Computational Robust (Fuzzy) Extractors for CRS-dependent Sources with Minimal Min-entropy

Hanwen Feng[1] and Qiang Tang[2]✉

[1] Alibaba Group, fenghanwen.fhw@alibaba-inc.com
[2] The University of Sydney, qiang.tang@sydney.edu.au

Abstract. Robust (fuzzy) extractors are very useful for, e.g., authenticated key exchange from a shared weak secret and remote biometric authentication against active adversaries. They enable two parties to extract the same uniform randomness with a "helper" string. More importantly, they have an authentication mechanism built in that tampering of the "helper" string will be detected. Unfortunately, as shown by Dodis and Wichs, in the information-theoretic setting, a robust extractor for an (n, k)-source requires $k > n/2$, which is in sharp contrast with randomness extractors which only require $k = \omega(\log n)$. Existing works either rely on random oracles or introduce CRS and work only for CRS-independent sources (even in the computational setting).

In this work, we give a systematic study about robust (fuzzy) extractors for general CRS *dependent* sources. We show in the information-theoretic setting, the same entropy lower bound holds even in the CRS model; we then show we *can* have robust extractors in the computational setting for general CRS-dependent source that is only with minimal entropy. We further extend our construction to robust fuzzy extractors. Along the way, we propose a new primitive called κ-MAC, which is unforgeable with a weak key and hides all partial information about the key (both against auxiliary input); it may be of independent interests.

1 Introduction

Randomness extractors are well-studied tools that enable one to extract uniform randomness (usually with the help of a short random seed) from a weak random source with sufficient entropy. Robust (fuzzy) extractors, which are randomness extractors that can be against an active attacker, are very useful in the settings of authenticated key exchange (AKE) from shared weak secrets and remote biometric authentication. Sometimes, a one-message AKE protocol from weak secrets is directly known as a robust extractor (for close secrets, a robust fuzzy extractor) [4,7,9,10,19,22]. Informally, a robust extractor consists of a generation algorithm Gen producing a nearly-uniform string R along with a public helper string P

(message sent in public) from a source W, and a reproduction algorithm Rep recovering R from P and W. Besides the normal requirement as a randomness extractor that the extracted R should be uniform, the *robustness* ensures that any manipulation on P by active attackers will be detected. Furthermore, for composition with other applications that will use the extracted randomness, stronger robustness (called *post-application robustness*) is usually required, by allowing adversaries to have R directly, which ensures the security even after adversaries learning information about R from applications using R.

Robust extractors turn out to be expensive. It is known that information-theoretic robust extractors require the (min-)entropy k of the source $W \in \{0,1\}^n$ to be larger than $n/2$ [10, 12], which is in contrast with regular randomness extractors that only require a minimal entropy $\omega(\log n)$ from the source. Naturally, leveraging a random oracle as a "super" randomness extractor could circumvent this entropy lower bound. Indeed, one can directly hash a source (with a minimal entropy like $\omega(\log n)$) for this purpose. Moreover, one can also transform a fuzzy extractor [3, 11] into a robust fuzzy extractor [4]. However, it is always desirable to see whether we can remove this heuristic assumption [6], particularly in the setting of randomness extraction.

The other approach uses a common reference string (CRS), which could be generated by a trusted third party once and for all. It enables us to transform a strong extractor into a robust extractor by using the CRS as the seed. Clearly, this approach will not require more entropy from the source than the underlying extractor. It also can be extended to the fuzzy setting [7, 19, 20, 22]. However, as the seed has to be independent of the source, this approach so far only works for CRS-independent sources.

In many cases, sources could be dependent on the CRS. For example, for sources generated from devices such as PUFs, adversaries might manufacture the devices after seeing the CRS and insert some CRS-dependent backdoor into the device to gain advantages. More seriously, for all sources, given a CRS-dependent leakage (which is possible as the leakage function is adversarially chosen after seeing the CRS), the distribution of the remained secret will be dependent on the CRS as well. We are interested in the following natural open question:

Can we have a robust (fuzzy) extractor that works for general CRS-dependent sources with minimal min-entropy ($\omega(\log n)$) without relying on an RO? [3]

[3] For the non-fuzzy case, Dodis *et al.* [9] presented a partial solution in the computational setting. However, their construction only works for a very special source: the

Our results. We systematically investigate this question, in both computational and information-theoretic settings, for both non-fuzzy and fuzzy cases. All related results are summarized in Table 1.

Fuzzy?	Schemes	Model	CRS-Dependent?	IT/Computational?	Low Entropy Rate?	General Sources?
Non	Naive-RO	RO	-	Computational	$\surd$	$\surd$
	[10]	Plain	-	IT	$\times$	$\surd$
	Naive-CRS	CRS	$\times$	IT	$\surd$	$\times$
	[9]	CRS	$\surd$	Computational	$\times$	$\times$
	Ours(Sect.5)	CRS	$\surd$	Computational	$\surd$	$\surd$
Fuzzy	[4]	RO	-	Computational	$\surd$	$\surd$
	[10, 16]	Plain	-	IT	$\times$	$\surd$
	[7]	CRS	$\times$	IT	$\surd$	$\times$
	[19, 20, 22]	CRS	$\times$	Computational	$\surd$	$\times$
	Ours (Sect.6)	CRS	$\surd$	Computational	$\surd$	$\surd$

Table 1. Comparison between known robust (fuzzy) extractors. "Low Entropy-Rate?" asks whether the scheme works for (n, k)-sources with $k = \omega(\log n)$; "General Sources?" asks whether the scheme works for sources without other requirements beyond that on (n, k) (so CRS-independent ones are all not general). "Naive-RO" denotes the trivial construction that extracts randomness $H(w)$ using a random oracle H; "Naive-CRS" denotes a strong extractor using the CRS as the seed.

Lower-bound in the information-theoretic setting. We first give a negative answer in the information-theoretic setting by proving that the lower bound for plain-model constructions [12] also holds in CRS-dependent constructions. Namely, if there is a CRS-model information-theoretically-secure (IT-secure) *pre-application* robust extractor working for every source $W \in \{0, 1\}^n$ that has min-entropy greater than k even conditioned on the CRS (we refer such a source an (n, k)-source), it must be that $k > n/2$. This new lower bound justifies the necessity of the CRS-independent requirement in existing CRS-model IT-secure robust (fuzzy) extractors [7].

A generic construction of computational CRS-model robust extractors. We then consider circumventing our new lower bound in the computational setting. We present a generic construction of CRS-dependent *post-application* robust extractors and thus firmly confirm its existence. This construction is built upon a conventional randomness extractor and a novel message authentication code (MAC) termed by key-private auxiliary-input MAC (κ-MAC for short) for which we give efficient constructions

sample consists of (w, c) where c is a ciphertext that probabilistically encrypts 0s under w; they further require the source to have any linear fraction of min-entropy. In comparison, we are aiming for general sources that only have minimal super logarithmic entropy. For the fuzzy case, there is no feasibility result at all.

from well-studied assumptions. Our construction works for any efficiently samplable sources that have sufficient min-entropy (conditional on CRS) just to admit a conventional randomness extractor.

An extended construction for robust fuzzy extractors. We further extend our solution and construct a computational CRS-dependent robust *fuzzy* extractor by using a conventional randomness extractor, a secure sketch, and a stronger κ-MAC that can work in the fuzzy setting. Here, a q-secure sketch is a tool allowing one to convert a weak secret W' to a q-close one W with the help of a small amount of information about W, which is the core of many fuzzy extractors and has IT-secure instantiations.

For achieving error tolerance t, (namely, two close secrets W and W' whose distance is within t), our construction requires the source to support a $2t$-secure sketch [4]. This requirement indeed matches the requirement made by many existing CRS-model robust fuzzy extractors [19,20], while our construction is the first one working for CRS-dependent sources.

Our techniques. We give a technical overview as follows.

Proving lower-bounds for CRS-model IT-secure robust extractor. Our main technique for the generalized lower bound is to show that a CRS-model IT-secure robust extractor implies a plain-model IT-secure "authentication scheme", which was the main tool for showing the lower bound of entropy rate [12].

Note that a CRS-model robust extractor for all (n, k)-sources trivially implies a CRS-model "authentication scheme" $\{\mathsf{Auth}, \mathsf{Vrfy}\}$: Auth runs the generation algorithm Gen and outputs the helper string P as an "authentication tag" ς; Vrfy runs Rep on input P and outputs 1 unless Rep fails. For any (n, k)-source W and any unbounded adversary $\mathcal{A}$, the scheme is correct and unforgeable w.r.t. a randomly sampled $\mathtt{crs}$ according to the CRS distribution CRS. To show a CRS-model "authentication scheme" gives a plain model one: we prove that there exist at least one concrete CRS string $\mathtt{crs}^*$ such that it will enable "correct" authentication *and* "unforgeability" for *all* CRS-dependent sources.

For unforgeability, assume that the advantage of any adversary forging a tag in the CRS-model scheme is bounded by δ. First, we show that, for each source W, any adversary $\mathcal{A}$, and any constants $c_0, c_1 \in (0, 1)$, there will be a good set $S_{W,\mathcal{A}}$ with weight at least c_0 (namely, $\Pr[\mathsf{CRS} \in S_{W,\mathcal{A}}] \geq c_0$) such that for every $\mathtt{crs} \in S_{W,\mathcal{A}}$, the advantage of $\mathcal{A}$ forging a valid tag for W is bounded by δ/c_0.

[4] Note that secure sketches achieving t error tolerance are also subject to some entropy-rate lower-bounds [14]. However, for almost all error-rate t/n (except a small range), the bound is notably smaller than $1/2$.

Note that the above discussions give a "locally good" set for each W, but we need a "globally good" set of CRSs for all sources and all adversaries. For any $\mathcal{A}$, we show that, $\widehat{S}_{\mathcal{A}}$, the intersection of $\{S_{W,\mathcal{A}}\}$ for all sources W, is with weight at least c_0; for every $\mathtt{crs} \in \widehat{S}_{\mathcal{A}}$, $\mathcal{A}$'s advantage is bounded by δ/c_0. We proceed with proof by contradiction: if not, its complement $\widehat{S}_{\mathcal{A}}^C$ will have the weight of at least $(1-c_0)$. By definition, for every $\mathtt{crs}^{(i)} \in \widehat{S}_{\mathcal{A}}^C$, there is one source W (whose conditional distribution is $W_{\mathtt{crs}}^{(i)}$) s.t. $\mathcal{A}$ has advantage greater than δ/c_0. We can define a "new" (n,k)-source $W^* = \{W|_{\mathtt{crs}}\}$ where $W|_{\mathtt{crs}^{(i)}} = W_{\mathtt{crs}}^{(i)}$ if $\mathtt{crs}^i \in \widehat{S}_{\mathcal{A}}^C$ and uniform otherwise. For such W^* and $\mathcal{A}$, there is no good $S_{W^*,\mathcal{A}}$ with weight greater than c_0, which contradicts our previous argument. Finally, we can prove $\bigcap_{\mathcal{A}} \widehat{S}_{\mathcal{A}}$ is globally good, as otherwise, we can "construct" an adversary $\mathcal{A}^*$ contradicting the existence of $\widehat{S}_{\mathcal{A}^*}$.

By similar arguments, we can show there is a globally good CRS set $\widetilde{S}$ for correctness as well. Then by adequately choosing c_0 and c_1, the sum weight of $\widehat{S}$ and $\widetilde{S}$ can be greater than 1, thus there exists a $\mathtt{crs}^*$ which is globally good for both correctness and unforgeability. Hardcoded with this string $\mathtt{crs}^*$, the CRS-model authentication scheme gives a plain-model authentication scheme.

Adding post-application robustness *to randomness extractor for "free".* We then turn to computational setting. In a conventional strong extractor Ext (which converts a weak secret w into a uniform r with the help of a uniform seed s), we may view the seed as the "helper string". To make it robust, we could let the "helper string" additionally include a MAC tag for the seed such that adversaries cannot malleate it without being detected. One might want to use r as the key, but the verifier will not have r until receiving s, which leads to circularity. We consider taking w as the MAC key directly.

We can see that a normal MAC will be insufficient. On the one hand, the secret w is non-uniform, especially when we consider post-application robustness, the randomness r and the seed s together give non-trivial information about w and will be leaked to adversaries. On the other hand, the authentication tag itself may contain information about w, which in turn affects the quality of randomness extraction.

We, therefore, introduce a new MAC called κ-MAC. Besides unforgeability, it satisfies *key privacy*, that is, adversaries cannot learn anything new about the key from an authentication tag. Thus, the authentication tag will not affect the randomness extraction (in the computational setting). Moreover, both unforgeability and key privacy should hold even when adversaries have arbitrary admissible auxiliary information about the secret, making this primitive co-exist with (r, s). We define κ-MAC in the CRS model and allow the distribution of secrets to be arbitrarily dependent on the CRS, as long as it is efficiently samplable and has sufficient min-entropy (conditioned on the CRS). We remark that a *one-time* κ-MAC suffices for constructing robust extractors.

κ-MAC from sLRH relation. It is natural to view κ-MAC as a special leakage-resilient (more precisely, auxiliary-input secure) MAC; then upgrade it to add key privacy. The known approach to auxiliary-input MAC is using the auxiliary-input signature in the symmetric setting by taking both verification key vk and signing key sk as the MAC key k. But in κ-MAC, k is just a non-uniform string sampled from the source, which may not have a structure like (vk, sk); we have to deal with it carefully.

We revisit Katz-Vaikuntanathan signature [17] that is shown to be auxiliary-input secure [13]. On rough terms, they used a true-simulation-extractable NIZK (tSE-NIZK) [8] to prove the knowledge of a witness k^* w.r.t. a statement y (contained in the verification key), such that (k^*, y) satisfy a *leakage-resilient hard* (LRH) relation. In an LRH relation, for honest generated (y, k), and given y and leakage about k, it is infeasible to find a witness of y. If there is a successful forgery, we can extract k^* for y (by tSE-NIZK), which contradicts the LRH relation.

For our κ-MAC, we take the signing key sk as the authentication key k, but vk cannot be posted on a trusted bulletin board, as in signatures, or be in k as the source might not be structured. We address this challenge as follows. First, there is a part of vk (denoted by $\mathtt{pp}$) that can be generated without k and reused across users, and we put it in the CRS. For the other part (denoted by yk), while adversaries can manipulate it, we strengthen the LRH relation to ensure this manipulation will not give advantages. Specifically, we define the strengthened LRH relation (sLRH relation): given honestly generated $(\mathtt{pp}, yk)$ along with leakage about k, adversaries cannot find a (yk', k') such that both $(\mathtt{pp}, yk', k')$ and $(\mathtt{pp}, yk', k)$ satisfy the sLRH relation. This strengthening is sufficient, since using tSE-NIZK to prove knowledge of k w.r.t. $(\mathtt{pp}, yk)$ and attaching yk (and the proof) to the authentication tag could give an auxiliary-input MAC from weak secrets. Here, the verifier algorithm checks whether $(\mathtt{pp}, yk', k)$ satisfies this relation and whether π is valid, and a forgery violates either the sLRH relation or tSE-NIZK.

