers*, 58(5):662–676, 2009.

[26] Ilya Mironov and Noah Stephens-Davidowitz. Cryptographic reverse firewalls. In *EUROCRYPT*, pages 657–686, 2015.

[27] Nicole Perlroth, Jeff Larson, and Scott Shane. N.S.A. able to foil basic safeguards of privacy on web. *The New York Times*, September 2013.

[28] Phillip Rogaway. The moral character of cryptographic work. *IACR Cryptology ePrint Archive*, 2015:1162, 2015.

[29] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. Cliptography: Clipping the power of kleptographic attacks. In *ASIACRYPT*, pages 34–64, 2016.

[30] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. Generic semantic security against a kleptographic adversary. In *ACM CCS*, pages 907–922, 2017.

[31] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. Correcting subverted random oracles. In *CRYPTO*, pages 241–271, 2018.

[32] Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In *CRYPTO*, pages 51–67, 1983.

[33] Gustavus J. Simmons. The subliminal channel and digital signature. In *EUROCRYPT*, pages 364–378, 1984.

[34] Luca Trevisan and Salil P. Vadhan. Extracting randomness from samplable distributions. In *FOCS*, pages 32–42, 2000.

[35] Adam L. Young and Moti Yung. The dark side of "black-box" cryptography, or: Should we trust Capstone? In *CRYPTO*, pages 89–103, 1996.

[36] Adam L. Young and Moti Yung. Kleptography: Using cryptography against cryptography. In *EUROCRYPT*, pages 62–74, 1997.