Leakage-Resilient Tweakable Encryption from One-Way Functions

Suvradip Chakraborty[1], Chester Rebeiro[1], Debdeep Mukhopadhyay[2] and C. Pandu Rangan[1]

[1] Department of Computer Science and Engineering,
Indian Institute of Technology Madras, India
{suvradip,chester,rangan}@cse.iitm.ac.in
[2] Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur
debdeep@cse.iitkgp.ernet.in

Abstract. In this paper, we initiate the study of leakage-resilient tweakable encryption schemes in the relative key-leakage model, where the adversary can obtain (arbitrary) partial information about the secret key. We also focus on the minimal and generic assumptions needed to construct such a primitive. Interestingly, we show provably secure constructions of leakage-resilient (LR) tweakable encryption based on the sole assumption that one-way functions (OWF) exist via some interesting intermediate generic connections. A central tool used in our construction of LR-tweakable encryption is the notion of Symmetric-key tweakable weak hash proof system, which we introduce. This can be seen as a generalization of the Symmetric-key weak hash proof framework of Hazay et. al (Eurocrypt'13). Along the way, we also introduce a new primitive called tweakable weak pseudo-random functions (t-wPRF) and show how to generically construct it from weak-PRF. We then construct LR-version of t-wPRF and use it to construct LR-tweakable encryption.

Keywords: Tweakable-weak PRF, symmetric key tweakable weak hash proof system, tweakable encryption, relative-leakage model, after-the-fact leakage, one-way function

1 Introduction

Tweakable encryption is similar to a normal symmetric-key encryption scheme in the sense that it takes the usual inputs - message and cryptographic keys along with a third input, the "tweak". A tweakable encryption scheme can be constructed from a tweakable PRP/PRF generically as shown in [30], [18]. A tweakable PRF (t-PRF) extends the viewpoint of a normal PRF by adding in a second dimension called "tweak" in addition to the "key" of a PRF. There is a semantic asymmetry between the key and the tweak: the key is private whereas the tweak may be public, giving rise to variability. Tweakable encryption finds interesting applications in popular disk encryption mechanisms and format preserving encryption schemes [30]. In case of disk encryption, the tweak can be interpreted as the sector number holding the encrypted message. Since the sector numbers vary, encryption of the same message under different sectors/tweaks are different giving rise to variability.

Another application of tweakable encryption is in countering side-channel attacks [15,27]. These are implementation based attacks, where unintended information may leak through side channels such as timing, power, faults, electromagnetic radiation, memory, and many more [16, 21, 22]. These attacks enable an adversary to learn much more information about the internal state of the system than that predicted by the system designer. Tweakable block cipher constructions to counter side-channel attacks exist, that are also efficient to implement in hardware and software. However, a major limitation of these schemes is that they are proven secure only under a particular class of side-channel attacks, rather than provably resisting against a broad class of known or even future side-channel attacks. In [15] for instance, the leakage is assumed to take the form of Hamming weight of intermediate data. The information theoretic proofs show that the scheme is resilient to first order differential power attacks [21]. A later publication [27] showed resistance to differential fault attacks as well. So, these constructions can guarantee security only under certain (restricted) classes of side-channel attacks, and hence its usefulness against other broader classes of side-channel attacks are not proven.

In this paper, we provide a more *general* construction of a leakage-resilient tweakable encryption scheme that is provably capable of tolerating a wide range of side-channel attacks. We base our construction on the *key-leakage attack model* that was first formalized by Akavia, Goldwasser and Vaikuntanathan [2]. In this model, it is assumed that the leakage ℓ can be any arbitrary efficiently computable length-shrinking function of the secret key. In other words, the adversary can learn any arbitrary information (leakage) about the secret state of the system, the only restriction being an upper bound on the size of leakage. The leakage bound that the scheme can tolerate is fixed a priori. Theoretically, this is modeled by giving the adversary access to a leakage oracle which can be queried by the attacker using any arbitrary polynomial time computable function $f_i : \{0,1\}^* \rightarrow \{0,1\}^{\lambda_i}$, and he can learn the value of the function f_i applied to the internal state of the system. Note that no restriction is placed on the type of functions the adversary submits to the oracle, instead it bounds the amount of leakage. From a practical standpoint, this model captures a wide range of side-channel attacks including power attacks, memory attacks, timing attacks, and electro-magnetic radiation based attacks. It also captures the "cold boot attack" of Halderman *et. al.* [16], where the adversary can learn a noisy version of the secret key stored in memory. Usually, the leakage bound ℓ of the system is also related to the size of the secret key $|s|$, and we can define the *relative leakage rate* to be the fraction of secret key bits leaked, i.e., $\frac{\ell}{|s|}$.

We concentrate on the *minimal assumptions* required to build leakage-resilient tweakable encryption schemes. Much research in theoretical cryptography has been centered around finding the weakest possible assumptions required to construct major cryptographic primitives. Ever since the introduction to modern cryptography by Diffie and Hellman [8], one of the central goal in the cryptographic research community is to base cryptosystems on assumptions that are as weak and generic as possible, possibly one-way functions and one-way trapdoor functions. In this paper, we also follow this line of research and attempt to construct leakage-resilient tweakable encryption from minimal assumptions, in particular from the sole assumption that one-way function exists. We concentrate mainly on the setting of *key leakage attacks* or *memory attacks* and its *bounded-leakage variant*, although later we consider several generalizations of this framework to account for more general leakage settings.

We first construct a *leakage-resilient tweakable weak PRF/PRP* (a notion we introduce) and use it to construct leakage-resilient tweakable encryption. Interestingly, we show that the existence of one-way functions imply the existence of leakage-resilient tweakable weak PRF/PRPs via some interesting intermediate generic connections. Specifically, we generalize the symmetric-key hash proof system (HPS) of Hazay *et. al.* [19] and introduce a new HPS, which we call *Symmetric-key tweakable weak HPS* and use it to construct leakage-resilient tweakable encryption. The envisioned attack-model we consider is a chosen plaintext attack: the adversary can learn the ciphertext C for any plaintext P and tweak T, and in the challenge phase it has to distinguish between the encryption of two adversarially-chosen message-tweak pairs with non-negligible probability, doing which we say the adversary *wins* the game.

Related Works. Although many recent works address the problem of leakage-resilience in the context public key encryption [2–4, 6, 7, 10, 19, 20, 25], not much work has been reported in the context of symmetric-key cryptography. Dziembowski and Pietrzak [13] proposed the first construction of leakage-resilient stream cipher in the Only Computation Leaks Information (OCLI) axiom of Micali and Reyzin [24]. This construction was later simplified in [28] using a weak PRF. Later Faust *et. al.* [14] constructed leakage-resilient block cipher using this PRF. However, their overall construction was shown to be inefficient by Bernstein in the rump session of CHES'12 [29]. Hazay *et. al.* [19] showed the first construction of a generic leakage-resilient symmetric key encryption from any weak PRF and hence relying only on the assumption that one-way function exists. Besides their constructions addressed the more generalized setting of relative leakage model, as ours. Abdalla, Belaid and Fouque [1] later constructed a more practical and efficient symmetric key encryption scheme by introducing a new leakage-resilient re-keying technique. They instantiated the re-keying scheme with a secure AES block cipher. However, the encryption scheme is only secure against non-adaptive leakage functions, where the adversary has to specify the leakage function at the beginning of the security game.