For *key privacy*, we need yk to hide partial information about k, i.e., one can simulate the yk distribution without k. Accordingly, we formulate the privacy of generators for a sLRH relation. With a sLRH relation and its generator satisfying privacy (called a private generator), we have a κ-MAC construction in this way.

Constructing sLRH relation from DPKE+NIZK. The privacy of generator indeed prevents adversaries from finding k from (pp, yk) and the leakage. If it further has a kind of "collision-resistance", namely, even when k is given, it is infeasible to find a distinct k' along with yk' such that both (pp, yk', k) and (pp, yk', k') belong to R_{LR}, R_{LR} with a private generator will be a sLRH relation. Specifically, consider an adversary that outputs (yk', k') and breaks the sLRH relation; if $k = k'$, it contradicts the privacy of generator; otherwise, it violates this "collision-resistance".

We use an auxiliary-input-secure deterministic encryption scheme to instantiate an NP relation R_{de} with a private generator. Specifically, $(pk, c, m) \in R_{\mathsf{de}}$ iff $c = \mathsf{DEnc}(pk, m)$. From the security of DPKE, (pk, c) could hide partial information about m. For handling all hard-to-invert auxilairy information, the DPKE scheme from exponentially hard DDH assumption [24] will be the only choice.

Note that pk has to be a part of yk (not pp) since DPKE only works for message distributions independent of pk, and we need work for CRS-dependent sources. Now, the adversary can replace pk with a "bad" pk' such that $(pk', c' = \mathsf{DEnc}(pk', m))$ cannot uniquely determine the message m; so this relation (together with its private generator) is not a sLRH relation. To get around this obstacle, we let yk include a NIZK proof π (besides (pk, c)) demonstrating that pk defines an injection $\mathsf{DEnc}(pk, \cdot)$. Though NIZK needs a CRS as well, it is secure even when statements and witnesses are dependent on the CRS.

Extending to the fuzzy case. Finally, we extend our solutions to the fuzzy case. The starting point is using κ-MAC to authenticate the helper string of a fuzzy extractor. We take the standard secure-sketch-based fuzzy extractor as a building block, in which one can recover the secret w using his secret w' first.

The κ-MAC we just defined will be insufficient for the fuzzy case. Adversaries may manipulate the helper string, such that one recovers another secret w'' (which is t-close to w') that a forged tag can be verified under w''. We therefore need κ-MAC to satisfy *fuzzy unforgeability*, that is, given an authentication tag from w, adversaries cannot forge an authentication tag being accepted by any string close to w. Note that the distance between w'' and w is bounded by $2t$, the fuzzy unforgeability should prevent from a forgery w.r.t. any $2t$-close secret.

To construct a fuzzy unforgeable κ-MAC, we first introduce a fuzzy version of sLRH relation. More specifically, for a $2t$-fuzzy sLRH relation, it is infeasible to find (yk', k') to "frame" any secret k^* which is $2t$-close to k. It is easy to verify the according κ-MAC satisfies $2t$-fuzzy unforgeability.

Interestingly, we do not need other tools to construct a fuzzy sLRH relation. Our construction of sLRH relation is fuzzy already. Particularly, if a sLRH relation is "collision-resistant", the adversary can "frame" some k'' only when she exactly finds k''. It remains to argue that, given $(\mathtt{pp}, yk)$ from a private generator on input k and the leakage about k, *can adversaries find a secret k'' that is $2t$-close to k?*

This question seems straightforward at first glance but turns out to need some care. Note that the privacy of generator cannot ensure that $(\mathtt{pp}, yk)$ hides *all* partial information about k, as $(\mathtt{pp}, yk)$ itself must be non-trivial about k. A safe way to check whether a value can be recovered from $(\mathtt{pp}, yk)$ is to see whether this value is useful for distinguishing yk and $\widehat{yk}$; anything can be used to distinguish cannot be recovered. For small t (say, logarithmic in the security parameter), one knowing $k'' \in B_{2t}(k)$ can guess the original k with a non-negligible probability, and then she can use k to distinguish. The situation gets complicated when t is large and $B_{2t}(k)$ has exponentially many points. In this case, one cannot naively guess k according to k''. We overcome this challenge by observing the task of recovering k from k'' can be done with the help of $2t$-secure sketch. More specifically, assume an adversary can recover k'' from $(\mathtt{pp}, yk)$. Then, the distinguisher specifies the leakage as a $2t$-secure sketch, invokes the adversary to have this $k'' \in B_{2t}(k)$, and converts k'' to k with the help of the secure sketch. Usually, auxiliary inputs are considered a "bad" object to be against, but our proof leverages the auxiliary input to get around barriers of security proof.

2 Preliminaries

Notations. All adversaries considered in this paper are non-uniform, and we model an adversary $\mathcal{A}$ by a family of circuits $\{A_\lambda\}_{n\in\mathbb{N}}$. For a set $\mathbb{X}$, $x \leftarrow\!\!\$\, \mathbb{X}$ denotes sampling x from the uniform distribution over $\mathbb{X}$. For a distribution X, $x \leftarrow X$ denotes sampling x from X. Let (X, Y) be a joint distribution, $X|_y$ denotes the conditional distribution of X conditioned on $Y = y$.

Min-entropy. The min-entropy of a distribution W is defined by $\mathbf{H}_\infty(W) = -\min_{w\in\mathsf{Supp}(W)} \log \Pr[W = w]$. We say W has min-entropy of $\widehat{k}$ conditioned on Z, if $\mathbf{H}_\infty(W|_z) \geq \widehat{k}$ for every $\mathsf{z} \in \mathsf{Supp}(Z)$.

Strong extractor. Let n, k, ℓ be integer functions of the security parameter. An (n, k, ℓ) strong randomness extractor Ext is a deterministic algorithm, which on inputs $w \in \{0,1\}^{n(\lambda)}$ along with a public seed i_{ext} (with length $si(\lambda)$) outputs another randomenss $r \in \{0,1\}^{\ell(\lambda)}$. Ext satisfies ϵ-privacy, if for any polynomial-time $\mathcal{A}$ and any (n,k)-sources $\mathcal{W}$, it holds that $\mathsf{Adv}^{\mathrm{ext}}_{\mathcal{W},\mathcal{A}}(\lambda) \leq \epsilon(\lambda)$, where $\mathsf{Adv}^{\mathrm{ext}}_{\mathcal{W},\mathcal{A}}(\lambda)$ is defined as

$$\left| \Pr\begin{bmatrix} w \leftarrow W_\lambda, i_{\mathsf{ext}} \leftarrow\!\$\, \{0,1\}^{si(\lambda)} \\ r \leftarrow \mathsf{Ext}(i_{\mathsf{ext}}, w) : \\ 1 \leftarrow \mathcal{A}(i_{\mathsf{ext}}, r) \end{bmatrix} - \Pr\begin{bmatrix} w \leftarrow W_\lambda, i_{\mathsf{ext}} \leftarrow\!\$\, \{0,1\}^{si(\lambda)} \\ r \leftarrow\!\$\, \{0,1\}^{(\ell(\lambda))} : \\ 1 \leftarrow \mathcal{A}(i_{\mathsf{ext}}, r) \end{bmatrix} \right|.$$

Metric spaces. A metric space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda \in \mathbb{N}}$ is a collection of sets with a distance function $\mathsf{dist} : \mathcal{M}_\lambda \times \mathcal{M}_\lambda \to [0,\infty)$. Throught this paper we consider $\mathcal{M}_\lambda = \{0,1\}^{n(\lambda)}$ equipped with a distance function (e.g., Hamming distance).

Secure sketch. Let $\mathcal{M}$ be a metric space. An $(\mathcal{M}, k, k', t)$-secure sketch scheme is a pair of PPT algorithms SS and Rec that satisfies correctness and security. For every $\lambda \in \mathbb{N}$, SS on input $w \in \mathcal{M}_\lambda$, outputs a sketch ss; Rec takes as inputs a sketch ss and $\widetilde{w} \in \mathcal{M}_\lambda$, and ouputs w'.
Correctness. $\forall \widetilde{w} \in \mathcal{M}_\lambda$, if $\mathsf{dist}(w, \widetilde{w}) \leq t(\lambda)$, then $\mathsf{Rec}(\widetilde{w}, \mathsf{SS}(w)) = w$.
Security. For every λ, any distribution W over $\mathcal{M}_\lambda$ with min-entropy at least $k(\lambda)$, it holds that $H_\infty(W|\mathsf{SS}(W)) \geq k'(\lambda)$.

We may abbreviate an $(\mathcal{M}, k, k', t)$-secure sketch by t-secure sketch without specifying other parameters.

NIZK. A non-interactive zero-knowledge proof system (NIZK) Π for an NP relation R can be described by the following three algorithms. $\mathsf{Setup}(1^\lambda)$ generates a CRS $\mathtt{crs}$; $\mathsf{Prove}(\mathtt{crs}, x, \psi)$ takes as inputs a CRS $\mathtt{crs}$, a statement x and a witness ψ, and outputs a proof π; $\mathsf{Verify}(\mathtt{crs}, x, \pi)$ checks the validity of π.

Π satisfies the *perfect completeness*, if for any $\lambda \in \mathbb{N}$ and for any $(x, \psi) \in R$,

$$\Pr[\mathtt{crs} \leftarrow \mathsf{Setup}(1^\lambda); \pi \leftarrow \mathsf{Prove}(\mathtt{crs}, x, \psi) : \mathsf{Verify}(\mathtt{crs}, x, \pi) = 1] = 1.$$

Π satisfies ϵ_{snd}-*adaptive soundness*, if for any polynomial-time adversary $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathrm{snd}}_{\mathcal{A}}(\lambda) \leq \epsilon_{\mathsf{snd}}(\lambda)$, where $\mathsf{Adv}^{\mathrm{snd}}_{\mathcal{A}}(\lambda)$ is defined as

$$\Pr[\mathtt{crs} \leftarrow \mathsf{Setup}(1^\lambda); (x, \pi) \leftarrow \mathcal{A}(\mathtt{crs}) : \mathsf{Verify}(\sigma, x, \pi) = 1 \wedge (\forall \psi, (x, \psi) \notin R)].$$

For zero-knowledgenss, we introduce the single theorem version, which sufficies for our applications. Namely, we say Π satisfies ϵ_{zk}-ZK, if there exists a simulator $(\mathsf{SimSetup}, \mathsf{SimProve})$, such that for any polynomial-time $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, it holds that $\mathsf{Adv}^{\mathrm{zk}}_{\mathcal{A}}(\lambda) \leq \epsilon_{\mathsf{zk}}(\lambda)$, where $\mathsf{Adv}^{\mathrm{zk}}_{\mathcal{A}}(\lambda)$ is defined as

$$\left| \Pr\begin{bmatrix} \texttt{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ (x, \psi, st) \leftarrow \mathcal{A}_1(\texttt{crs}) \\ \pi \leftarrow \mathsf{Prove}(\texttt{crs}, x, \psi) : \\ 1 \leftarrow \mathcal{A}_2(st, \pi) \end{bmatrix} - \Pr\begin{bmatrix} (\texttt{crs}, \texttt{tk}) \leftarrow \mathsf{SimSetup}(1^\lambda) \\ (x, \psi, st) \leftarrow \mathcal{A}_1(\texttt{crs}) \\ \pi \leftarrow \mathsf{SimProve}(\texttt{crs}, \texttt{tk}, x) : \\ 1 \leftarrow \mathcal{A}_2(st, \pi) \end{bmatrix} \right| .$$

Furthermore, we will need a strengthened soundness termed by **true-simulation-extractability (tSE)** [8], which says that any efficient adversary $\mathcal{A}$ cannot produce a valid proof π^* for x^* without knowing x^*'s witness, even $\mathcal{A}$ can see a simulated proof for a valid statement x. Note that a tSE-NIZK is implied by a simulation-extractable NIZK [18] which allows adversaries to see simulated proofs on arbitrary statements, including false statements. Moreover, tSE-NIZK may have more efficient constructions [8].

We now present the single-theorem version. We say $\mathit{\Pi}$ satisfies $(\epsilon_{\mathsf{tse1}}, \epsilon_{\mathsf{tse2}})$-tSE, if there exists a simulation-knowledge extractor $(\mathsf{SESetup}, \mathsf{SimProve}, \mathsf{KExt})$, such that for any polynomial-time adversary $\mathcal{A}$ and $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$, $\mathsf{Adv}^{\mathrm{tse1}}_{\mathcal{A}}(\lambda) \leq \epsilon_{\mathsf{tse1}}(\lambda)$, where $\mathsf{Adv}^{\mathrm{tse1}}_{\mathcal{A}}(\lambda)$ is defined as

$$\left| \Pr\begin{bmatrix} (\texttt{crs}, \texttt{tk}, \texttt{ek}) \leftarrow \mathsf{SESetup}(1^\lambda) : \\ 1 \leftarrow \mathcal{A}(\texttt{crs}, \texttt{tk}) \end{bmatrix} - \Pr\begin{bmatrix} (\texttt{crs}, \texttt{tk}) \leftarrow \mathsf{SimSetup}(1^\lambda) : \\ 1 \leftarrow \mathcal{A}(\texttt{crs}, \texttt{tk}) \end{bmatrix} \right| ,$$

and $\mathsf{Adv}^{\mathrm{tse2}}_{\mathcal{A}}(\lambda) \leq \epsilon_{\mathsf{tse2}}(\lambda)$, where $\mathsf{Adv}^{\mathrm{tse2}}_{\mathcal{A}}(\lambda)$ is defined as

$$\Pr\begin{bmatrix} (\texttt{crs}, \texttt{tk}, \texttt{ek}) \leftarrow \mathsf{SESetup}(1^\lambda), (x, \psi, st) \leftarrow \mathcal{B}_1(\texttt{crs}), \pi \leftarrow \mathsf{SimProve}(\texttt{crs}, \\ \texttt{tk}, x), (x^*, \pi^*) \leftarrow \mathcal{B}_2(st, \pi), w^* \leftarrow \mathsf{KExt}(\texttt{crs}, \texttt{tk}, x^*, \pi^*) : (x^*, w^*) \notin R \end{bmatrix} .$$

Deterministic public-key encryption A deterministic public-key encryption (DPKE) scheme $\mathit{\Sigma}$ is defined by a triple of PPT algorithms $\{\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$ where Enc and Dec are deterministic.

A DPKE scheme $\mathit{\Sigma}$ is $(n, \epsilon_{\mathsf{hv}}, \epsilon_{\mathsf{ind}})$-PRIV-IND-secure [5], if for any message source $\mathcal{W}$ defined over $\{\{0,1\}^{n(\lambda)}\}_{\lambda \in \mathbb{N}}$ and any function ensemble $\mathcal{F} = \{f_\lambda\}_{\lambda \in \mathbb{N}}$ such that $\mathcal{F}$ is ϵ_{hv}-hard-to-invert w.r.t. $\mathcal{W}$, for any polynomial-time adversary $\mathcal{A}$, it follows that $\mathsf{Adv}^{\mathrm{ind}}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda) \leq \epsilon_{\mathsf{ind}}(\lambda)$, where $\mathsf{Adv}^{\text{ind}}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda)$ is defined as

$$\left| \Pr\begin{bmatrix} (pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ m \leftarrow W_\lambda, \\ c \leftarrow \mathsf{Enc}(pk, m): \\ 1 \leftarrow \mathcal{A}(c, pk, f_\lambda(m)) \end{bmatrix} - \Pr\begin{bmatrix} (pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ m \leftarrow W_\lambda, m' \leftarrow\!\$\, \{0,1\}^{n(\lambda)}, \\ c \leftarrow \mathsf{Enc}(pk, m'): \\ 1 \leftarrow \mathcal{A}(c, pk, f_\lambda(m)) \end{bmatrix} \right|.$$

We assume w.l.o.g. that Σ has a key relation R_{pk} s.t. for every $(pk, sk) \in R_{pk}$, it follows that $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$ for any message m.

3 CRS-Model Robust Extractor: Definitions

In this section, we present both information-theoretic and computational definitions of robust extractors in the CRS model.

CRS-dependent sources. Being different from all previous CRS-model works of fuzzy extractors [7,19–22] that require sources to be independent of the CRS, we consider all sources that could potentially depend on the CRS while having sufficient conditional min-entropy. Formally, We model a source $\mathcal{W}$ as an ensemble of distributions $\mathcal{W} = \{W_\lambda\}_{\lambda\in\mathbb{N}}$. Let $\mathsf{CRS} = \{\mathsf{CRS}_\lambda\}_{\lambda\in\mathbb{N}}$ be an ensemble of CRS distributions, and we denote each W_λ by a collection $\{W_\lambda|_{\mathtt{crs}}\}_{\mathtt{crs}\in\mathsf{Supp}(\mathsf{CRS}_\lambda)}$. Here $W_\lambda|_{\mathtt{crs}}$ is used to denote the conditional distribution of W_λ conditioned on that $\mathsf{CRS}_\lambda = \mathtt{crs}$. For a distribution W_λ independent of CRS, it holds that $W_\lambda = W_\lambda|_{\mathtt{crs}}$ for every $\mathtt{crs}$. Moreover, any collection $\{W_{\lambda,\mathtt{crs}}\}_{\mathtt{crs}\in\mathsf{Supp}(\mathsf{CRS}_\lambda)}$ in turn defines a distribution W_λ for which $W_\lambda|_{\mathtt{crs}} = W_{\lambda,\mathtt{crs}}$.

Let n and k be integer functions of the security parameter. For a source $\mathcal{W}$ defined over $\{\{0,1\}^{n(\lambda)}\}_{\lambda\in\mathbb{N}}$, we call it an (n,k)-source (w.r.t. CRS), if for any λ, the distribution W_λ is an $(n(\lambda), k(\lambda))$-distribution (w.r.t. CRS_λ). Namely,

$$\mathbf{H}_\infty(W_\lambda) \geq k(\lambda) \text{ (or for any } \mathtt{crs} \in \mathsf{Supp}(\mathsf{CRS}_\lambda), \mathbf{H}_\infty(W_\lambda|_{\mathtt{crs}}) \geq k(\lambda).)$$

In the computational setting, we further require each W_λ to be efficiently samplable by a polynomial-bounded circuit.

Definition 1 (Efficiently-samplable source w.r.t. CRS). *For a distributions ensembles $\mathsf{CRS} = \{\mathsf{CRS}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{W} = \{W_\lambda\}_{\lambda\in\mathbb{N}}$, we call W_λ an efficiently-samplable distribution w.r.t. CRS_λ, if there is a circuit G_λ whose running time is polynomial in λ, such that for every $\mathtt{crs} \in \mathsf{Supp}(\mathsf{CRS}_\lambda)$, it holds that*

$$G_\lambda(\mathtt{crs}) = W_\lambda|_{\mathtt{crs}}.$$

If for every $\lambda \in \mathbb{N}$, W_λ is an efficiently-samplable distribution w.r.t. CRS_λ*, we call $\mathcal{W}$ an efficiently-samplable source w.r.t.* CRS*.*

Remark 1. We consider efficiently samplable sources in the computational setting, as the dependence between a source being extracted and the CRS distribution is usually caused by an efficient adversary. A typical scenario could be that a non-uniform PPT adversary $\mathcal{A} = \{A_\lambda\}_{\lambda\in\mathbb{N}}$ "creates" a source after seeing the CRS. Therefore, we ask a *uniform* polynomial-bounded circuit G_λ (which can be considered as A_λ) for every $\mathtt{crs} \in \mathsf{Supp}(\mathsf{CRS}_\lambda)$, rather than different polynomial-bounded circuits for different $\mathtt{crs}$. Similar settings appeared in the recent works on two sources extractors [1,15].

Robust extractor. A robust extractor rExt in the CRS-model is defined by a triplet of efficient algorithms $\{\mathsf{CRS}, \mathsf{Gen}, \mathsf{Rep}\}$. CRS is a sampler algorithm that specifies the CRS distribution. Gen takes as inputs a CRS and a weak secret w and outputs a randomness R along with a helper string P. Then, Rep can recover R from P using w. rExt requires *privacy* and *robustness*. The former says R is pseudorandom conditioned on P, and the latter captures the infeasibility of forging a different P that will not lead to the failure of Rep. Particularly, when $\mathcal{A}$ is given both R and P, the robusntess is called *post-application* robustness; when only P is given, it is called *pre-applicaiton* robustness.

Formally, we define a robust extractor below.

Definition 2 (Robust extractor). *For interger functions n, k, ℓ of the security parameter, an (n, k, ℓ)-robust extractor* rExt *is defined by the following PPT algorithms.*

- $\mathtt{crs} \leftarrow \mathsf{CRS}(1^\lambda)$*. On input the security parameter λ, it outputs a CRS* $\mathtt{crs}$*, whose distribution is denoted by* CRS_λ*.*
- $(R, P) \leftarrow \mathsf{Gen}(\mathtt{crs}, w)$*. On inputs* $\mathtt{crs}$ *and a string $w \in \{0,1\}^{n(\lambda)}$, it outputs a randomness $R \in \{0,1\}^{\ell(\lambda)}$ along with a helper string P.*
- $R \leftarrow \mathsf{Rep}(\mathtt{crs}, w, P)$*. It recover the randomness R from P using w.*

Correctness: *For a function $\rho : \mathbb{N} \to [0,1]$, we say* rExt *satisfies ρ-correctness, if for any (n,k)-source $\mathcal{W}$, for every λ, it holds that*

$$\Pr\left[\begin{matrix}\mathtt{crs} \leftarrow \mathsf{CRS}_\lambda; w \leftarrow W_\lambda|_{\mathtt{crs}};\\ (R,P) \leftarrow \mathsf{Gen}(\mathtt{crs}, w) : \mathsf{Rep}(\mathtt{crs}, w, P) = R\end{matrix}\right] \geq \rho(\lambda).$$

Privacy: *For $\epsilon : \mathbb{N} \to (0,1)$,* rExt *satisfies the ϵ-IT-privacy, if for any unbounded adversary $\mathcal{A}$ and any (n,k)-source $\mathcal{W}$, it holds that*

$$\mathsf{Adv}^{\mathsf{priv}}_{\mathcal{A},\mathcal{W}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{priv},0}_{\mathcal{A},\mathcal{W}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{priv},1}_{\mathcal{A},\mathcal{W}}(\lambda) = 1]| \leq \epsilon(\lambda).$$

Robustness: *For $\delta : \mathbb{N} \to (0,1)$,*$\mathsf{rExt}$ *satisfies the δ-IT-post-application-robustness (or pre-application robustness, without boxed items in the experiment $\mathsf{Exp}^{\mathsf{rob}}_{\mathcal{A},\mathcal{W}}$), if for any unbounded adversary $\mathcal{A}$, and any (n,k)-source $\mathcal{W}$, it holds that $\mathsf{Adv}^{\mathsf{rob}}_{\mathcal{A},\mathcal{W}}(\lambda) = \Pr[\mathsf{Exp}^{\mathsf{rob}}_{\mathcal{A},\mathcal{W}}(\lambda) = 1] \leq \delta(\lambda)$.*

```
$\mathsf{Exp}^{\mathsf{priv},b}_{\mathcal{A},\mathcal{W}}(\lambda)$
$\mathtt{crs} \leftarrow \mathsf{CRS}_\lambda; w \leftarrow W_\lambda|_{\mathtt{crs}}; (R,P) \leftarrow \mathsf{Gen}(\mathtt{crs}, w);$
$R_0 \leftarrow\$ \{0,1\}^{\ell(\lambda)}; R_1 = R; b' \leftarrow \mathcal{A}(\mathtt{crs}, P, R_b)$
**return** $b'$

$\mathsf{Exp}^{\mathsf{rob}}_{\mathcal{A},\mathcal{W}}(\lambda)$
$\mathtt{crs} \leftarrow \mathsf{CRS}_\lambda; w \leftarrow W_\lambda|_{\mathtt{crs}}$
$(R,P) \leftarrow \mathsf{Gen}(\mathtt{crs}, w); P^* \leftarrow \mathcal{A}(\mathtt{crs}, P\boxed{, R})$
**if** $P^* \neq P \wedge \mathsf{Rep}(\mathtt{crs}, P^*, w) \neq \perp)$ **then return** $1$
**return** $0$
```

Computational definitions can be defined by only considering polynomial-time adversaries and efficiently-samplable sources. We directly call these computational versions ϵ-privacy and δ-post-application-robustness (by removing "IT").

Robust fuzzy extractor. When the generation algorithm Gen and the reproduction algorithm Rep could use different but close secrets $w, \widetilde{w}$, $\{\mathsf{CRS}, \mathsf{Gen}, \mathsf{Rep}\}$ defines a robust fuzzy extractor. More formally, we require that w and $\widetilde{w}$ are in a metric space $\mathcal{M}$ with a distance function dist. For an integer $\widehat{t}$, we say w is $\widehat{t}$-close to $\widetilde{w}$, if $\mathsf{dist}(w, \widetilde{w}) \leq \widehat{t}$. For $\mathcal{W} = \{W_\lambda\}_{\lambda \in \mathbb{N}}$ and $\widetilde{\mathcal{W}} = \{\widetilde{W_\lambda}\}_{\lambda \in \mathbb{N}}$ defined over $\mathcal{M}$, we say $(\mathcal{W}, \widetilde{\mathcal{W}})$ a t-pair for an integer function t, if for every $\lambda \in \mathbb{N}$ and $\mathtt{crs} \in \mathsf{Supp}(\mathsf{CRS}_\lambda)$, it holds that $\Pr[(w, \widetilde{w}) \leftarrow (W_\lambda|_{\mathtt{crs}}, \widetilde{W_\lambda}|_{\mathtt{crs}}) : \mathsf{dist}(w, \widetilde{w}) \leq t(\lambda)] = 1$. For simplicity, we assume $\mathcal{M}$ is $\{\{0,1\}^{n(\lambda)}\}_{\lambda \in \mathbb{N}}$ equipped with a distance function dist (e.g., Hamming distance).

We call $\mathsf{rfExt} = \{\mathsf{CRS}, \mathsf{Gen}, \mathsf{Rep}\}$ an $(\mathcal{M}, k, \ell, t)$-robust fuzzy extractor, if it satisfies correctness, privacy, and robustness w.r.t. any t-pair of (n,k)-sources $(\mathcal{W}, \widetilde{W})$. Formal definitions are given in the full paper.

4 A New Lower Bound for IT-secure Robust Extractors

As briefly explained in the introduction, a plain-model IT-secure robust extractor for all (n, k)-sources exists only when $k > n/2$ [12]. This lower bound can be trivially circumvented by assuming a CRS and work only for the special sources that are *independent* of the CRS. We are interested in the case for general sources which may be CRS-dependent. This section gives a negative result that IT-secure robust extractors for all (n, k)-sources also require that $k > n/2$ in the CRS setting. The fuzzy case trivially inherits this generalized lower bound.

Previous tool for the plain model lower bound. Dodis and Wichs's [12] lower-bound comes from a plain-model IT-secure authentication scheme (for an-$(\widehat{n}, \widehat{k})$-distribution W), which is trivially implied by an IT-secure robust extractor. Such an authentication scheme could be described by a pair of randomized functions $\{\mathsf{Auth}, \mathsf{Vrfy}\}$, formed by $\mathsf{Auth} : \{0,1\}^{\widehat{n}} \to \{0,1\}^{\widehat{s}}$, and $\mathsf{Vrfy} : \{0,1\}^{\widehat{n}} \times \{0,1\}^{\widehat{s}} \to \{0,1\}$, where $\widehat{n}, \widehat{s}$ are integers. It satisfies (1) $\widehat{\rho}$-correctness: $\Pr[w \leftarrow W : \mathsf{Vrfy}(w, \mathsf{Auth}(w)) = 1] \geq \widehat{\rho}$; and (2) $\widehat{\delta}$-unforgeability: for any adversary $\mathcal{A}$, $\Pr[w \leftarrow W, \varsigma \leftarrow \mathsf{Auth}(w), \varsigma^* \leftarrow \mathcal{A}(\varsigma) : \mathsf{Vrfy}(w, \varsigma^*) = 1] \leq \widehat{\delta}$.

Lemma 1 **([12]).** *If there exists an authentication scheme for all $(\widehat{n}, \widehat{k})$-distributions with $\widehat{\rho}$-correctness and $\widehat{\delta}$-unforgeability, and $\widehat{\delta} < \widehat{\rho}^2/4$, it follows that $\widehat{k} > \widehat{n}/2$.*

Generalizing the lower-bound. We present a new lower bound for the CRS-model in the following theorem; our main technical lemma is to show that a CRS-model authentication scheme could imply that in the plain model (Lemma 2).