Paper Organization. The paper is organized as follows. In section 2, we list down our contributions. In section 2.1, we give high level overview of our techniques. In section 3, we give the preliminaries required for our paper. In section 4 we give the detailed construction of our leakage-resilient tweakable encryption. In section 5 we combine all our results together and state the main theorem. In section 6 we show extensions of our construction to more generalized leakage settings. Finally section 7 concludes our paper.

2 Our Contributions

As our main contribution, we construct the *first* leakage-resilient tweakable encryption scheme in the relative leakage model. Along the way, we develop new notions and get results of independent interest. Our results are summarized as follows:

1. Revisit the framework of *key leakage attack* introduced by Akavia *et al.* [2] in the setting of symmetric-key encryption, particularly in the context of *tweakable encryption.*
2. Introduce a new definition of *leakage-resilient tweakable weak pseudo-random functions* (LR-twPRF). In the leakage-free (standard) setting this also introduces a new primitive namely *tweakable weak PRF* (t-wPRF) (or 0-leakage t-wPRF), which is a natural relaxation of the definition of tweakable PRF (similar to the relaxation in the definition of weak PRF (wPRF) over (standard) PRF).
3. Show two *generic* constructions of tweakable weak PRF (t-wPRF) from any weak PRF. The first construction is trivial and is not very efficient. More precisely,

it requires $2n$ invocations of the underlying wPRF to encrypt n blocks of the tweakable PRF. Our second construction is efficient and it can be viewed as a simplification or relaxed version of the XTS mode of operation [30] (XTS mode of operation is used to construct (standard) tweakable PRF starting from a (standard) PRF) which we call *simplified* XTS mode of operation. This mode only requires n invocations of the underlying wPRF to encrypt n blocks as opposed to $(n + 1)$ invocations in case of XTS.

4. Introduce a new notion of *Symmetric-key tweakable weak hash proof system* (S-twHPS), which is the central tool in constructing and analyzing our leakage-resilient tweakable weak PRF (LR-twPRF) and leakage-resilient tweakable encryption. Note that, in the *non-leakage* setting, it is indeed possible to construct a tweakable weak PRF (tw-PRF) from a (standard) weak PRF (wPRF) as we show in this paper (section 4.2). However, in the presence of leakage it is not clear whether it is possible to directly construct a LR-twPRF from a LR-wPRF, without making further assumptions on the leakage model (more on this under section 2.1). Hence, we have to introduce our new hash proof system (S-twHPS) and construct a LR-twPRF from a S-twHPS following an indirect route, without making any additional restrictions/assumptions.
5. Show how to construct our Symmetric-key tweakable hash proof system (S-twHPS) *generically* starting from any (standard/ 0-LR) tweakable weak PRF (t-wPRF). This also implies we can construct S-twHPS starting from any weak-PRF (wPRF). However the leakage rate our initial construction is rather poor.
6. Finally, we amplify the leakage rate of our construction by *parallel repetition.* Although there are some counterexamples against this leakage-amplification technique, we argue that our ideas/technique essentially bypass those arguments and we can amplify the tolerable leakage bound of our construction to any arbitrary polynomial in the security parameter.
7. *Extend* our constructions to more general settings than relative-leakage model namely *Entropy-bounded leakage* and *After-the-Fact leakage* settings. We show that our constructions meet these new (generalized) security definitions in the context of tweakable encryption.

So putting these all-together we get the following connections as shown below. This essentially says that given a one-way function (OWF) we can construct a leakage-resilient tweakable encryption scheme via some intermediate generic transformations.

$$\begin{tikzcd} \boxed{\text{OWF}} \arrow[r, mapsto] & \boxed{\text{w-PRF}} \arrow[r, mapsto] & \boxed{\text{t-wPRF}} \arrow[r, mapsto] & \boxed{\text{S-twHPS}} \arrow[r, mapsto] & \boxed{\text{LR-twPRF}} \arrow[r, mapsto] & \boxed{\text{LR-tweakable Enc.}} \end{tikzcd}$$

2.1 Overview of our Results

An encryption scheme is said to be *secure* in the relative leakage model if it is *semantically secure* even given these key leakages. We revisit this framework of key-leakage attacks in the context of symmetric-key tweakable encryption. Informally, a tweakable encryption scheme is said to be secure in this framework of relative-leakage if it is semantically secure (with access to the encryption oracle) with respect to message-tweak pairs, even given partial leakage from the secret key of the tweakable encryption scheme.

A generic construction. A central tool used in our construction of leakage-resilient tweakable encryption is the notion of *Symmetric-key tweakable weak hash proof system* (S-twHPS), that we introduce. Our notion of symmetric key tweakable hash proof system can be seen as a generalization of the symmetric key weak hash proof system (S-wHPS) introduced by Hazay *et. al.* [19]. Informally, a symmetric-key tweakable wHPS can be thought of as a tweakable weak pseudo-random function $E_K(T, .)$[3] with some special properties.

- We define *four* distributions: two pair of *valid* and *invalid* distribution– one pair defined on the *input space* of the t-wPRF and another pair defined on the *tweak space* of the t-wPRF, in addition to the two normal distributions corresponding to sampling the elements of the input space and tweak space uniformly at random. We require that samples from *like-joint distribution* of the input and the tweak are indistinguishable even when given the secret key K. We say a joint distribution to be *like-joint distribution* when both the samples of the joint distribution are sampled from similar distributions (in our case either both are sampled from *valid* or both from *invalid* or both uniformly at random). We refer to this property as the "*joint-input indistinguishability*" property of the S-twHPS.
- Given multiple input-output tuples namely, $\{X_i, T_i, E_K(T_i, X_i)\}$ for various random *valid* X and *valid* T, and a random choice of an *invalid* X^* and an *invalid* T^*, the output $E_K(T^*, X^*)$ is *uniformly random* and *independent*, where the randomness comes from the choice of the secret key K. In other words, there are many possible secret keys corresponding to $(X^*, T^*, E_K(T^*, X^*))$. We refer to this property as the *smoothness* property of the underlying S-twHPS.

In other words, the secret key maintains real entropy even conditioned on seeing many valid $(X_i, T_i, E_K(T_i, X_i))$ tuples, and this entropy is transfered to the output $y^* = E_K(T^*, X^*)$, on a random invalid X^* and a random invalid T^*.