Theorem 1. *Let $n, k, \ell : \mathbb{N} \to \mathbb{N}$ and $\rho, \delta : \mathbb{N} \to \{0, 1\}$ be functions of the security parameter. If there exists an (n, k, ℓ) IT-secure robust extractor with ρ-correctness and δ-pre-application-robustness, then for any $\lambda \in \mathbb{N}$ s.t. $\delta(\lambda) \leq \rho(\lambda)^2/4$, it follows that $k(\lambda) > n(\lambda)/2$.*

Proof. We first define a CRS-model authentication scheme, which consists $\{\mathsf{CAuth}, \mathsf{CVrfy}\}$ (randomized) along with a CRS distribution $\widehat{\mathsf{CRS}}$, satisfying the following, for any $(\widehat{n}, \widehat{k})$-source W:

- $\widehat{\rho}$-correctness: $\Pr[\mathtt{crs} \leftarrow \widehat{\mathsf{CRS}}, w \leftarrow W|_{\mathtt{crs}} : \mathsf{Vrfy}(\mathtt{crs}, w, \mathsf{Auth}(\mathtt{crs}, w)) = 1] \geq \widehat{\rho}$.
- $\widehat{\delta}$-unforgeability: for any adversary $\mathcal{A}$,

$$\Pr\left[\begin{array}{r} \mathtt{crs} \leftarrow \widehat{\mathsf{CRS}}, w \leftarrow W|_{\mathtt{crs}}, \varsigma \leftarrow \mathsf{Auth}(\mathtt{crs}, w), \\ \varsigma^* \leftarrow \mathcal{A}(\mathtt{crs}, \varsigma) : \mathsf{Vrfy}(\mathtt{crs}, w, \varsigma^*) = 1. \end{array}\right] \leq \widehat{\delta}.$$

It is easy to see that, if there is a CRS-model IT-secure (n, k, ℓ)-robust extractor $\{\mathsf{CRS}, \mathsf{Gen}, \mathsf{Rep}\}$ with ρ-correctness and δ-robustness, for each $\lambda \in \mathbb{N}$, we can construct $\{\mathsf{CAuth}, \mathsf{CVrfy}\}$ along with a CRS distribution $\widehat{\mathsf{CRS}} = \mathsf{CRS}_\lambda$ that satisfies $\widehat{\rho} = \rho(\lambda)$-correctness and $\widehat{\delta} = \delta(\lambda)$-unforgeability w.r.t all $(n(\lambda), k(\lambda))$-distributions . More detailly,

- $\mathsf{CAuth}(\mathtt{crs}, w)$: Invoke $(R, P) \leftarrow \mathsf{Gen}(\mathtt{crs}, w)$, and return $\sigma = P$;
- $\mathsf{CVrfy}(\mathtt{crs}, w, \sigma)$: If $\mathsf{Rep}(\mathtt{crs}, w, \sigma) =\perp$, return 0; otherwise, return 1.

Next, we give our main technical lemma for the CRS-model authentication scheme, whose detailed proof is deferred later.

Lemma 2. *If there exists a CRS-model IT-secure authentication scheme* $\{\mathsf{CAuth}, \mathsf{CVrfy}\}$ *(along with a CRS distribution* $\widehat{\mathsf{CRS}}$*) for all* $(\widehat{n}, \widehat{k})$*- distributions with* $\widehat{\rho}$*-correctness and* $\widehat{\delta}$*-unforgeability, then for any* $\widehat{c}_0, \widehat{c}_1 \in (0, 1)$ *satisfying* $(1-\widehat{c}_1)\widehat{\rho}+\widehat{c}_0 > 1$*, there exists a plain-model IT-secure authentication scheme* $\{\mathsf{Auth}, \mathsf{Vrfy}\}$ *for all* $(\widehat{n}, \widehat{k})$*-distributions with* $\widehat{c}_1\widehat{\rho}$*-correctness and* $\widehat{\delta}/\widehat{c}_0$*-unforgeability.*

By Lemma 1, if $\widehat{\delta}/\widehat{c}_0 < (\widehat{c}_1\widehat{\rho})^2/4$, $\{\mathsf{Auth}, \mathsf{Vrfy}\}$ established in Lemma 2 exists only when $\widehat{k} > \widehat{n}/2$. Putting requirements together, $\{\mathsf{CAuth}, \mathsf{CVrfy}\}$ with $\widehat{\rho}$-correctness and $\widehat{\delta}$-unforgeability could imply such $\{\mathsf{Auth}, \mathsf{Vrfy}\}$, if there exists $\widehat{c}_0, \widehat{c}_1 \in \{0, 1\}$, such that

$$\widehat{\delta} < \frac{\widehat{c}_0\widehat{c}_1^2\widehat{\rho}^2}{4}, \quad \text{and} \quad (1-\widehat{c}_1)\widehat{\rho}+\widehat{c}_0 > 1. \tag{1}$$

It remains to show when such $(\widehat{c}_0, \widehat{c}_1)$ exist. Note for any $\widehat{\rho} \in (0, 1)$, there always exists $(\widehat{c}_0, \widehat{c}_1) \in (0, 1)^2$ satisfying $(1-\widehat{c}_1)\widehat{\rho}+\widehat{c}_0 > 1$ (denote the solution space by $S_{\widehat{\rho}}$). Then, we can have $(\widehat{c}_0, \widehat{c}_1)$ satisfying Eq.1 for $(\widehat{\rho}, \widehat{\delta})$, unless $\frac{4\widehat{\delta}}{\widehat{\rho}^2} \geq \widehat{c}_0\widehat{c}_1^2$ for any $(\widehat{c}_0, \widehat{c}_1) \in S_{\widehat{\rho}}$.

By standard analysis, we have the following result: for any $\widehat{\rho}, \widehat{v} \in (0, 1)$, there always eixsts $(\widehat{c}_0, \widehat{c}_1) \in S_{\widehat{\rho}}$ such that $\widehat{c}_0\widehat{c}_1^2 > \widehat{v}$. It follows that whenever $\widehat{\delta} < \widehat{\rho}^2/4$, such $(\widehat{c}_0, \widehat{c}_1)$ exist. Recall that for any λ *s.t.* $\delta(\lambda) < \rho(\lambda)^2/4$, the robust extractor could give such $\{\mathsf{CAuth}, \mathsf{CVrfy}\}$ for all $(n(\lambda), k(\lambda))$-distributions. It follows $k(\lambda) < n(\lambda)/2$ in this case. □

Deferred proof for Lemma 2. The over goal is to show there exists a "good" CRS $\mathtt{crs}^*$ in the support of $\widehat{\mathsf{CRS}}$, such that with $\mathtt{crs}^*$ hardcoded, $\{\mathsf{CAuth}(\mathtt{crs}^*, \cdot), \mathsf{CVrfy}(\mathtt{crs}^*, \cdot)\}$ is the plain-model authentication scheme. For both *correctness* and *unforgeability*, we will prove that there exist a sufficiently large "good" set of CRSs ($\mathtt{S}$ and $\widetilde{\mathtt{S}}$) for each of them. Then by properly tuning parameters, we can see $\mathtt{S} \cap \tilde{\mathtt{S}} \neq \emptyset$, thus we can find a string $\mathtt{crs}^*$.

In the claim below, we show the existence of $\mathtt{S}$ (for *correctness*). We proceed in two steps. (i) For each source W and a randomly sampled crs, we have ρ-correctness; then, by simple probabilistic analysis, there must exist a large enough "good" set $\mathtt{S}_W$ that every element of it will enable "correctness" (with a smaller correctness parameter). (ii) To show $\bigcap_W \mathtt{S}_W$ is still with sufficient size, we can use proof by contradiction in a sense that if it does not hold, we can define a special source W^* whose "good" set $\mathtt{S}_{W*}$ will be smaller than that established in the previous step.

Claim. For any constant $\widehat{c}_1 \in (0,1)$, there exists a set $\mathtt{S} \in \mathsf{Supp}(\widehat{\mathsf{CRS}})$ such that $\Pr[\widehat{\mathsf{CRS}} \in \mathtt{S}] \geq (1-\widehat{c}_1)\widehat{\rho}$, and for any $\mathtt{crs} \in \mathtt{S}$ and any $(\hat{n}, \hat{k})$-distribution W, it holds that

$$\Pr\left[w \leftarrow W|_{\mathtt{crs}}, \varsigma \leftarrow \mathsf{CAuth}(\mathtt{crs}, w) : \mathsf{CVrfy}(\mathtt{crs}, w, \varsigma) = 1\right] \geq \widehat{c}_1\widehat{\rho}.$$

Proof (of claim). For convience, we define the "verified correctly" event w.r.t. W and $\mathtt{crs}$:

$$\mathtt{VC}_{W,\mathtt{crs}} := [w \leftarrow W|_{\mathtt{crs}}, \varsigma \leftarrow \mathsf{CAuth}(\mathtt{crs}, w) : \mathsf{CVrfy}(\mathtt{crs}, w, \varsigma) = 1].$$

Then define a "good" set $\mathtt{S}$ for an $(\widehat{n}, \widehat{k})$-distribution W. Namely,

$$\mathtt{S}_W := \{\mathtt{crs} \in \mathsf{Supp}(\mathsf{CRS}) : \Pr[\mathtt{VC}_{W,\mathtt{crs}}] \geq \widehat{c}_1\widehat{\rho}\}. \tag{2}$$

We now show

$$\Pr[\widehat{\mathsf{CRS}} \in \mathtt{S}_W] \geq (1-\widehat{c}_1)\widehat{\rho} \tag{3}$$

for any $(\hat{n}, \hat{k})$-distribution W. If not, for some W, we have the following,

$$\begin{aligned} &\Pr[\mathtt{crs} \leftarrow \widehat{\mathsf{CRS}} : \mathtt{VC}_{W,\mathtt{crs}}] \\ \leq &\Pr[\mathtt{VC}_{W,\mathtt{crs}} | \mathtt{crs} \notin \mathtt{S}_W] \Pr[\widehat{\mathsf{CRS}} \notin \mathtt{S}_W] + \Pr[\widehat{\mathsf{CRS}} \in \mathtt{S}_W] \\ \leq &\widehat{c}_1\widehat{\rho} + (1-\widehat{c}_1)\widehat{\rho} = \widehat{\rho}, \end{aligned}$$

which contradicts the assumption that $\{\mathsf{CAuth}, \mathsf{CVrfy}\}$ along with $\widehat{\mathsf{CRS}}$ satisfies the $\widehat{\rho}$-correctness.

Note that $\mathtt{S}_W$ is a "locally good" set for W, and we need a "globally good" set $\mathtt{S}$ for all $(\widehat{n}, \widehat{k})$-distributions. By definition, $\mathtt{S}$ will be the intersection of all $\mathtt{S}_W$, namely,

$$\mathtt{S} = \bigcap_{\forall (\widehat{n}, \widehat{k})\text{-distribution } W} \mathtt{S}_W.$$

Our goal is to show $\Pr[\widehat{\mathsf{CRS}} \in \mathtt{S}] \geq (1-\widehat{c}_1)\widehat{\rho}$. We proceed it by contradication. Specifically, if not, the complement of $\mathtt{S}$ (denoted by $\mathtt{S}^C$) will satisfy $\Pr[\widehat{\mathsf{CRS}} \in \mathtt{S}^C] > 1-(1-\widehat{c}_1)\widehat{\rho}$. By definition, for every $\mathtt{crs}_i \in \mathtt{S}^C$, there exists a $(\widehat{n},\widehat{k})$-distribution W_i, such that

$$\Pr[\mathtt{VC}_{W_i,\mathtt{crs}_i}] < \widehat{c}_1\widehat{\rho}.$$

Next, we can define a distribution W^* for which the set $\mathtt{S}_{W^*}$ does not sastify Eq.3. Specifically, $W^* = \{W^*|_{\mathtt{crs}_i}\}_{\mathtt{crs}_i \in \mathsf{Supp}(\widehat{\mathsf{CRS}})}$, where

$$W^*|_{\mathtt{crs}_i} = \begin{cases} W_i|_{\mathtt{crs}_i}, \text{ if } \mathtt{crs}_i \in \mathtt{S}^C, \\ U_{\widehat{n}}, \text{ if } \mathtt{crs}_i \in \mathtt{S}. \end{cases} \tag{4}$$

Here $U_{\widehat{n}}$ denotes the uniform distribution over $\{0,1\}^{\widehat{n}}$. It is easy to verify W^* is an $(\widehat{n},\widehat{k})$-distribution. However, from the definition of W^*, it follows that $\mathtt{S}_{W^*} \bigcap \mathtt{S}^C = \emptyset$, and thus $\Pr[\mathsf{CRS} \in \mathtt{S}_{W^*}] < (1-\widehat{c}_1)\widehat{\rho}$, which contradicts the result Eq.3. □

For *unforgeability*, it follows similar idea. We have the following claim whose formal proof is given in the full paper.

Claim. For any constant $\widehat{c}_0 \in (0,1)$, there exists a set $\widetilde{\mathtt{S}} \in \mathsf{Supp}(\widehat{\mathsf{CRS}})$ such that $\Pr[\widehat{\mathsf{CRS}} \in \widetilde{\mathtt{S}}] \geq \widehat{c}_0$, and for any $\mathtt{crs} \in \widetilde{\mathtt{S}}$, any $(\hat{n},\hat{k})$-distribution W, and any adversary $\mathcal{A}$, it holds that

$$\Pr\left[\begin{matrix} w \leftarrow W|_{\mathtt{crs}}, \varsigma \leftarrow \mathsf{CAuth}(\mathtt{crs}, w), \\ \varsigma^* \leftarrow \mathcal{A}(\mathtt{crs},\varsigma) : \mathsf{CVrfy}(\mathtt{crs}, w, \varsigma^*) = 1 \end{matrix}\right] < \widehat{\delta}/\widehat{c}_0.$$

Finally, by the parameter condition in Eq.1 that $(1-\widehat{c}_1)\widehat{\rho} + \widehat{c}_0 > 1$, it follows that $\mathtt{S} \cap \widetilde{\mathtt{S}} \neq \emptyset$. We pick one $\mathtt{crs}^* \in \mathtt{S} \cap \widetilde{\mathtt{S}}$, and define an ensemble of randomized function pairs $\{\mathsf{Auth} = \mathsf{CAuth}(\mathtt{crs}^*, \cdot), \mathsf{Vrfy} = \mathsf{CVrfy}(\mathtt{crs}^*, \cdot)\}$. It is easy to verify this $\{\mathsf{Auth}, \mathsf{Vrfy}\}$ satisfies $\hat{c}_1\hat{\rho}$-correctness and $\hat{\delta}/\hat{c}_0$ for all $(\hat{n},\hat{k})$-distributions. □

5 Computational Robust Extractors

In this section, we provide a generic framework in the CRS model that compiles any computational extractor into a robust one. Compared with previous works, our construction is the first that can work for any CRS *dependent* source with minimal entropy ($\omega(\log n)$ instead of $n/2$ as in the IT setting).

Intuitions. As briefly discussed in Introduction, a fairly intuitive idea is to add a MAC tag on the helper string. Namely, with a MAC $\{\mathsf{Tag}, \mathsf{Verify}\}$ (for simplicity here we omit the public parameters) and a strong extractor Ext, the generation procedure produces a helper string formed by $(s, \mathsf{Tag}(w, s))$ along with a randomness r, where s is the seed for Ext and r is the extracted randomness by Ext. The reproduce procedure first checks the validity of $\mathsf{Tag}(w, s)$, and reproduces $r = \mathsf{Ext}(s, w)$ if the tag is valid.

However, it is not hard to see the insufficiency of a normal MAC here. First, the secret w is non-uniform, and some information about w will be further leaked by (s, r) (for the strong post-application robustness), while a MAC usually requires a uniform key. Moreover, the tag $\mathsf{Tag}(w, s)$ may also leak partial information about w (e.g., some bits of it) and thus affect the quality of r. The above issues inspire us to consider a special MAC that can addresses the concerns above simultaneously. At a high level, 1) it should be secure w.r.t. auxiliary information about the *weak* secret w, as both the seed i_{ext} and the extracted string r generated from w are leaked to adversaries; 2) the tag of this MAC should also hide all partial information about w, such that given the tag the extracted string r remains pseudorandom. We call such a MAC κ-MAC (**K**ey-**P**rivate **A**uxiliary-input Message Authentication). But, for constructing a robust extractor, we only need to ask the *one-time* security of κ-MAC. [5]

We formally define κ-MAC, present and analyze our framework of robust extractors from κ-MAC. Then, we show how to construct (one-time) κ-MAC from well-studied assumptions.

κ-MAC: definitions. We define the syntax of κ-MAC in the CRS model.

Syntax. A κ-MAC scheme Σ consists of a triple of algorithms $\{\mathsf{Init}, \mathsf{Tag}, \mathsf{Verify}\}$, with associated key space $\mathcal{K} = \{\mathcal{K}_\lambda\}_{\lambda\in\mathbb{N}}$, message space $\mathcal{M}es = \{\mathcal{M}es_\lambda\}_{\lambda\in\mathbb{N}}$, and tag space $\mathcal{T} = \{\mathcal{T}_\lambda\}_{\lambda\in\mathbb{N}}$.

- $\mathsf{Init}(1^\lambda)$. On input a security parameter 1^λ, it outputs a $\mathtt{crs}$ whose distribution is denoted by CRS_λ.
- $\mathsf{Tag}(\mathtt{crs}, k, m)$. The authentication algorithm takes as inputs a CRS $\mathtt{crs}$, a key $k \in \mathcal{K}_\lambda$, and a message $m \in \mathcal{M}es_\lambda$. It outputs a tag $\varsigma \in \mathcal{T}_\lambda$.
- $\mathsf{Verify}(\mathtt{crs}, k, m, \varsigma)$. The verification algorithm takes as inputs a CRS $\mathtt{crs}$, a key k, a message m, and an authentication tag ς. It outputs either 1 accepting (m, ς) or 0 rejecting (m, ς).

[5] The RO-based MAC (where $\mathsf{Tag}(w, m) = H(w, m)$ for a random oracle H) employed in Boyen *et al.*'s robust (fuzzy) extractor [4] captures all above intuitions, and thus it can be considered as a κ-MAC in the random oracle model.

The correctness states that for every $\mathtt{crs} \leftarrow \mathsf{Init}(1^\lambda)$, every secret $k \in \mathcal{K}_\lambda$, and every message $m \in \mathcal{M}es_\lambda$, we have $\Pr[\mathsf{Verify}(\mathtt{crs}, k, m, \mathsf{Tag}(\mathtt{crs}, k, m))] = 1$. A secure κ-MAC scheme should satisfy *unforgeability*, which is similar to regular MAC, and *key privacy*, which requires the tag to be simulatable without using the key. The main difference (with the conventional definitions) in the security notions is that they are all under auxiliary input. We first discuss the admissible auxiliary input and then present the formal definitions.

Admissible auxiliary inputs. Note that the auxiliary information cannot be arbitrary. (1) it must be hard-to-invert leakage, as defined by Dodis *et al.* [9]. Namely, the auxiliary input is a function $f(w)$ of the secret w, and we say f is hard-to-invert w.r.t. a distribution W, if it is infeasible to recover w from $f(w)$, for a random sample $w \leftarrow W$. (2) to avoid triviality, the auxiliary information should not contain a valid authentication tag. Note that the authentication algorithm is indeed "hard-to-invert", and thus we have to put other restrictions on the leakage function to exclude the trivial case. Similar issues arise in auxiliary-input secure digital signatures [13] that they require the admissible function f to be *exponentially* hard-to-invert. For our purpose, however, this treatment will put unnecessary restrictions on either the sources being extracted or the underlying extractor. Instead, we observe and leverage the following asymmetry: the authentication algorithm is only required to be hard-to-invert for a randomly chosen CRS; while the auxiliary-input function, particularly, the Gen of the underlying extractor, can be hard-to-invert for *every* CRS. By defining the hardness of inverting over every CRS, we can exclude the authentication algorithm from admissible auxiliary-input functions. By design, we can further ensure that any efficient algorithm that produces valid authentication tags may not be "hard-to-invert" for some CRSs. Considering all the above, we define admissible auxiliary inputs below.

Definition 3. *Let $\mathsf{CRS} = \{\mathsf{CRS}_\lambda\}_{\lambda\in\mathbb{N}}$ be an ensemble of CRS distributions and $\mathcal{W}$ be a source that may depend on CRS. We call an efficiently computable function ensemble $\mathcal{F} = \{f_\lambda\}_{\lambda\in\mathbb{N}}$ ϵ-hard-to-invert w.r.t. $\mathcal{W}$ and CRS, if for any polynomial-time $\mathcal{A}$, any $\lambda \in \mathbb{N}$ and any $\mathtt{crs} \in \mathsf{Supp}(\mathsf{CRS}_\lambda)$, it holds that $\Pr[k \leftarrow W_\lambda|_{\mathtt{crs}} : \mathcal{A}(\mathtt{crs}, f(\mathtt{crs}, k)) = k] \leq \epsilon(\lambda)$.*

One-time unforgeability. The unforgeability captures the infeasibility of forging an authentication tag being accepted by a secret key k drawn from a high-entropy source. Particularly, it considers a key from a non-uniform distribution and allows adversaries to obtain auxiliary information.

Definition 4 (One-time unforgeability). *Let $\Sigma = \{\mathsf{Init}, \mathsf{Tag}, \mathsf{Verify}\}$ be a κ-MAC scheme with the key space $\{0,1\}^{n(\lambda)}$. We say Σ satisfies $(n, \epsilon_{\mathsf{unf}}, \epsilon_{\mathsf{hv}})$ one-time unforgeability, if for any polynomial-time adversary $\mathcal{A}$, any efficiently-samplable source $\mathcal{W}$ (defined over $\{\{0,1\}^{n(\lambda)}\}_{\lambda\in\mathbb{N}}$) and any function ensemble $\mathcal{F}$ s.t. $\mathcal{F}$ is ϵ_{hv} hard-to-invert w.r.t. $\mathcal{W}$ and* CRS, *it holds that $\mathsf{Adv}^{\mathsf{unf}}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda) = \Pr[\mathsf{Exp}^{\mathsf{unf}}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda) = 1] \leq \epsilon_{\mathsf{unf}}(\lambda)$. The experiment $\mathsf{Exp}^{\mathsf{unf}}_{\mathcal{A},\mathcal{W},\mathcal{F}}$ is defined below.*

```
$\mathsf{Exp}^{\mathsf{unf}}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda)$
$\mathtt{crs} \leftarrow \mathsf{Init}(1^\lambda); k \leftarrow W_\lambda|_{\mathtt{crs}}$
$(m, st) \leftarrow \mathcal{A}(\mathtt{crs}, f_\lambda(\mathtt{crs}, k)); \varsigma \leftarrow \mathsf{Tag}(\mathtt{crs}, k, m)$
$(m^*, \varsigma^*) \leftarrow \mathcal{A}(\varsigma, st)$
if $(m^*, \varsigma^*) \neq (m, \varsigma) \wedge \mathsf{Verify}(\mathtt{crs}, k, m^*, \varsigma^*) = 1$ then    return $1$
return $0$
```

One-time key privacy. This property seeks to capture that an adversary cannot learn anything new about the secret from an authentication tag.

We follow the simulation paradigm that was developed for defining non-interactive zero-knowledge [2]. Namely, with the help of some "trapdoor" information about the CRS, these tags can be simulated without the secret, and adversaries cannot distinguish simulated tags from real ones. The simulation procedure is done by the following pair: $\mathsf{SimInit}(1^\lambda)$ – the init simulation algorithm outputs a CRS $\mathtt{crs}$ along with its trapdoor τ. $\mathsf{SimTag}(\mathtt{crs}, \tau, m)$ – the tag simulation algorithm outputs a simulated tag ς for m. With the simulation algorithms, we can formally define this property.

Definition 5 (One-time key privacy). *Let $\Sigma = \{\mathsf{Init}, \mathsf{Tag}, \mathsf{Verify}\}$ be a κ-MAC scheme with the key space $\{0,1\}^{n(\lambda)}$. We say Σ satisfies $(n, \epsilon_{\mathsf{kpriv}}, \epsilon_{\mathsf{hv}})$ one-time key privacy, if there is a pair of PPT algorithms* $(\mathsf{SimInit}, \mathsf{SimTag})$, *and for any polynomial-time adversary $\mathcal{A}$, any efficiently-samplable source $\mathcal{W}$ (defined over $\{\{0,1\}^{n(\lambda)}\}_{\lambda\in\mathbb{N}}$) and any function ensemble $\mathcal{F}$ s.t. $\mathcal{F}$ is ϵ_{hv} hard-to-invert w.r.t. $\mathcal{W}$ and* CRS, *it holds that*

$$\mathsf{Adv}^{\mathrm{kpriv}}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{kpriv},0}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{kpriv},1}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda) = 1]| \leq \epsilon_{\mathsf{unf}}(\lambda).$$

The experiments $\mathsf{Exp}^{\mathsf{kpriv},0}_{\mathcal{A},\mathcal{W},\mathcal{F}}$ and $\mathsf{Exp}^{\mathsf{kpriv},1}_{\mathcal{A},\mathcal{W},\mathcal{F}}$ are defined below.

```
$\mathsf{Exp}^{\mathsf{kpriv},0}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda)$
$(\mathtt{crs},\tau) \leftarrow \mathsf{SimInit}(1^\lambda); k \leftarrow W_\lambda|_{\mathtt{crs}}$
$(m, st) \leftarrow \mathcal{A}(\mathtt{crs}, f_\lambda(\mathtt{crs}, k))$
$\varsigma \leftarrow \mathsf{SimTag}(\mathtt{crs}, \tau, m); b' \leftarrow \mathcal{A}(\varsigma, st)$
return $b'$
```

```
$\mathsf{Exp}^{\mathsf{kpriv},1}_{\mathcal{A},\Sigma,\mathcal{W},\mathcal{F}}(\lambda)$
$\mathtt{crs} \leftarrow \mathsf{Init}(1^\lambda); k \leftarrow W_\lambda|_{\mathtt{crs}}$
$(m, st) \leftarrow \mathcal{A}(\mathtt{crs}, f_\lambda(\mathtt{crs}, k))$
$\varsigma \leftarrow \mathsf{Tag}(\mathtt{crs}, k, m); b' \leftarrow \mathcal{A}(\varsigma, st)$
return $b'$
```

Making any computational extractor robust without requiring more entropy. We then show how to compile a strong extractor into a robust extractor (for general CRS dependent sources) using one-time κ-MAC. Let Ext be a (n, k, ℓ) strong extractor (working on (n, k)-sources, and output ℓ bits) with the seed length $s\ell$, and let $\Sigma = \{\mathsf{Init}, \mathsf{Tag}, \mathsf{Verify}\}$ be a κ-MAC scheme with the key space $\mathcal{K} = \{\{0,1\}^{n(\lambda)}\}_{\lambda\in\mathbb{N}}$ and the message space $\mathcal{M}es$ that contains $\{\{0,1\}^{\ell(\lambda)+s\ell(\lambda)}\}_{\lambda\in\mathbb{N}}$. Then, we illustrate our robust extractor construction $\mathcal{E} = \{\mathsf{CRS}, \mathsf{Gen}, \mathsf{Rep}\}$ in Fig.1.

```
$\mathsf{CRS}(1^\lambda)$
$\mathtt{crs} \leftarrow \mathsf{Init}(1^\lambda)$
return $\mathtt{crs}$
```

```
$\mathsf{Gen}(\mathtt{crs}, w)$
$s \leftarrow\$ \{0,1\}^{s\ell(\lambda)}, r \leftarrow \mathsf{Ext}(s, w)$
$\varsigma \leftarrow \mathsf{Tag}(\mathtt{crs}, w, s)$
return $R = r, P = (s, \varsigma)$
```

```
$\mathsf{Rep}(\mathtt{crs}, w, P)$
if $\mathsf{Verify}(\mathtt{crs}, w, s, \varsigma) = 1$
    return $R = \mathsf{Ext}(s, w)$
return $\perp$
```

Fig. 1. Robust extractor from randomness extractor + one time κ-MAC

Analysis. The correctness and security of our construction are fairly straightforward. We remark that we only require the source to have minimal min-entropy to enable a strong extractor. Formally, we have the following:

Theorem 2. *Let* Ext *be an* (n, k, ℓ)*-strong extractor with* ϵ_{ext}*-privacy,* Σ *be a* κ*-MAC with* $(n, \epsilon_{\mathsf{kpriv}}, \epsilon_{\mathsf{hv}})$ *one-time key privacy and* $(n, \epsilon_{\mathsf{unf}}, \epsilon_{\mathsf{hv}})$ *one-time robustness. If* $\epsilon_{\mathsf{hv}} \geq \epsilon_{\mathsf{ext}}$*, then for any* ϵ_{priv}*,*δ_{rob}*, satisfying* $\epsilon_{\mathsf{priv}} \geq \epsilon_{\mathsf{ext}} + 2\epsilon_{\mathsf{kpriv}}$*, and* $\delta_{\mathsf{rob}} > \epsilon_{\mathsf{unf}}$*, the construction in Fig.1 is an* (n, k, ℓ)*-robust extractor with* ϵ_{priv}*-privacy and* δ_{rob}*-post-application-robustness (defined in Sec. 4).*

We prove privacy and robustness in Lemma 3 and Lemma 4, respectively.

Lemma 3. *Assume that* Ext *satisfies* ϵ_{ext}*-privacy, and* Σ *satisfies* $(n, \epsilon_{\mathsf{kpriv}}, \epsilon_{\mathsf{hv}})$ *one-time key privacy, where* $\epsilon_{\mathsf{hv}} \geq \epsilon_{\mathsf{ext}}$*. Then,* rExt *(in Fig.3) satisfies* ϵ_{priv}*-privacy, for any* $\epsilon_{\mathsf{priv}} > \epsilon_{\mathsf{ext}} + 2\epsilon_{\mathsf{kpriv}}$*.*

Proof. We prove this lemma by contradiction. Assume there is $\epsilon_0 > \epsilon_{\mathsf{ext}} + 2\epsilon_{\mathsf{kpriv}}$, and we have a polynomial-time adversary $\mathcal{B}$ who has an advantage greater than ϵ_0 w.r.t. some efficiently-samplable (n, k)-source $\mathcal{W}$. Then, we leverage $\mathcal{B}$ to construct a polynomial-time adversary $\mathcal{A}_{\mathsf{ext}}$ for Ext, and two polynomial-time adversaries $\mathcal{A}_{\mathsf{mac},0}$ and $\mathcal{A}_{\mathsf{mac},1}$ for κ-MAC Σ, such that, for the source $\mathcal{W}$,

$$\mathsf{Adv}^{\mathrm{ext}}_{\mathcal{A}_{\mathsf{ext}},\mathcal{W}}(\lambda) + \mathsf{Adv}^{\mathrm{kpriv}}_{\mathcal{A}_{\mathsf{mac},0},\mathcal{W},\mathcal{F}}(\lambda) + \mathsf{Adv}^{\mathrm{kpriv}}_{\mathcal{A}_{\mathsf{mac},1},\mathcal{W},\mathcal{F}}(\lambda) > \epsilon_0, \tag{5}$$

where $\mathcal{F}$ is a funtion ensemble implementing Ext. As $\epsilon_{\mathsf{hv}} \geq \epsilon_{\mathsf{ext}}$, such $\mathcal{F}$ is an admissible auxiliary inputs. Now, since we assume $\epsilon_0 > \epsilon_{\mathsf{ext}} + 2\epsilon_{\mathsf{kpriv}}$, it follows that either $\mathsf{Adv}^{\mathrm{ext}}_{\mathcal{A}_{\mathsf{ext}},\mathcal{W}}(\lambda) > \epsilon_{\mathsf{ext}}$, $\mathsf{Adv}^{\mathrm{kpriv}}_{\mathcal{A}_{\mathsf{mac},0},\mathcal{W},\mathcal{F}}(\lambda) > \epsilon_{\mathsf{kpriv}}$, or $\mathsf{Adv}^{\mathrm{kpriv}}_{\mathcal{A}_{\mathsf{mac},1},\mathcal{W},\mathcal{F}}(\lambda) > \epsilon_{\mathsf{kpriv}}$.