Achieving leakage-resilience via S-twHPS. We show how to achieve leakage-resilient tweakable encryption using symmetric-key twHPS as the basic building block. It relies on no other computational assumptions other than the existence of such a S-twHPS. Our construction closely follows the approach of Naor and Segev [25] and Hazay *et. al.* [19] while introducing the additional second dimension namely the "tweak". The main idea of the construction is to evaulate the tweakable weak PRF (t-wPRF) on random input-tweak pair (X, T), and apply the strong average-case extractor to the output of the t-wPRF. In the proof of security, we let the attacker see many *valid* evaluations of the t-wPRF on random (X, T)-pairs. In the challenge phase, we change the distribution of the input-tweak pair (X^*, T^*) from *valid* to *invalid.* Note that by the input indistinguishability property, this change is oblivious to the adversary. Then, we argue by the smoothness property of S-twHPS that the $y^* = E_K(T^*, X^*)$ is uniform and independent (information-theoretically). So, even if the adversary observes some bounded leakage from the secret key, it does not reduce the entropy of y^* by much, if y^* is sufficiently large. Finally we transform the output to a uniformly random value using a average-case strong extractor whose output is statistically close to the uniform distribution defined over the appropriate domain.

Necessity of the Symmetric-key tweakable HPS: For our construction of LR-tweakable encryption, we introduce Symmetric-key tweakable hash proof system (S-twHPS) as a central tool. The main idea is to give a generic construction of a leakage-resilient tweakable weak PRF (LR-twPRF) from the above S-twHPS in the

[3] Informally, a tweakable weak PRF can be thought of as a weak PRF by adding in a second dimension called "tweak" in addition to the "key" of a weak PRF. Please refer to section 4 for the formal definition.

bounded memory leakage model and then show how to construct a (bounded) LR-tweakable encryption from such a LR-twPRF. However, one may also try to construct a LR-tweakable weak PRF (LR-twPRF) directly given a LR-weak PRF. In fact, in the non-leakage setting there are well-known constructions of tweakable-PRFs from standard PRFs [18, 23]. In this work, we also show black-box constructions of tweakable weak-PRF (t-wPRF) from weak-PRF (w-PRF) in the non-leakage setting. We also know how to construct LR-wPRF by the ideas of Hazay et. al. [19]. So, a natural question is that : given a LR-wPRF (in the bounded memory leakage model) can we construct a LR-twPRF in the same leakage model? If one can come up with such a transformation, then there is no need to introduce the S-twHPS, and it is indeed possible to get a more direct construction of LR-tweakable encryption than our construction.

However, given the current constructions of t-wPRF from wPRF, it is not immediately clear how to port these constructions in the setting of bounded (non-split state) memory leakage model. In particular, in the XTS mode of operation [30], the secret key comprises of two sub-keys K_1 and K_2. Now, in our model of bounded memory leakage, the adversary obtains leakage from both K_1 and K_2. Now even if we replace the normal PRF F with a leakage-resilient version of that, it is not clear whether the security of the final XTS mode of operation holds under joint leakage from both the keys. The leakage from the right sub-key K_2 and from the left sub-key K_1 can independently be analyzed using leakage-resilient PRFs; however, the effect of global leakage *simultaneously* from both K_1 and K_2 seems difficult to analyze. One solution would be to rely on the *split-state* assumption, where the adversary can get access to leakage from each sub-key separately, but not to a global leakage from the entire secret state. However, this necessarily weakens the leakage model in contrast to the model we are considering. Besides, the split-state leakage model also does not capture various classes of well-known side-channel attacks like the Hamming-weight attacks. To get around this problem, we *propose* the new notion of Symmetric-key tweakable HPS as a central tool to analyze our construction of leakage-resilient tweakable weak PRF.

Constructing Tweakable weak PRF. As already mentioned, our symmetric-key tweakable weak HPS can be viewed as a tweakable weak PRF (t-wPRF) with some special properties. To this end, we first define tweakable weak PRF. Basically this is a relaxation of the notion of tweakable PRFs (t-PRF), where instead of adversarially-chosen inputs and tweaks, the input-tweak pairs are randomly chosen by the challenger and their evaluations are given to the adversary. The indistinguishability requirement remains the same as in normal t-PRFs, namely the adversary should be able to tell apart whether it is interacting with a truly random function or a pseudorandom function. We present generic constructions of t-wPRFs using any weak PRF (w-PRF). In particular, we show that a *simplified* variant of the XTS mode of operation suffices to construct t-wPRFs. Using this simplified XTS mode, to encrypt n blocks it requires only n invocations of the underlying wPRF. In the normal XTS mode of operation (which is used to construct a tweakable PRF from a weak-PRF), the tweak is encrypted using a symmetric-key encryption scheme in the beginning and then it involves n invocations of the underlying wPRF; thus requiring a total of $(n + 1)$-invocations of the wPRF. If the tweak is used in un-encrypted form in the construction, then it leads to attacks on the scheme as shown in Section 4.2. However, we observe that for constructing tweakable weak PRFs, this first step of encrypting the tweak is not required, thus requiring only n invocations of the weak PRF. This is possible because of the *weaker* requirement of t-wPRF as compared to t-PRF.

Constructing Symmetric-key t-wHPS. We construct Symmetric-key tweakable wHPS generically from a tweakable weak PRF (t-wPRF). As a warm-up construction, we show how to construct a single-bit S-twHPS using a t-wPRF and a CPA-secure symmetric key encryption (which can also be generically constructed given a w-PRF very easily as shown in Sec 4.3).

1. Run the KeyGen algorithm of the t-wPRF to generate two secret keys , i.e., $K_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $K_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$. The secret key of the scheme is set to $K = (b, K_b)$, where $b \leftarrow \{0, 1\}$.
2. The *valid* sampling algorithm defined over the tweak space $\mathcal{T}$ of the S-twHPS is defined as follows: Run the encryption algorithm Enc of the CPA-secure symmetric key encryption and set $\hat{t} = (\hat{t}_0, \hat{t}_1)$, where $\hat{t}_0 = \mathsf{Enc}_{K_0}(t)$, and $\hat{t}_1 = \mathsf{Enc}_{K_1}(t)$, where $t \in \{0, 1\}$ (i.e., both encrypt the *same* value t).
3. The *invalid* sampling algorithm defined over the tweak space $\mathcal{T}$ of the S-twHPS is defined as follows: $\hat{t} = (\hat{t}_0, \hat{t}_1)$, where $\hat{t}_0 = \mathsf{Enc}_{K_0}(t)$, and $\hat{t}_1 = \mathsf{Enc}_{K_1}(1 - t)$, where $t \in \{0, 1\}$ (i.e., both encrypt *different* values t and $(1 - t)$ respectively).
4. The *valid* sampling algorithm defined over the input space $\mathcal{D}$ of the S-twHPS is defined as follows: Run the encryption algorithm Enc of the CPA-secure tweakable encryption and set $C = (C_0, C_1)$, where $C_0 = \mathsf{Enc}_{K_0}(\hat{t}_0, r)$, and $C_1 = \mathsf{Enc}_{K_1}(\hat{t}_1, r)$, where $r \in \{0, 1\}$, and $\hat{t}_0$, $\hat{t}_1$ are sampled from *valid* distributions (i.e., both encrypt the same value t), and hence both C_0 and C_1 encrypt the *same* value.
5. The *invalid* sampling algorithm defined over the input space $\mathcal{D}$ of the S-twHPS is defined as follows: Run the encryption algorithm Enc of the CPA-secure tweakable encryption and set $C = (C_0, C_1)$, where $C_0 = \mathsf{Enc}_{K_0}(\hat{t}_0, r)$, and $C_1 = \mathsf{Enc}_{K_1}(\hat{t}_1, 1 - r)$, where $r \in \{0, 1\}$, and $\hat{t}_0$, $\hat{t}_1$ are sampled from *invalid* distributions and hence both C_0 and C_1 encrypt the *different* values.

The *joint-input indistinguishability* of the above construction follows from the CPA-security of the symmetric-key encryption and also CPA-security of the tweakable encryption even given the secret key (b, K_b). The *smoothness* property follows since the decryption of a random ciphertext $C^* = (C_0^*, C_1^*)$ is uniformly random and independent over the choice of the secret key bit b.

Leakage Amplification via parallel repetition. The above construction yields a S-twHPS with 1-bit output. However, we can easily amplify the output size of the S-twHPS to any arbitrary polynomial $m = m(\lambda)$, simply by doing parallel repetition of the above scheme. In particular, we run concurrently m independent copies of the scheme in parallel. This amplifies the output size of the S-twHPS by a factor of m. The secret key of the new construction has 2^m many possibilities. Since the amount of leakage is roughly equal to the size of the output of S-twHPS, we can amplify the leakage amount to any arbitrary polynomial $m = m(\lambda)$. At first, this might seem as a contradiction to the fact that parallel repetition does not always amplify the leakage as shown in [3–5]. However, we argue that we can bypass these counterexamples as we are not directly amplifying the leakage amount. Instead, we amplify the output space of the S-twHPS and since the leakage amount is related to the output size of the S-twHPS, this also implicitly amplifies the tolerable amount of leakage of the new construction.

Leakage tolerated by our construction. Our scheme can tolerate an arbitrarily large amount of absolute leakage ℓ. However, the relative leakage rate of our construction is rather poor. In particular, the leakage rate of our construction is

$O(\ell(\lambda)/s(\lambda))$, where $s(\lambda)$ is the size of the secret key and λ is the security parameter. We leave open the problem of constructing leakage-resilient tweakable encryption schemes under general assumptions with higher leakage rates.

Extensions to more generalized leakage settings. The leakage model we have considered so far is length-bounded leakage. So it restricts the output of all the leakage function to be upper bounded by the length of the secret key. So we consider more generalized leakage settings than the relative leakage model. In particular we consider the setting of *entropy bounded leakage*, where the length of the leakage function may exceed even the length of the secret key, but the only requirement is that the secret key should maintain enough min-entropy even given such leakage. Our construction of leakage-resilient tweakable encryption also satisfies this definition in the presence of entropy-bounded leakage. Next we extend the framework of tweakable encryption in the relative-leakage model to *After-the-Fact* leakage model, where the adversary can even get access to the leakage oracle after receiving the challenge ciphertext. We provide a new definition of tweakable encryption under this new leakage model and show that our construction also satisfies this notion of security. These are illustrated in Sections **6.1** and **6.2** respectively.

It will be interesting to come up with new constructions of tweakable encryption schemes in other stronger leakage models like the continuous memory leakage model [9] or auxiliary input leakage model [11].

Applicability of leakage-resilient tweakable encryption and further directions. One of the main applications of tweakable encryption is that it can be used to construct *format preserving encryption* (FPE) and *full disk encryption* (FDE) schemes. It will be worth exploring the connections between leakage-resilient tweakable encryption and leakage-resilient versions of FPE and FDE schemes. For this, the current constructions of FPE and FDE from tweakable encryption may not suffice and we need to define new leakage-resilient modes of operation for the corresponding primitives. We leave this as an interesting open problem for further investigation.

3 Preliminaries

In this section we provide some basic notations, definitions and tools used in our construction.

Notations. Throughout this work, we denote the security parameter by λ. We assume that all the algorithms take as input (implicitly) the security parameter represented in unary, i.e., 1^λ. For an integer n, we use the notation $[n]$ to denote the set $[n] \overset{\text{def}}{=} \{1, \ldots, n\}$. For a randomized function f, we write $f(x; r)$ to denote the unique output of f on input x with random coins r. We write $f(x)$ to denote a random variable for the output of $f(x; r)$, over the random coins r. For a set S, we let U_S denote the uniform distribution over S. For an integer $r \in \mathbb{N}$, let U_r denote the uniform distribution over $\{0,1\}^r$, the bit strings of length r. For a distribution or random variable X, we denote $x \leftarrow X$ the action of sampling an element x according to X. For a set S, we write $s \leftarrow S$ to denote sampling s uniformly at random from the S, i.e. $s \leftarrow U_S$. The *statistical distance* between two random variables X and Y over a finite domain Ω is defined as $\mathrm{SD}(X,Y) = \frac{1}{2}\Sigma_{w\in\Omega}|\Pr[X = w] - \Pr[Y = w]|$. We say that two variables are ϵ-close, and write $X \approx_\epsilon Y$, if their statistical distance is at most ϵ. We write $X \equiv Y$ to mean that X and Y are identically distributed and $X \approx_c Y$ to mean that X and Y are computationally indistinguishable. We use $\mathrm{negl}(\lambda)$ to denote a function that vanishes faster than the inverse of any polynomial, i.e., it denotes the set of negligible functions $\mu(\lambda) = \lambda^{-\omega(1)}$. We denote an ensemble $\mathcal{X}$ as a collection of distributions $\{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$. We sometimes drop the subscript λ when clear from context and write $x \leftarrow \mathcal{X}$ instead of $x \leftarrow \mathcal{X}_\lambda$ to denote sampling an element x from $\mathcal{X}_\lambda$. For brevity, we will write $\overrightarrow{X}|_n$ to denote the vector $(X_1, X_2, \ldots, X_n)$ of length n. We assume familiarity with the basic notions of information theory, in particular the notions of min-entropy, average conditional min-entropy, and related results.

The Leakage Oracle. To model *key leakage* attacks, we provide the adversary access to a leakage oracle. The adversary can adaptively query this oracle with arbitrary leakage functions to learn information about the secret key or functions of the secret key. However, if we do not put any restriction on the type of leakage functions or the overall amount of leakage, the adversary can learn the entire secret key by querying the leakage oracle and then we cannot define any meaningful notion of security in the presence of such unrestricted leakages. In this work we do not put any restriction on the class of leakage functions, rather we limit the amount of leakage the adversary can get. More formally, the adversary submits any length-bounded arbitrary polynomial-time computable function $f_i : \{0,1\}^{|SK|} \to \{0,1\}^{\ell_i}$ to the leakage oracle $\mathcal{O}^{\ell}_{SK}(.)$, parameterized by a secret key SK and a leakage parameter ℓ and gets back as response $f_i(SK)$. The oracle keeps track of the output sizes ℓ_i of all the leakage queries so far, and only responds to the q^{th} leakage query if $\sum_{i=1}^{q} \ell_i \leq l$. In this way the length of the outputs given by the oracle is bounded by ℓ bits, where this leakage bound ℓ is pre-determined from the beginning. It is sometimes convenient to interpret the leakage function as a polynomial sized circuit (note that since the attacker is polynomial time bounded so it can query only polynomial size circuits to the leakage oracle), which in turn implies the leakage is poly-time computable. Note that our definition of *length-bounded* leakage naturally generalizes to the case of *noisy* leakage or *entropy-bounded* leakage, where the leakage is not of bounded length, but it is guaranteed that the secret key is still unpredictable given the leakage which is quantified by *min-entropy* of the secret key given the leakage information.