Now, we give the code of each adversary in Fig.2.

```
Algorithm $\mathcal{A}_{\mathsf{ext}}(i_{\mathsf{ext}}, r)$
$(\mathtt{crs}, \tau) \leftarrow \mathsf{SimInit}(1^{\lambda})$
$\varsigma \leftarrow \mathsf{SimTag}(\mathtt{crs}, \tau, i_{\mathsf{ext}})$
$b' \leftarrow \mathcal{B}(\mathtt{crs}, (i_{\mathsf{ext}}, \varsigma), r)$
return $b'$

Algorithm $\mathcal{A}^{\mathcal{O}_{\beta}}_{\mathsf{mac},b}(\mathtt{crs}, (i_{\mathsf{ext}}, r))$
Query $\mathcal{O}_{\beta}$ with $i_{\mathsf{ext}}$, and obtain $\varsigma$
$R_0 \leftarrow\$ \{0,1\}^{\ell(\lambda)}, R_1 = r$
$\beta' \leftarrow \mathcal{B}(\mathtt{crs}, (i_{\mathsf{ext}}, \varsigma), R_b)$
return $\beta'$
```

Fig. 2. Construction of $\mathcal{A}_{\mathsf{ext}}$ and $\mathcal{A}_{\mathsf{am},b}$. In $\mathcal{A}_{\mathsf{ext}}$, $(\mathsf{SimInit}, \mathsf{SimTag})$ is the simulator of κ-MAC. In $\mathcal{A}_{\mathsf{mac},b}$, r is the extracted randomness from w with the seed i_{ext}. $\mathcal{O}_{\beta}$ returns a real tag when $\beta = 1$ or returns a simulated tag when $\beta = 0$.

It is easy to see that $\mathcal{A}_{\mathsf{ext}}$ and $\mathcal{A}_{\mathsf{am},b}$ are polynomial-time. Now, we argue advantages of each adversary.

Recall the privacy definition of a robust extractor (cf. Def.2). The adanvantage of $\mathcal{B}$ against rExt's privay w.r.t. $\mathcal{W}$ is defined by $\mathsf{Adv}^{\mathrm{priv}}_{\mathcal{B},\mathcal{W}}(\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{priv},0}_{\mathcal{B},W}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{priv},1}_{\mathcal{B},W}(\lambda) = 1]|$. Let us assume that

$$p_0 = \Pr\begin{bmatrix} w \leftarrow W_{\lambda}, i_{\mathsf{ext}} \leftarrow\$ \{0,1\}^{si(\lambda)} \\ r \leftarrow\$ \{0,1\}^{\ell(\lambda)} : 1 \leftarrow \mathcal{A}_{\mathsf{ext}}(i_{\mathsf{ext}}, r) \end{bmatrix},$$

$$p_1 = \Pr\begin{bmatrix} w \leftarrow W_{\lambda}, i_{\mathsf{ext}} \leftarrow\$ \{0,1\}^{si(\lambda)} \\ r \leftarrow \mathsf{Ext}(i_{\mathsf{ext}}, w) : 1 \leftarrow \mathcal{A}_{\mathsf{ext}}(i_{\mathsf{ext}}, r) \end{bmatrix}.$$

Then, by definition, the advanatage of $\mathcal{A}_{\mathsf{ext}}$ against Ext is $\mathsf{Adv}^{\mathrm{ext}}_{\mathcal{A}_{\mathsf{ext}},\mathcal{W}}(\lambda) = |p_0 - p_1|$. For $b \in \{0, 1\}$, we denote $\Pr[\mathsf{Exp}^{\mathsf{priv},b}_{\mathcal{B},\mathcal{W}}(\lambda) = 1] - p_b = \Delta_b$. By standard arguments, we have

$$\mathsf{Adv}^{\mathrm{priv}}_{\mathcal{B},\mathcal{W}}(\lambda) = \mathsf{Adv}^{\mathrm{ext}}_{\mathcal{A}_{\mathsf{ext}},\mathcal{W}}(\lambda) + |\Delta_0| + |\Delta_1| \tag{6}$$

It is easy to verify that, at the point of $\mathcal{B}$'s view, the experiment $\mathsf{Exp}^{\mathsf{priv},b}_{\mathcal{B},\mathcal{W}}$ is identical to $\mathsf{Exp}^{\mathsf{kpriv},1}_{\mathcal{A}_{\mathsf{mac},b},\mathcal{W},\mathcal{F}}$ (cf. Def.5), and thus $\Pr[\mathsf{Exp}^{\mathsf{priv},b}_{\mathcal{B},\mathcal{W}}(\lambda) = 1] = \Pr[\mathsf{Exp}^{\mathsf{kpriv},1}_{\mathcal{A}_{\mathsf{mac},b},\mathcal{W},\mathcal{F}}(\lambda) = 1]$. Similarly, we have $p_b = \Pr[\mathsf{Exp}^{\mathsf{kpriv},0}_{\mathcal{A}_{\mathsf{mac},b},\mathcal{W},\mathcal{F}} = 1]$. Notice that $\mathsf{Adv}^{\mathrm{kpriv}}_{\mathcal{A}_{\mathsf{mac},b},\mathcal{W},\mathcal{F}}(\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{kpriv},0}_{\mathcal{A}_{\mathsf{mac},b},\mathcal{W},\mathcal{F}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{kpriv},1}_{\mathcal{A}_{\mathsf{mac},b},\mathcal{W},\mathcal{F}}(\lambda) = 1]|$, we have $\mathsf{Adv}^{\mathrm{kpriv}}_{\mathcal{A}_{\mathsf{mac},b},\mathcal{W},\mathcal{F}}(\lambda) = \Delta_b$, thus Eq.6. □

Lemma 4. *Assume that* Ext *satisfies* ϵ_{ext}*-privacy, and* Σ *satisfies* $(n, \epsilon_{\mathsf{unf}}, \epsilon_{\mathsf{hv}})$ *one-time unforgeability, where* $\epsilon_{\mathsf{hv}} \geq \epsilon_{\mathsf{ext}}$*. Then,* rExt *(in Fig.3) satisfies* δ_{rob}*-post-application-robustness, for any* $\delta_{\mathsf{rob}} \geq \epsilon_{\mathsf{unf}}$*.*

Proof. We prove this lemma by contradiction. Assume there is $\delta_0 > \epsilon_{\mathsf{unf}}$, and we have a polynomial-time adversary $\mathcal{B}$ who has an advantage greater than δ_0 w.r.t. some efficiently-samplable (n, k)-source $\mathcal{W}$. Then, we leverage $\mathcal{B}$ to construct a polynomial adversary $\mathcal{A}_{\mathsf{mac}}$ against the unforgeability of κ-MAC Σ w.r.t. $\mathcal{W}$, with advantage $\mathsf{Adv}^{\mathrm{unf}}_{\mathcal{A}_{mac},\mathcal{W},\mathcal{F}}(\lambda) > \delta_0 > \epsilon_{\mathsf{unf}}$. Here $\mathcal{F}$ is the function ensemble implementing Ext.

$\mathcal{A}_{\mathsf{mac}}$ can be easilly constructed. Given $\mathtt{crs}$ of Σ and (i_{ext}, r) which are the seed and the extracted randomness respectively from w (treated as auxiliary input), $\mathcal{A}_{\mathsf{mac}}$ asks an authentication tag ς on i_{ext}, and invokes $\mathcal{B}$ by giving $(\mathtt{crs}, (i_{\mathsf{ext}}, \varsigma), r)$. When $\mathcal{B}$ breaks the robustness, *i.e.*, it outputs $P^* = (i^*_{\mathsf{ext}}, \varsigma^*) \neq (i_{\mathsf{ext}}, \varsigma)$ s.t. $\mathsf{Verify}(\mathtt{crs}, w, i^*_{\mathsf{ext}}, \varsigma^*) = 1$, $\mathcal{A}_{\mathsf{am}}$ can output $(i^*_{\mathsf{ext}}, \varsigma^*)$ as a forgery. It is easy to see that $\mathcal{A}_{\mathsf{am}}$ is polynomial-time. □

Constructing one-time κ-MAC. Now we discuss how to construct a κ-MAC. It is natural to view κ-MAC as a special leakage-resilient MAC, then upgrade it to add "key privacy". Given state of the art, the only known approach to MACs tolerating hard-to-invert leakage is using auxiliary-input secure signatures [13, 23]. However, when considering weak keys and key privacy, it turns out to be more involved. We have to revisit the design framework of auxiliary-input secure signatures, adapt it to the symmetric setting, and address the subsequent challenges for realizing the new framework. To illustrate the challenges and ideas towards κ-MAC we first briefly recall Katz-Vaikuntanathan's leakage-resilient signature scheme [17] which was later shown by Faust *et al.* [13] to be secure against hard-to-invert leakage (with minor modifications). For the sake of clarification, we follow Dodis *et al.*'s [8] insightful abstraction, which bases KV signature upon the following building blocks.

- A leakage-resilient hard relation R_{LR} with its sampling algorithm $\mathsf{Gen}_{\mathsf{LR}}$. R is an NP relation, and $\mathsf{Gen}_{\mathsf{LR}}$ is a PPT algorithm which always outputs $(y, k) \in R_{\mathsf{LR}}$. We say R_{LR} is leakage-reslient, if for any efficient adversary $\mathcal{A}$ and any admissible leakage function f, we have

$$\Pr[(y, k) \leftarrow \mathsf{Gen}_{\mathsf{LR}}(1^{\lambda}), k^* \leftarrow \mathcal{A}(y, f(y, k)) : (y, k^*) \in R_{\mathsf{LR}}] \leq \mathsf{negl}(\lambda).$$

- A true-simulation-extractable NIZK (tSE-NIZK) [8] $\mathit{\Pi}$ for the relation $\bar{R}_{\mathsf{LR}} := \{(y, k, m) : (y, k) \in R_{\mathsf{LR}}\}$. $\mathit{\Pi}$ consists of a setup algorithm S_{zk}, a prover algorithm P_{zk}, and a verifier algorithm V_{zk}.

Informally, Katz-Vaikuntanathan signature proceeds as follows: To sign a message m, the signer with sk proves the knowledge of k for a statement $(y, k, m) \in \bar{R}_{\mathsf{LR}}$ and returns the proof π as the signature σ, where $(y, k) \in R_{\mathsf{LR}}$ is part of the verification key. Given that $\mathit{\Pi}$ is a tSE-NIZK, a successful forgery will violate that R_{LR} is a leakage-resilient hard relation. Specifically, the zero-knowledge guarantees the signature will not leak new information about k, and the true-simulation-extractability ensures that an adversary who successfully generated a forgery must have k^* *s.t.* $(y, k^*) \in R_{\mathsf{LR}}$. It follows that this adversary could produce k^* only given the verification key y and the leakage $f(y, k)$, which contradicts our assumption that R_{LR} is leakage-resilient hard.

Towards κ-MAC. While we can trivially use a signature scheme as a MAC by taking both vk and sk as the authentication key, this approach will require the key to be uniform. However, κ-MAC needs to work for weak keys. The central question is how to safely generate and share (vk, sk) between the sender and the receiver (verifier), while they initially only have a weak key in common that relates to the CRS.

It is safe to treat the CRS of tSE-NIZK (contained in the verification key vk) as a part of CRS in our κ-MAC construction. We then deal with $(y, k) \in R_{\mathsf{LR}}$. A natural approach is to take the shared weak key as k and efficiently generate y according to k. However, while signatures can assume a bulletin board for posting verification keys, in κ-MAC, y has to be sent to the verifier via an unauthenticated channel (namely, being a part of the authentication tag). Consequently, adversaries might alter y to y', as the verifier will not notice this change if $(y', k) \in R_{\mathsf{LR}}$. To prevent those attacks, we take the following steps.

- Observe that there might be a part of y (denoted by pp) that could be generated without k and reused across statements. We let pp be a part of CRS such that adversaries cannot modify it.

- We strengthen the definition of leakage-resilient hard relation against adversaries who alter the other part of y (denoted by yk). Namely, given (pp, yk) and leakage about k, adversaries cannot generate (yk', k') s.t. $((\mathsf{pp}, yk'), k') \in R_{\mathsf{LR}}$ and $((\mathsf{pp}, yk'), k) \in R_{\mathsf{LR}}$. We call such a relation a strengthened leakage-resilient hard relation (sLRH relation).

Next, for *key privacy*, yk (as a statement) should be indistinguishable with another $\widetilde{yk}$ (simulated without k). Note that this requirement cannot be bypassed, even when yk is uniquely determined by (pp, k) and is not contained in the authentication tag explicitly, since a NIZK proof is not supposed to hide the statement being proved. We therefore require the generator of κ-MAC to be a private generator.

We formalize all notions and intuitions in the following definition.