3.1 Entropy and Randomness Extraction

Definition 1. (Min-Entropy). The *min-entropy* of a random variable X, *denoted as* $\mathrm{H}_\infty(X)$ *is defined as* $\mathrm{H}_\infty(X) \overset{\text{def}}{=}$ -log$(\max_x \Pr[X = x])$.
This is a standard notion of entropy used in cryptography, since it measures the worst-case predictability of X.

Definition 2. (Average Conditional Min-Entropy). The *average-conditional min-entropy* of a random variable X conditioned on a (possibly) correlated variable Z, denoted as $\mathrm{H}_\infty(X|Z)$ is defined as

$$\mathrm{H}_\infty(X|Z) = \text{- log}\left(\mathbb{E}_{z\leftarrow Z}\ [\max_x \Pr[X = x|Z = z]\right) = \text{- log}\left(\mathbb{E}_{z\leftarrow Z}\ [2^{\mathrm{H}_\infty(X|Z=z)}]\right)$$

This measures the worst-case predictability of X by an adversary that may observe a correlated variable Z.

The following bound on average min-entropy was proved in [12].

Lemma 1. [12] *For any random variable X, Y and Z, if Y takes on values in* $\{0,1\}^\ell$, *then*

$$\widetilde{\mathrm{H}}_\infty(X|Y,Z) \geq \widetilde{\mathrm{H}}_\infty(X|Z) - \ell \quad \text{and} \quad \widetilde{\mathrm{H}}_\infty(X|Y) \geq \widetilde{\mathrm{H}}_\infty(X) - \ell$$

A main tool we require in our constructions is a *average-case strong randomness extractor*. An extractor [26] can be used to extract uniform random bits out of a weakly-random source which is only assumed to have sufficient min-entropy. The following definition naturally generalizes the standard definition of a strong extractor to the setting of average min-entropy as given in [12]:

Definition 3. (Randomness Extractor). We say that an efficient randomized function $\mathsf{Ext}\colon \mathcal{X} \times \mathcal{S} \to \mathcal{Y}$ is an (υ, ε)-extractor if for all (correlated) random variables X, Z such that the support of X is $\mathcal{X}$ and $\widetilde{\mathrm{H}}_\infty(X|Z) \geq \upsilon$, we get $(Z, S, \mathsf{Ext}(X; S)) \approx_\varepsilon (Z, S, U_\mathcal{Y})$, where S is uniform over $\mathcal{S}$, and $U_\mathcal{Y}$ denotes the uniform distribution over the range of the extractor $\mathcal{Y}$.

Dodis *et al.* [12] proved that any strong extractor is in fact an average-case strong extractor, for an appropriate setting of the parameters:

Lemma 2. [12] *For any $\delta > 0$, if Ext is a (worst-case) $(k - \log(1/\delta), \varepsilon)$-strong extractor, then Ext is also an average-case $(k, \varepsilon + \delta)$-strong extractor.*

In particular, they proved that any family of pairwise independent hash functions is an average-case strong extractor.

Lemma 3. [12, 26] Let $\mathcal{H} = \{h_s : \mathcal{X} \to \mathcal{Y}\}$ be a *universal family of hash functions* meaning that for all $x \neq x' \in \mathcal{X}$, we have $\Pr_{s \in \mathcal{S}}[h_s(x) = h_s(x')] \leq \frac{1}{|\mathcal{Y}|}$. Then $\mathsf{Ext}(x; s) \overset{\text{def}}{=} h_s(x)$, is a (υ, ε)- extractor for any parameter $\upsilon \geq \log|\mathcal{Y}| + 2\log(1/\varepsilon)$

4 Leakage Resilient Tweakable weak Pseudo-Random Functions

We begin by defining tweakable weak PRFs (t-wPRF). But first, let's recall the standard definition of tweakable PRF. A tweakable PRF (t-PRF) [23, 30] extends the viewpoint of a normal PRF by adding in a second dimension called "tweak" in addition to the "key" of a PRF. In particular a t-PRF $E : \mathcal{K} \times \mathcal{T} \times \mathcal{D} \to \mathcal{R}$ is a family of functions indexed by $(K, T) \in \mathcal{K} \times \mathcal{T}$ such that for every $K \in \mathcal{K}$ and $T \in \mathcal{T} \subseteq \{0, 1\}^*$, the mapping $E_K(T, .)$ defines a value over the set $\mathcal{R}$. The set $\mathcal{T}$ is called the "tweak space" of the t-PRF and the element $T \in \mathcal{T}$ is called the "tweak". Also recall a function family $\mathcal{F} = \{f_K\}_{K \in \mathcal{K}} : \mathcal{D} \to \mathcal{R}$ where $\mathcal{K}$ is the key space, is said to be a *weak* PRF family if for any polynomial-sized $n = n(\lambda)$, randomly chosen $f \in_R \mathcal{F}$ and $(x_1, x_2, \ldots x_n) \leftarrow \mathcal{D}^n$, the distribution $\{(x_i, f(x_i)) \mid i \in [n]\}$ is computationally indistinguishable from the uniform distribution over $(\mathcal{D}, \mathcal{R})^n$, i.e., an adversary for a weak-PRF aims to distinguish a random member of the family from a truly random function after observing a polynomially-bounded number of samples. A wPRF is called (t, Q, ϵ)-wPRF if for all t-time adversaries $\mathcal{A}$ making at most Q queries to the function, the advantage in distinguishing the above two distributions is at most ϵ.

Tweakable-weak PRF: We now introduce the notion of a *tweakable weak* PRF (t-wPRF). This essentially tells us that, given arbitrary many *uniformly random* inputs $(x_1, x_2, \ldots x_n) \leftarrow \mathcal{D}^n$ and $(T_1, T_2, \ldots T_n) \leftarrow \mathcal{T}^n$, the output of the t-wPRF $\{(x_i, T_i, y_i = E_k(T_i, x_i)) \mid i \in [n]\}$ look pseudorandom, in particular the above outputs should be computationally indistinguishable from the uniform distribution over $(\mathcal{D}, \mathcal{T}, \mathcal{R})^n$. This is in contrast to the actual definition of standard t-PRFs where the pseudorandomess requirement holds over *worst case (adversarial)* choice of input-tweak pairs (x_i, T_i). Our definition of leakage-resilient t-wPRF (LR-twPRF) requires the security of the t-wPRF to hold even if the attacker can get some information (which is quantified by giving the adversary access to the leakage oracle $\mathcal{O}^{\ell}_{SK}(.)$ as described earlier) about the secret key of t-wPRF. Our general definition of LR-t-wPRF also captures the standard security requirement of a t-wPRF in a non-leakage scenario (when $\ell = 0$).

Definition 4. Let $\mathcal{K}, \mathcal{D}, \mathcal{R}, \mathcal{T}$ denote some ensembles. We require that these ensembles be *efficiently samplable,* meaning that the operation of sampling an element say $x \leftarrow \mathcal{D}$ and testing whether $x \in \mathcal{D}$ can be performed in poly(λ) time. This should hold true for any of these ensembles. Let $\widetilde{\mathcal{E}} = \{E_K : \mathcal{T} \times \mathcal{D} \rightarrow \mathcal{R}\}_{K \in \mathcal{K}}$ be an *efficient* function family. We say E is an $\ell(\lambda)$-leakage resilient tweakable weak PRF (t-wPRF), if for all probabilistic polynomial time (PPT) adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game is negligible:

1. **Initialization:** *The challenger chooses a uniformly random $K \leftarrow \mathcal{K}_\lambda$ and proceeds with the game as follows.*
2. **Learning Stage:** *The attacker $\mathcal{A}^{\mathcal{O}^{\ell}_{K}, E_K(\$,\$)}$ gets oracle access to the leakage oracle $\mathcal{O}^{\ell}_{K}$ which allows him/her to learn at most ℓ bits of information about the secret key K of the* t-wPRF[4]*. Besides the adversary can also query the* t-wPRF *oracle $E_K(\$,\$)$. This oracle does not take any input, and on each invocation samples a uniformly random $X \leftarrow \mathcal{D}$ and also $T \leftarrow \mathcal{T}$ and outputs $(X, T, E_K(T, X))$.*
3. **Challenge Stage:** *The challenger chooses a bit $b \leftarrow \{0,1\}$, a random input $X^* \leftarrow \mathcal{D}$ and a random tweak $T^* \leftarrow \mathcal{T}$. If $b = 0$, it sets $Y^* = \widetilde{E}_K(T^*, X^*)$, and if $b = 1$ it chooses $Y^* \leftarrow \mathcal{R}$. The challenger gives (X^*, T^*, Y^*) to $\mathcal{A}$ who then outputs a bit b'.*

We define the advantage of the attacker $\mathcal{A}$ as $\mathrm{Adv}^{\text{LR-twPRF}}_{\mathcal{A}}(\lambda) = |\Pr[b' = b] - \frac{1}{2}|$.

We say that the LR-twPRF is $(t, q_{t\text{-}wPRF}, q_{leak}, \epsilon)$-secure if for all t-resource bounded adversaries making at most $q_{t\text{-}wPRF}$ t-wPRF oracle queries and at most q_{leak} leakage queries, $Adv^{\text{LR-twPRF}}_{\mathcal{A}}(\lambda) \leq \epsilon$.

Remark 1. Note that in the context of no leakage (i.e., when $\ell = 0$), the definition of LR-twPRF reduces to the definition of a t-wPRF.

Remark 2. Since the challenge points X^*, T^* needs to be *"fresh"*, the size of the input domains $|\mathcal{D}|$ and $|\mathcal{T}|$ should be super-polynomial. This in turn also ensures that the value $E_K(T^*, X^*)$ given out at the challenge phase is not given out in the learning phase.

Remark 3. Note that our definition can be further generalized to *multi-challenge* variant, where the challenge phase may consists of giving polynomially many challenge tuples of the form $(X^*_1, T^*_1, Y^*_1) \ldots, (X^*_q, T^*_q, Y^*_q)$, where $q = q(\lambda)$ is some polynomial and $Y^*_i = E_K(T^*_i, X^*_i)$, if $b = 0$, otherwise $Y^*_i \leftarrow \mathcal{R}$. This requires the output space $|\mathcal{R}|$ also to be super polynomial.

[4] Note that w.lo.g., we can assume the attacker makes a single call to the leakage oracle $\mathcal{O}^{\ell}_{K}(.)$ (since we can encode the adaptive behavior of the adversary in a function), after making all the calls to the t-wPRF oracle $E_K(\$,\$)$.

4.1 Leakage Resilient Tweakable Encryption

Let $\mathcal{M}$, $\mathcal{K}$, $\mathcal{T}$, $\mathcal{C}$ be *efficient* ensembles. Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a tweakable encryption scheme. The key generation algorithm KeyGen takes as input the security parameter 1^λ, and samples a uniform random key $K \in \mathcal{K}$. The encryption algorithm takes as input the key K, a tweak $T \in \mathcal{T}$, and a message $M \in \mathcal{M}$ and produces a ciphertext $C \in \mathcal{C}$. The decryption algorithm takes as input a ciphertext $C \in \mathcal{C}$, the key K, and the tweak T and outputs the corresponding plaintext M or reject outputting $\perp$. We need the obvious correctness condition or *completeness* condition to hold: $\forall K \leftarrow \mathsf{KeyGen}(1^\lambda)$, $M \in \mathcal{M}$, and $T \in \mathcal{T}$, we require that $\mathsf{Dec}_K(T, \mathsf{Enc}_K(T, M)) = M$ with probability 1.

The security definition of ℓ-LR-CPA secure tweakable symmetric-key encryption consists of an initial learning stage where an attacker can adaptively ask arbitrary chosen plaintext encryption queries interleaved with leakage-queries to the leakage oracle $\mathcal{O}^\ell_K(.)$. Later, after getting the challenge ciphertext, the attacker can ask for additional chosen-plaintext encryption queries but *not* leakage queries. We need this restriction, because if the adversary is allowed access to the leakage oracle $\mathcal{O}^l_K(.)$, it can encode the decryption function, the two messages M_0, M_1, and the challenge ciphertext to leak the bit b that we are trying to hide and trivially win the security game. We later show how to remove this restriction by giving appropriate definitions and showing how our (modified) construction meet this definition. We say $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a $\ell(\lambda)$-leakage resilient tweakable encryption scheme if for all probabilistic polynomial time (PPT) adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game is negligible:

1. **Initialization:** The challenger chooses a uniformly random $K \leftarrow \mathsf{KeyGen}(1^\lambda)$ and proceeds with the game as follows.
2. **Learning Stage:** The attacker $\mathcal{A}^{\mathcal{O}^\ell_K, \mathsf{Enc}_K(.,.)}$ gets oracle access to the leakage oracle $\mathcal{O}^\ell_K$ which allows him/her to learn at most ℓ bits of information about the secret key K of the Π. Besides the adversary can also query the encryption oracle $\mathsf{Enc}_K(.,.)$. The adversary queries the oracle with tuples $(T_i, M_i)_{i\in[q]}$ for some polynomial $q = q(\lambda)$, $T_i \in \mathcal{T}$ and $M_i \in \mathcal{M}$, and receives as output $C_i \leftarrow \mathsf{Enc}_K(T_i, M_i)$ where $C_i \in \mathcal{C}$.
3. **Challenge Stage:** In this stage, the adversary submits two message-tweak pairs (M_0, T_0) and (M_1, T_1), where $|M_0| = |M_1|$. The challenger chooses a random bit $b \leftarrow \{0, 1\}$, compute $C_b = \mathsf{Enc}_K(M_b, T_b)$ and sends C_b to the adversary. The adversary $\mathcal{A}$ then outputs a bit b'.