Definition 6. *Let R_{LR} be an NP relation defined over $\{Y_\lambda \times \{0,1\}^{n(\lambda)}\}_{\lambda\in\mathbb{N}}$,*

- ***Generator.*** *A pair of PPT algorithms* $(\mathsf{PGen}, \mathsf{SGen})$ *is a generator of R_{LR}, if for every $\lambda \in \mathbb{N}$ and $k \in \{0,1\}^{n(\lambda)}$, it follows that*

$$\Pr[pp \leftarrow \mathsf{PGen}(1^\lambda), yk \leftarrow \mathsf{SGen}(pp, k) : ((pp, yk), k) \in R_{\mathsf{LR}}] = 1.$$

- ***sLRH relation.*** *R_{LR} along with* $(\mathsf{PGen}, \mathsf{SGen})$ *is an $(n, \epsilon_{\mathsf{lr}}, \epsilon_{\mathsf{hv}})$-sLRH relation, if for any efficiently-samplable source $\mathcal{W}$ (over $\{\{0,1\}^{n(\lambda)}\}_{\lambda\in\mathbb{N}}$ and dependent of* PGen*) and any function ensemble $\mathcal{F}$ s.t. $\mathcal{F}$ is ϵ_{hv} hard-to-invert w.r.t. $\mathcal{W}$ and* PGen*, for any P.P.T adversary $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathrm{slrh}}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda) \le \epsilon_{\mathsf{lr}}(\lambda)$ where $\mathsf{Adv}^{\mathrm{slrh}}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda)$ is defined as*

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{PGen}(1^\lambda), k \leftarrow W_\lambda|_{pp}, yk \leftarrow \mathsf{SGen}(pp, k), \\ (yk', k') \leftarrow \mathcal{A}(pp, yk, f_\lambda(pp, k)) : (pp, yk', k'), (pp, yk', k) \in R_{\mathsf{LR}} \end{array}\right].$$

- ***Private generator.*** $(\mathsf{PGen}, \mathsf{SGen})$ *satisfies $(n, \epsilon_{\mathsf{pr}}, \epsilon_{\mathsf{hv}})$-privacy, if for $(\mathcal{A}, \mathcal{W}, \mathcal{F})$ above, $\mathsf{Adv}^{\mathrm{pr}}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda) \le \epsilon_{\mathsf{pr}}(\lambda)$, where $\mathsf{Adv}^{\mathrm{pr}}_{\mathcal{A},\mathcal{W},\mathcal{F}}(\lambda) =$*

$$\left| \Pr\left[\begin{array}{l} pp \leftarrow \mathsf{PGen}(1^\lambda) \\ k \leftarrow W_\lambda|_{pp} \\ yk \leftarrow \mathsf{SGen}(pp, k) : \\ 1 = \mathcal{A}(pp, yk, f_\lambda(pp, k)) \end{array}\right] - \Pr\left[\begin{array}{l} pp \leftarrow \mathsf{PGen}(1^\lambda) \\ k \leftarrow W_\lambda|_{pp}, k' \leftarrow\$ \{0,1\}^{n(\lambda)} \\ yk \leftarrow \mathsf{SGen}(pp, k') : \\ 1 = \mathcal{A}(pp, yk, f_\lambda(pp, k)) \end{array}\right] \right|.$$

Remark 2. The auxiliary-input function f does not take as input yk, because yk is generated by the authentication algorithm, and the auxiliary input is supposed to be leaked before authenticating. The source $\mathcal{W}$ and the leakage are dependent on pp since it is a part of the CRS. Other parts of CRS are not considered explicitly since SGen does not use them.

The final κ-MAC construction. Using an sLRH relation R_{LR} along with its private generator $(\mathsf{PGen}, \mathsf{SGen})$ and a tSE-NIZK $\mathit{\Pi} = \{\mathsf{S}_{\mathsf{zk}}, \mathsf{P}_{\mathsf{zk}}, \mathsf{V}_{\mathsf{zk}}\}$ for the relation $\bar{R}_{\mathsf{LR}} := \{(\mathsf{pp}, yk, k, m) : ((\mathsf{pp}, yk), k) \in R_{\mathsf{LR}}\}$, we construct an one-time κ-MAC scheme in Fig.3. [6]

```
$\mathsf{Init}(1^\lambda)$
    $\mathtt{crs}_{\mathsf{zk}} \leftarrow \mathsf{S}_{\mathsf{zk}}(1^\lambda)$
    $\mathsf{pp} \leftarrow \mathsf{PGen}(1^\lambda)$
    return
        $\mathtt{crs} = (\mathtt{crs}_{\mathsf{zk}}, \mathsf{pp})$

$\mathsf{Tag}(\mathtt{crs}, k, m)$
    $yk \leftarrow \mathsf{SGen}(\mathsf{pp}, k)$
    $\pi \leftarrow \mathsf{P}_{\mathsf{zk}}(\mathtt{crs}_{\mathsf{zk}},$
        $(\mathsf{pp}, yk, m), k)$
    return $\varsigma = (yk, \pi)$

$\mathsf{Verify}(\mathtt{crs}, k, m, \varsigma)$
    return 1 iff
        $(\mathsf{pp}, yk, k) \in R_{\mathsf{LR}}$ and
        $\mathsf{V}_{\mathsf{zk}}(\mathtt{crs}_{\mathsf{zk}}, (\mathsf{pp},$
            $yk, m), \pi) = 1$
```

Fig. 3. One-time κ-MAC from tSE-NIZK + sLRH relation

Analysis. Correctness is easy to see. Regarding security: from the privacy of the generator SGen and the zero-knowledgeness of $\mathit{\Pi}$, efficient adversaries cannot learn new information about k from the tag (y, π), and the key privacy follows. The tSE-NIZK ensures an adversary who successfully forges an authentication tag can also output a pair $(y', k') \in R_{\mathsf{LR}}$ *s.t.* $(y', k) \in R_{\mathsf{LR}}$, which contradicts the sLRH relation, and thus the unforgeability follows. Formal analysis is presented in the full paper.

Theorem 3. *Let* $(\mathsf{PGen}, \mathsf{SGen})$ *be an* $(n, \epsilon_{\mathsf{pr}}, \epsilon_{\mathsf{hv}})$*-private generator for an NP relation* R_{LR}*, and* R_{LR} *along with* $(\mathsf{PGen}, \mathsf{SGen})$ *be an* $(n, \epsilon_{\mathsf{lr}}, \epsilon_{\mathsf{hv}})$*-sLRH relation. Let* $\mathit{\Pi} = \{\mathsf{S}_{\mathsf{zk}}, \mathsf{P}_{\mathsf{zk}}, \mathsf{V}_{\mathsf{zk}}\}$ *be a NIZK for the relation* $\bar{R}_{\mathsf{LR}}$ *satisfying* ϵ_{zk}*-ZK and* $(\epsilon_{\mathsf{tse1}}, \epsilon_{\mathsf{tse2}})$*-tSE. Then, the construction in Fig.3 satisfies* $(n, \epsilon_{\mathsf{kpriv}}, \epsilon_{\mathsf{hv}})$ *one-time key privacy and* $(n, \epsilon_{\mathsf{unf}}, \epsilon_{\mathsf{hv}})$ *one-time unforgeability, for any* $\epsilon_{\mathsf{kpriv}} \geq \epsilon_{\mathsf{pr}} + \epsilon_{\mathsf{zk}}$*, and any* $\epsilon_{\mathsf{unf}} \geq \epsilon_{\mathsf{zk}} + \epsilon_{\mathsf{tse1}} + \epsilon_{\mathsf{tse2}} + \epsilon_{\mathsf{lr}}$*.*

As shown by Dodis *et al.* [8], a tSE-NIZK could be constructed using CPA-secure PKE and standard NIZK, or CCA-secure PKE and simulation-sound NIZK. Both approaches can be based on standard assumptions. However, while a leakage-resilient hard relation can be instantiated with a second-preimage-resistant hash function H, the statement $y = H(k)$ will leak some information about k. For key privacy, we need new constructions for strengthened LRH relations.

[6] The one-time κ-MAC is enough for our purpose; we may generalize our construction to get a full-fledged κ-MAC using multi-message secure DPKE [5], which will require concrete entropy bound on the source though.

sLRH relation from deterministic PKE. Note that the privacy of generator is not an orthogonal property of sLRH relation; it indeed prevents adversaries from finding the exact k from (pp, yk) and the leakage. If it is further ensured that adversaries cannot find a distinct k' along with yk' such that both (pp, yk', k) and (pp, yk', k') belong to R_{LR}, R_{LR} with a private generator will be a sLRH relation. We therefore abstract a useful property of R_{LR} called "collision resistance" below.

Definition 7. *R_{LR} is $(n, \epsilon_{\mathsf{cr}})$-collision-resistant w.r.t.* PGen*, if for any polynomial-time $\mathcal{A}$, it holds that*

$$\Pr\left[\begin{array}{l} \boldsymbol{pp} \leftarrow \mathsf{PGen}(1^\lambda), (yk, k, k') \leftarrow \mathcal{A}(\boldsymbol{pp}) : \\ \qquad k \neq k' \wedge (\boldsymbol{pp}, yk, k) \in R_{\mathsf{LR}} \wedge (\boldsymbol{pp}, yk, k') \in R_{\mathsf{LR}} \end{array}\right] \leq \epsilon_{\mathsf{cr}}(\lambda).$$

As discussed before, a collision-resistant relation with a private generator will be a sLRH relation. (The formal proof is in the full paper.)

Lemma 5. *Let* $(\mathsf{PGen}, \mathsf{SGen})$ *be an $(n, \epsilon_{\mathsf{pr}}, \epsilon_{\mathsf{hv}})$-private generator for R_{LR}. If R_{LR} satisfies $(n, \epsilon_{\mathsf{cr}})$-collision-resistance w.r.t.* PGen*, R_{LR} with* $(\mathsf{PGen}, \mathsf{SGen})$ *is an $(n, \epsilon_{\mathsf{lr}}, \epsilon_{\mathsf{hv}})$-sLRH relation, for any $\epsilon_{\mathsf{lr}} \geq \epsilon_{\mathsf{pr}} + \epsilon_{\mathsf{cr}}$.*

We now construct a collision-resistant relation with a private generator. An auxiliary-input secure deterministic public-key encryption (DPKE) scheme is a natural tool for realizing an NP relation with a private generator. Since no randomness is used, it is easy to check whether a ciphetertext c_{de} encrypts a message m_{de} under a public key pk_{de}. We can define an NP relation R_{de} such that $(pk_{\mathsf{de}}, c_{\mathsf{de}}, m_{\mathsf{de}}) \in R_{\mathsf{de}}$ iff $c_{\mathsf{de}}{=}\mathsf{E}_{\mathsf{de}}(pk_{\mathsf{de}}, m_{\mathsf{de}})$. From the auxiliary-input security of DPKE, the key generation algorithm and the encryption algorithm will give a private generator for R_{de}.

The relation R_{de} is almostly collision-resistant. Under a valid public key pk_{de} (namely, there is a secret key sk_{de} to decrypt all ciphertexts under pk_{de}), the (perfect) correctness of DPKE ensures that for any ciphertext c_{de} there is at most one message m_{de} such that $c_{\mathsf{de}}{=}\mathsf{E}_{\mathsf{de}}(pk_{\mathsf{de}}, m_{\mathsf{de}})$. While it seems straightforward to ensure the validity of pk_{de} by putting it into the CRS, however, it violates security. The problem inherits from that DPKE only applies to message distributions *independent* of public key, but our goal is to have a construction for CRS-*dependent* sources.

We enforce the validity of public key as follows: note that a valid pair $(pk_{\mathsf{de}}, sk_{\mathsf{de}})$ defines an NP relation R_{pk}, and pk_{de} can be ensured valid (with overwhelming probability) using a NIZK proof demonstrating the knowledge of sk_{de} *s.t.* $(pk_{\mathsf{de}}, sk_{\mathsf{de}}) \in R_{\mathsf{pk}}$ (the key relation). Now, pk_{de}

(with its validity proof) can be outputted by SGen, and PGen is only used to establish a CRS of NIZK. Though CRS is still in need, adaptively secure NIZK does allow CRS-dependent statements. The relation R_{de} will be extended for verifying the proof. Formally, let $\Sigma_{\mathsf{de}} = \{\mathsf{K}_{\mathsf{de}}, \mathsf{E}_{\mathsf{de}}, \mathsf{D}_{\mathsf{de}}\}$ be an auxiliary-input secure DPKE scheme and the key relation R_{pk}, and $\Pi_{\mathsf{pk}} = \{\mathsf{S}_{\mathsf{pk}}, \mathsf{P}_{\mathsf{pk}}, \mathsf{V}_{\mathsf{pk}}\}$ be a NIZK for R_{pk}. We define an NP relation $R^{\mathsf{de}}_{\mathsf{LR}}$ and construct its generator $(\mathsf{PGen}_{\mathsf{de}}, \mathsf{SGen}_{\mathsf{de}})$ below.

- Let $\mathsf{pp} = \mathtt{crs}_{\mathsf{pk}}$, $yk = (c_{\mathsf{de}}, pk_{\mathsf{de}}, \pi_{\mathsf{de}})$ and $k = m_{\mathsf{de}}$. $(\mathsf{pp}, yk, k) \in R^{\mathsf{de}}_{\mathsf{LR}}$ iff $c_{\mathsf{de}} = \mathsf{E}_{\mathsf{de}}(pk_{\mathsf{de}}, m_{\mathsf{de}})$ and $\mathsf{V}_{\mathsf{pk}}(\mathtt{crs}_{\mathsf{pk}}, pk_{\mathsf{de}}, \pi_{\mathsf{de}}) = 1$.
- $\mathsf{PGen}_{\mathsf{de}}(1^\lambda)$. Invoke $\mathtt{crs}_{\mathsf{pk}} \leftarrow \mathsf{S}_{\mathsf{pk}}(1^\lambda)$, and return $\mathsf{pp} = \mathtt{crs}_{\mathsf{pk}}$.
- $\mathsf{SGen}_{\mathsf{de}}(\mathsf{pp}, k = m_{\mathsf{de}})$. Invoke $(pk_{\mathsf{de}}, sk_{\mathsf{de}}) \leftarrow \mathsf{K}_{\mathsf{de}}(1^\lambda)$, $\pi_{\mathsf{de}} \leftarrow \mathsf{P}_{\mathsf{pk}}(\mathtt{crs}_{\mathsf{pk}}, pk_{\mathsf{de}}, sk_{\mathsf{de}})$, and $c_{\mathsf{de}} \leftarrow \mathsf{E}_{\mathsf{de}}(pk_{\mathsf{de}}, m_{\mathsf{de}})$. Return $yk = (c_{\mathsf{de}}, pk_{\mathsf{de}}, \pi_{\mathsf{de}})$.

Summarizing above, we have the following result, whose formal analysis is in the full paper.

Lemma 6. *Let Σ_{de} be $(n, \epsilon_{\mathsf{hv}}, \epsilon_{\mathsf{ind}})$-PRIV-IND secure DPKE with message space $\{\{0,1\}^{n(\lambda)}\}_{\lambda\in\mathbb{N}}$, R_{pk} be its key relation. Let Π_{pk} be a NIZK for R_{pk} with ϵ_{zk}-ZK and ϵ_{snd}-adpative-soundness.* $(\mathsf{PGen}_{\mathsf{de}}, \mathsf{SGen}_{\mathsf{de}})$ *is a $(n, \epsilon_{\mathsf{pr}}, \epsilon_{\mathsf{hv}})$-private generator of $R^{\mathsf{de}}_{\mathsf{LR}}$ for any $\epsilon_{\mathsf{pr}} \geq \epsilon_{\mathsf{ind}} + 2\epsilon_{\mathsf{zk}}$, and $R^{\mathsf{de}}_{\mathsf{LR}}$ is $(n, \epsilon_{\mathsf{cr}})$-collision resistant w.r.t.* $\mathsf{PGen}_{\mathsf{de}}$*, for any $\epsilon_{\mathsf{cr}} \geq \epsilon_{\mathsf{snd}}$.*

Under the exponentially-hard DDH assumption [24], it is known to exist a DPKE which is perfectly correct and secure against any ϵ-hard-to-invert leakage (as long as ϵ is a negligible function and s is a polynomial). Following Theorem 3 and Lemma 6, we have a κ-MAC against any ϵ-hard-to-invert leakage and thus can compile any secure randomness extractor.