We define the advantage of the attacker $\mathcal{A}$ as $Adv^{\text{LR-t-CPA}}_{\mathcal{A}}(\lambda) = |\Pr[b' = b] - \frac{1}{2}|$. We say a leakage-resilient tweakable encryption is $(t', q'_{enc}, q'_{leak}, \epsilon')$-secure if for all t'-resource bounded adversaries making at most q'_{enc} encryption oracle queries and at most q'_{leak} leakage queries, $Adv^{\text{LR-t-CPA}}_{\mathcal{A}}(\lambda) \leq \epsilon'$.

4.2 Constructing Tweakable weak PRF from weak PRF

In this section we show how to construct Tweakable weak PRF (t-wPRF) from standard weak PRF (wPRF). We give two generic constructions. Our first construction (Construction 1) is trivial and is not very efficient. Our second construction (Construction 2) is more efficient and it can be viewed as a (simplified) variant of the XTS mode of operation [30].

Construction 1. Let $\mathcal{F} = \{f : \mathcal{K} \times \mathcal{D} \to \mathcal{R}\}$ be a (t, q, ϵ)-weak PRF family as defined in section 4 where $\mathcal{K}$, $\mathcal{D}$ and $\mathcal{R}$ denote respectively the keyspace, plaintext space and cipherspace of the wPRF. We construct a t-wPRF $\widetilde{\mathcal{E}} = \{E : \mathcal{K} \times \mathcal{T} \times \mathcal{D} \to \mathcal{R}\}$ as follows. In our construction we require $\mathcal{T} = \mathcal{D}$ for the t-wPRF $\widetilde{\mathcal{E}}$. The construction is fairly straightforward and is given as:

$$E_K(T, X) = f(f(K, T), X)$$

where $K \leftarrow \mathcal{K}$, $T \leftarrow \mathcal{T}$ and $X \leftarrow \mathcal{D}$.

Theorem 1. *Suppose $\mathcal{F} = \{f : \mathcal{K} \times \mathcal{D} \to \mathcal{R}\}$ be a (t, q, ϵ)-weak PRF family; then $\widetilde{\mathcal{E}} = \{E : \mathcal{K} \times \mathcal{T} \times \mathcal{D} \to \mathcal{R}\}$ constructed as above is a $(t, q, 2\epsilon)$- secure tweakable weak-PRF family.*

Proof. The claim follows simply by replacing the value $f(K, T)$ with a uniform random and independent value $\hat{K} \leftarrow \mathcal{K}$ from the key space of the wPRF and arguing that this change is indistinguishable by the security of the underlying wPRF. Then this modified function $f(\hat{K}, X)$ behaves like a normal w-PRF with a uniform random key. Now we simply replace this value by another uniformly random value $Y \leftarrow \mathcal{R}$, from the range space of the wPRF, and similarly argue that this change is indistinguishable from the viewpoint of the adversary by the security of the wPRF. The replacement of the inner evaluation of the wPRF with a uniform random key gives the adversary advantage at most ϵ, and the final replacement gives the adversary advantage at most another ϵ, resulting in the 2ϵ factor mentioned in the statement of the theorem. If ϵ is negligible, so is 2ϵ. We define a sequence of games– Game 1, 2 and 3 and let $\mathsf{Adv}_{\mathsf{Game}_i}(\mathcal{A}^{\mathsf{t\text{-}wPRF}})$ denote the advantage of the tweakable weak-PRF adversary $\mathcal{A}^{\mathsf{t\text{-}wPRF}}$ in Game i.

Game 0. This is the normal w-PRF game in which the challenger uses the bit $b = 0$. In other words, in the challenge phase, the challenger of the t-wPRF invokes the challenger of the w-PRF. The challenger of the w-PRF samples an element say $x \leftarrow \mathcal{D}$, and sends back $(x, (y = f(K, x))$ to the t-wPRF challenger. The t-wPRF challenger then sets $T := x$, samples $X \leftarrow \mathcal{D}$, and computes $Y = E(y, X)$. It then returns (X, T, Y) to the adversary of the t-wPRF.

Game 1. In this game, the challenger of the w-PRF uses the bit $b = 1$. In other words, the challenger returns (x, y) to the t-wPRF challenger, where $x \leftarrow \mathcal{D}$ and $y \leftarrow \mathcal{R}$. The t-wPRF challenger then sets $T := x$, samples $X \leftarrow \mathcal{D}$, and computes $Y = E(y, X)$. It then returns (X, T, Y) to the adversary of the t-wPRF. Note that the difference between Game 0 and Game 1 is the way in which the challenger of the w-PRF behaves. So, by the (t, q, ϵ) security of the weak PRF family, we have,

$$\left|\mathsf{Adv}_{\mathsf{Game}_1}(\mathcal{A}^{\mathsf{t\text{-}wPRF}}) - \mathsf{Adv}_{\mathsf{Game}_0}(\mathcal{A}^{\mathsf{t\text{-}wPRF}})\right| \leq \epsilon.$$

Game 2. In this game, both the challengers uses the bit $b = 1$. In other words, the challenger of the w-PRF returns (x, y) to the t-wPRF challenger, where $x \leftarrow \mathcal{D}$ and $y \leftarrow \mathcal{R}$. The t-wPRF challenger then sets $T := x$, samples $X \leftarrow \mathcal{D}$, and returns (X, T, Y) to the adversary, where $Y \leftarrow \mathcal{R}$. Note that the difference between Game 1 and Game 2 is the way in which the challenger of the t-wPRF behaves. Again, by the (t, q, ϵ) security of the weak PRF family, we have,

$$\left|\mathsf{Adv}_{\mathsf{Game}_2}(\mathcal{A}^{\mathsf{t\text{-}wPRF}}) - \mathsf{Adv}_{\mathsf{Game}_1}(\mathcal{A}^{\mathsf{t\text{-}wPRF}})\right| \leq \epsilon.$$

Finally, we have,

$$\mathsf{Adv}_{\mathsf{Game}_0}(\mathcal{A}^{\mathsf{t\text{-}wPRF}}) \leq \epsilon + \epsilon = 2\epsilon.$$

This completes the proof of Theorem 1. □

Construction 2. In our previous construction of t-wPRF, it can be seen that if we want to evaluate/encrypt n blocks with this t-wPRF, this would require $2n$ evaluations of the underlying w-PRF. This may be expensive in many application where we need to encrypt huge number of blocks for e.g, in most of the disk encryption systems we need to encrypt large number of sectors/blocks. So we propose a new construction that brings down the number of wPRF invocations from $2n$ to n for encrypting n blocks. Our construction is basically a *simplified* variant of the XTS mode of operation of tweakable-PRF (t-PRF) [30]. In XTS mode of operation, the key K comprises of two sub-keys (K_1, K_2), and the tweak space is represented as a binary tuple (T, i), where T represents the tweak (for disk encryption system T represents the sector number) and i represents the block number (comprising a sector). It first encrypts the tweak T using the right sub-key K_2, i.e., $\widetilde{T} := \mathsf{F}(K_2, T)$ where F is secure PRF, and computes the following: $E_{K=(K_1,K_2)}(T, X) = \mathsf{F}(K_1, X \oplus P(\widetilde{T}, i)) \oplus P(\widetilde{T}, i)$. Here P is a simple padding function. It can be proved that if F is a secure PRF, then this yields a secure tweakable PRF (t-PRF). However we note that to ensure security we need to encrypt the tweak first, else this can lead to an attack as described below. The adversary can simply query the encryption oracle using $X = P(T, i)$ for some block number i. The adversary receives as output $C_i = C_0 \oplus P(T, i)$, where C_0 is the encryption of the element "0" (since now $\widetilde{T} = P(T, i)$ and X is also $P(T, i)$). Similarly the adversary can again query the encryption oracle with $X = P(T, j)$ for some block number $j \neq i$. The adversary receives as output $C_j = C_0 \oplus P(T, j)$, and C_0 is the encryption of the element "0" as before. Now the adversary can perform $C_i \oplus C_j$ to get $P(T, i) \oplus P(T, j)$. In this way the adversary can distinguish a t-PRF from a random family of permutation by simply exor-ing the two ciphertexts C_i and C_j and checking whether the above equality holds.