6 Extension to Robust Fuzzy Extractors

In this section, we construct robust fuzzy extractors.

Intuition. Similar to the non-fuzzy case, we use a κ-MAC scheme to authenticate the helper string of the underlying fuzzy extractor. However, correctness and security will not directly inherit from the non-fuzzy case. Correctness can be fixed easily. We can use secure sketches to construct the underlying fuzzy extractor; thus, one can recover the original secret w from the helper string using a close secret w'.

We now discuss the obstacles towards security. While the helper string has to contain a secure sketch, the adversary may manipulate the secure sketch such that secret w'' recovered from it is not identical to the original secret w, and she may forge an authentication tag being accepted by w'' to break the robustness. We can simply reject all w'' that are not t-close to w' (in this case w'' must be incorrect), and an allowed w'' will be $2t$-close to w. The challenge is to ensure that adversaries cannot forge an authentication tag being accepted by this $2t$-close secret. In the following, we introduce *fuzzy unforgeability* of κ-MAC and show that the construction in the last section already satisfies this property. Then, we construct a robust fuzzy extractor for CRS-dependent sources by using fuzzy-unforgeable κ-MAC.

κ-MAC with fuzzy unforgeability. A κ-MAC scheme $\Sigma = \{\mathsf{Init}, \mathsf{Tag}, \mathsf{Verify}\}$ satisfies q-fuzzy unforgeability, if given an authentication tag ς from k along with an auxiliary input about k, one cannot forge a new authentication tag being accepted by any secret k' which is q-close to k. The formal definition (presented in the full paper) is parameterized by $(n, q, \epsilon_{\mathsf{unf}})$ along with $\mathbb{W}$ and $\mathbb{F}$, where n is the length of the secret, ϵ_{unf} is the advantage of polynomial-time adversaries, $\mathbb{W}$ is the admissible family of sources, and $\mathbb{F}$ is the family of admissible leakage functions.

Construction from fuzzy sLRH relation. Recall our κ-MAC construction in Fig.3. If an adversary who is given yk and leakage about k outputs a forgery being accepted by a secret k^*, then, by tSE-NIZK, the adversary is able to output (yk', k') such that both (pp, yk', k') and (pp, yk', k^*) belong to the relation R_{LR}. For one-time standard unforgeability, k and k^* are equal, and such an adversary contradicts the definition of sLRH relation. For one-time q-fuzzy unforgeability, k^* will just be q-close to w, and we therefore strengthen the sLRH relation into its fuzzy version accordingly. More precisely, we call an NP relation R_{LR} a q-fuzzy relation w.r.t. $(\mathsf{PGen}, \mathsf{SGen})$, if given (pp, yk) generated from k using the generator, one cannot find a new pair (yk', k') such that (pp, yk', k') and (pp, yk', k'') belong to R_{LR} for some $k'' \in B_q(k)$. We show the κ-MAC construction in Fig.3 will be a q-fuzzy unforgeable, if the underlying sLRH relation is a q-fuzzy sLRH relation. The formal definition of the relation and the proof will be deferred to the full paper.

Lemma 7. *Let R_{LR} along with $(\mathsf{PGen}, \mathsf{SGen})$ be an $(n, \epsilon_{\mathsf{lr}})$-$q$-fuzzy sLRH relation w.r.t. $\mathbb{W}$ and $\mathbb{F}$. Let $\Pi = \{\mathsf{S}_{\mathsf{zk}}, \mathsf{P}_{\mathsf{zk}}, \mathsf{V}_{\mathsf{zk}}\}$ be a NIZK for the relation $\bar{R}_{\mathsf{LR}}$ satisfying ϵ_{zk}-ZK and $(\epsilon_{\mathsf{tse1}}, \epsilon_{\mathsf{tse2}})$-tSE. Then, the construction in Fig.3 satisfies $(n, q, \epsilon_{\mathsf{unf}})$ one-time fuzzy-unforgeability w.r.t. $\mathbb{W}$ and $\mathbb{F}$, for any $\epsilon_{\mathsf{unf}} > \epsilon_{\mathsf{zk}} + \epsilon_{\mathsf{tse1}} + \epsilon_{\mathsf{tse2}} + \epsilon_{\mathsf{lr}}$.*

Fuzzy sLRH relation from collision-resistant relation with private generator. For a "collision-resistant" sLRH relation, the adversary can "frame" some k'' only when she finds k''. If given (pp, yk) finding $k'' \in B_q^t$ is hard, then the relation will be a q-fuzzy sLRH relation. We argue when we can have the latter property from the privacy of the generator.

Note that the privacy of generator cannot ensure that (pp, yk) hides *all* partial information about k, as (pp, yk) itself must be non-trivial about k. Actually, the privacy ensures that adversaries cannot learn anything which is useful for deciding that yk is either generated by using the leaked key k or using an independent key. Then, for small q such that $B_q(k)$ only contains polynomial points, $k'' \in B_q(k)$ is surely hard-to-find from (pp, yk). However, for large q such that $B_q(k)$ could contain super-polynomial points, this argument does not apply.

We overcome this challenge by observing the task of recovering k from k'' can be done with the help of $2t$-secure sketch. More specifically, assume an adversary can recover k'' from (pp, yk). Then, the distinguisher specifies the leakage as a $2t$-secure sketch, invokes the adversary to have this $k'' \in B_{2t}(k)$, and converts k'' to k with the help of the secure sketch. We establish the following theorem, whose analysis is in the full paper.

Theorem 4. *Let* $(\mathsf{PGen}, \mathsf{SGen})$ *be a* $(n, \epsilon_{\mathsf{pr}}, \epsilon_{\mathsf{hv}})$*-private generator for an NP relation* R_{LR}*, and let* R_{LR} *be* $(n, \epsilon_{\mathsf{cr}})$*-collision-resistant w.r.t.* PGen*. Then* R_{LR} *along with* $(\mathsf{PGen}, \mathsf{SGen})$ *will be a* $(n, q, \epsilon_{\mathsf{lr}})$*-fuzzy sLRH relation, for any* $\epsilon_{\mathsf{lr}} > \epsilon_{\mathsf{pr}} + \epsilon_{\mathsf{cr}}$*, w.r.t.* $\mathbb{W}$ *and* $\mathbb{F}$ *which satisfy the following conditions. (1) There is a* q*-secure sketch* $\{\mathsf{SS}, \mathsf{Rec}\}$ *for each* $\mathcal{W} \in \mathbb{W}$*. (2)For each* $f \in \mathbb{F}$*, there is a one-way permutation* g*, and define* $\widetilde{f} = (f, \mathsf{SS}, g)$*. Then* $\widetilde{f}$ *is* ϵ_{hv}*-hard-to-invert w.r.t. every* $\mathcal{W}$*.*

Constructing robust fuzzy extractors. For a robust fuzzy extractor with t-error tolerance, we use a $2t$-fuzzy unforgeable κ-MAC to authenticate the helper string of a fuzzy extractor with t-error tolerance. Note the helper string along with the extracted randomness forms the auxiliary input $f(w)$ of the κ-MAC, our $2t$-fuzzy unforgeable κ-MAC construction allows an auxiliary input function f when f together with a $2t$-secure sketch forms a hard-to-invert leakage. Therefore, although a t-secure sketch is sufficient for constructing a fuzzy extractor with t-error tolerance, we will use a $2t$-secure sketch instead, such that $f(w)$ along with a $2t$-secure sketch must be hard-to-invert.

Let $\{\mathsf{SS}, \mathsf{Rec}\}$ be a $2t$-secure sketch, $\Sigma = \{\mathsf{Init}, \mathsf{Tag}, \mathsf{Verify}\}$ be a κ-MAC with $2t$-fuzzy unforgeability, and Ext be a strong extractor. We present the detailed construction of robust fuzzy extractor in Fig.4.

Regarding security, we present the following theorem whose formal proof will be in the full paper.

```
$\mathsf{CRS}(1^\lambda)$
    $\mathtt{crs} \leftarrow \mathsf{Init}(1^\lambda)$
    return $\mathtt{crs}$
```

```
$\mathsf{Gen}(\mathtt{crs}, w)$
    $\mathsf{ss} \leftarrow \mathsf{SS}(w)$
    $i \leftarrow\!\$\, \{0,1\}^s, r \leftarrow \mathsf{Ext}(w, i)$
    $\varsigma \leftarrow \mathsf{Tag}(\mathtt{crs}, w, (\mathsf{ss}, i))$
    return $R = r, P = (\mathsf{ss}, i, \varsigma)$
```

```
$\mathsf{Rep}(\mathtt{crs}, w', P)$
    $w'' \leftarrow \mathsf{Rec}(\mathsf{ss}, w')$
    return $R \leftarrow \mathsf{Ext}(w'', i)$, if
        $\mathsf{dist}(w'', w') \le t$
        $\mathsf{Verify}(\mathtt{crs}, w'', (\mathsf{ss}, i), \varsigma) = 1$
    return $\perp$
```

Fig. 4. Robust fuzzy extractor from randomness extractor + secure sketch+ κ-MAC

Theorem 5. *Assume* $\{\mathsf{SS}, \mathsf{Rec}\}$ *is an* $(\mathcal{M}, k, k', 2t)$*-secure sketch scheme,* Ext *is an* (n, k', ℓ)*-strong extractor with* ϵ_{ext}*-privacy, and* Σ *is a* κ*-MAC with* $(n, 2t, \epsilon_{\mathsf{unf}})$*-fuzzy unforgeability w.r.t.* $\mathbb{W}$ *and* $\mathbb{F}$ *and* $(n, \epsilon_{\mathsf{kpriv}}, \epsilon_{\mathsf{hv}})$*. Then, if* $\mathbb{W}$ *is all* (n, k)*-sources,* $\mathbb{F}$ *contains function ensembles implementing* SS*, and* $\epsilon_{\mathsf{ext}} < \epsilon_{\mathsf{hv}}$*, the construction in Fig.4 is an* $(\mathcal{M}, k, \ell, t)$*-robust fuzzy extractor with perfect correctness,* ϵ*-privacy and* δ*-robustness, for any* $\epsilon > \epsilon_{\mathsf{ext}} + 2\epsilon_{\mathsf{kpriv}}$ *and* $\delta > \epsilon_{\mathsf{unf}}$*.*

7 Conclusion and Open Problems

We give the first CRS-dependent (fuzzy) robust extractors with minimal min-entropy requirement (super-logarithmic) on the source, in the computational setting. They close the major gap left by the state-of-the-art robust extractors which require a linear fraction. Along the way, we formulate a new primitive κ-MAC.

We believe our new robust extractors (and our new tool of κ-MAC) could have broader applications. Also, converting other fuzzy extractors (not from secure sketch) into robust fuzzy extractors may be applicable to more general sources. We leave them all as interesting open problems.

References

1. Aggarwal, D., Obremski, M., Ribeiro, J.L., Simkin, M., Siniscalchi, L.: Two-source non-malleable extractors and applications to privacy amplification with tamperable memory. IACR Cryptol. ePrint Arch. 2020, 1371 (2020)
2. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: STOC. pp. 103–112. ACM (1988)
3. Boyen, X.: Reusable cryptographic fuzzy extractors. In: ACM Conference on Computer and Communications Security. pp. 82–91. ACM (2004)
4. Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., Smith, A.D.: Secure remote authentication using biometric data. In: EUROCRYPT. vol. 3494, pp. 147–163. Springer (2005)

5. Brakerski, Z., Segev, G.: Better security for deterministic public-key encryption: The auxiliary-input setting. In: CRYPTO. vol. 6841, pp. 543–560. Springer (2011)
6. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: STOC. pp. 209–218. ACM (1998)
7. Cramer, R., Dodis, Y., Fehr, S., Padró, C., Wichs, D.: Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 4965, pp. 471–488. Springer (2008)
8. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient public-key cryptography in the presence of key leakage. In: ASIACRYPT. Lecture Notes in Computer Science, vol. 6477, pp. 613–631. Springer (2010)
9. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: STOC. pp. 621–630. ACM (2009)
10. Dodis, Y., Katz, J., Reyzin, L., Smith, A.D.: Robust fuzzy extractors and authenticated key agreement from close secrets. In: CRYPTO. Lecture Notes in Computer Science, vol. 4117, pp. 232–250. Springer (2006)
11. Dodis, Y., Reyzin, L., Smith, A.D.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 3027, pp. 523–540. Springer (2004)
12. Dodis, Y., Wichs, D.: Non-malleable extractors and symmetric key cryptography from weak secrets. In: STOC. pp. 601–610. ACM (2009)
13. Faust, S., Hazay, C., Nielsen, J.B., Nordholt, P.S., Zottarel, A.: Signature schemes secure against hard-to-invert leakage. In: ASIACRYPT. vol. 7658, pp. 98–115. Springer (2012)
14. Fuller, B., Reyzin, L., Smith, A.D.: When are fuzzy extractors possible? In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 10031, pp. 277–306 (2016)
15. Garg, A., Kalai, Y.T., Khurana, D.: Low error efficient computational extractors in the CRS model. In: EUROCRYPT (1). vol. 12105, pp. 373–402. Springer (2020)
16. Kanukurthi, B., Reyzin, L.: An improved robust fuzzy extractor. In: SCN. Lecture Notes in Computer Science, vol. 5229, pp. 156–171. Springer (2008)
17. Katz, J., Vaikuntanathan, V.: Signature schemes with bounded leakage resilience. In: ASIACRYPT. vol. 5912, pp. 703–720. Springer (2009)
18. Santis, A.D., Crescenzo, G.D., Ostrovsky, R., Persiano, G., Sahai, A.: Robust non-interactive zero knowledge. In: CRYPTO. Lecture Notes in Computer Science, vol. 2139, pp. 566–598. Springer (2001)
19. Wen, Y., Liu, S.: Robustly reusable fuzzy extractor from standard assumptions. In: ASIACRYPT (3). Lecture Notes in Computer Science, vol. 11274, pp. 459–489. Springer (2018)
20. Wen, Y., Liu, S., Gu, D.: Generic constructions of robustly reusable fuzzy extractor. In: Public Key Cryptography (2). Lecture Notes in Computer Science, vol. 11443, pp. 349–378. Springer (2019)
21. Wen, Y., Liu, S., Han, S.: Reusable fuzzy extractor from the decisional diffie-hellman assumption. Des. Codes Cryptogr. 86(11), 2495–2512 (2018)
22. Wen, Y., Liu, S., Hu, Z., Han, S.: Computational robust fuzzy extractor. Comput. J. 61(12), 1794–1805 (2018)
23. Yuen, T.H., Yiu, S., Hui, L.C.K.: Fully leakage-resilient signatures with auxiliary inputs. In: ACISP. vol. 7372, pp. 294–307. Springer (2012)
24. Zhandry, M.: On elfs, deterministic encryption, and correlated-input security. In: EUROCRYPT (3). vol. 11478, pp. 3–32. Springer (2019)