However, we show that this construction mentioned above (i.e. the unencrypted tweaked version of XTS) which we refer to as *simplified* XTS already suffices to be a t-wPRF. This is because in the construction of t-wPRF, we do not allow the adversary to query on input points; instead the challenger chooses the inputs uniformly at random. The probability that the challenger samples the value of $X = P(T, i)$ for some i is negligible; so that attack mentioned works only with negligible probability. More formally our construction is as follows:

$$E_{K=(K_1,K_2)}(\hat{T} = (T, i), X) = \mathsf{F}(K_1, X \oplus P(T, i)) \oplus P(T, i)$$

Note that in this construction, we do not need to encrypt the tweak, and so to encrypt n blocks we need *exactly* n invocations of the underlying wPRF as compared to $(n+1)$ invocations in case of normal XTS mode.

Theorem 2. *Suppose $\mathcal{F} = \{f : \mathcal{K} \times \mathcal{X}_{weak} \to \mathcal{Y}_{weak}\}$ be a secure weak PRF family; then $\widetilde{\mathcal{E}} = \{E : \mathcal{K}^2 \times \mathcal{T} \times \mathcal{D} \to \mathcal{R}\}$ constructed as above is a secure tweakable weak-PRF family.*

Proof. The proof is similar to the proof of the XTS mode of operation [30] with the above modification. We do not give the proof here and leave it to the reader to fill in the details of the proof.

4.3 Constructing CPA-secure tweakable encryption from t-wPRF

Here we show how to construct tweakable encryption from tweakable weak PRF. If the underlying t-wPRF is also *leakage resilient*, then the resulting tweakable encryption scheme is also *leakage resilient.* The security definitions of leakage resilient t-wPRF (LR-twPRF) and leakage resilient tweakable encryption is given in sections 4 and 4.1 respectively. The construction is rather straightforward given a LR-t-wPRF. Assume $\widetilde{\mathcal{E}} = \{E : \mathcal{K} \times \mathcal{T} \times \mathcal{D} \to \mathcal{R}\}$ is ℓ-LR-twPRF family, where the output domain $\mathcal{Y}$ and the tweak space $\mathcal{T}$ forms an additive group (e.g., bit-strings under XOR).

1. $\mathsf{KeyGen}(1^\lambda)$: Sample a uniformly random and independent $K \leftarrow \mathcal{K}$
2. $\mathsf{Enc}(M,\ K,\ T)$: The encryption algorithm on input a key $K \in \mathcal{K}$, a message $M \in \mathcal{R}$, and a tweak $T \in \mathcal{T}$, samples $X \leftarrow \mathcal{D}$, $\widehat{T} \leftarrow \mathcal{T}$ and compute:

$$1)\ \ \widetilde{T} := T \oplus \widehat{T}, \ \text{ and } \ 2)\ \ \mathsf{Enc}_K(T, M) = (X, \widehat{T}, E_K(\widetilde{T}, X) \oplus M)$$

3. $\mathsf{Dec}(K, X, T, \widehat{T}, C)$: The decryption algorithm takes in the secret key K and the tuple $(X, T, \widehat{T}, C)$ and computes $\widetilde{T} := T \oplus \widehat{T}$ and $M = C \oplus E_K(\widetilde{T}, X)$

Remark 4. Note that in this construction, we had to additionally introduce another random tweak from the tweak space. This is required to argue security since a t-wPRF requires both uniform random inputs and tweaks. In the CPA-security game the adversary may not query the t-wPRF oracle with random tweaks, so it is not immediately clear how to argue security in that case. So we had to introduce an additional tweak and transform it into a uniformly random tweak and use the modified tweak $\widetilde{T}$ in the encryption algorithm for the security proof to go through.

Theorem 3. *If $\widetilde{\mathcal{E}}$ is ℓ-leakage resilient t-wPRF, then the above encryption scheme is also ℓ-LR-CPA secure*

Proof. We proceed via sequence of games. In Game 0, the challenge ciphertext is computed as $\mathsf{Enc}_K(T, M_b) = (X, \widehat{T}, E_K(\widetilde{T}, X) \oplus M_b)$, where $b \in \{0, 1\}$. In the next game, Game 1, we replace the value $E_K(\widetilde{T}, X)$ with a uniformly random and independent value. We argue that this change is indistinguishable by the ℓ-LR-twPRF security of E. Hence the bit b is now information theoretically hidden from the view of the adversary and hence the advantage of the adversary is 0 in this game.

Game 0. In this game the ℓ-LR-CPA adversary asks encryption and leakage queries.
Learning Stage: In the learning phase, the adersary can encryption as well as leakage queries.
Encryption queries: When the ℓ-LR-CPA adversary queries with the pair (T, M), where $T \in \mathcal{T}$, and $M \in \mathcal{M}$ (recall $\mathcal{M}$ denotes the message space of the encryption scheme and for our purpose $\mathcal{M} = \mathcal{R}$), the ℓ-LR-CPA challenger (which is also the ℓ-LR-twPRF adversary) invokes the ℓ-LR-twPRF challenger with the tweak T. The ℓ-LR-twPRF challenger samples $X \leftarrow \mathcal{D}$, $\widehat{T} \leftarrow \mathcal{T}$, and compute (1) $\widetilde{T} := T \oplus \widehat{T}$ and (2) $Y = E_K(\widetilde{T}, X)$ and returns $(X, \widehat{T}, Y)$ to the ℓ-LR-CPA challenger. The ℓ-LR-CPA challenger then computes $Z = Y \oplus M$, and returns $(X, \widehat{T}, Z)$ to the ℓ-LR-CPA adversary.

Leakage queries: When the adversary asks leakage queries to the to the ℓ-LR-CPA challenger it forwards the leakage queries to its challenger and returns back the response to the ℓ-LR-CPA adversary.
Challenge: In the challenge phase when the ℓ-LR-CPA adversary gives two message-tweak pairs (M_0, T_0) and (M_1, T_1), the ℓ-LR-CPA challenger choses $b \in \{0, 1\}$, and sends the tweak T_b to its challenger. The challenger samples $X^* \leftarrow \mathcal{D}$, $\widehat{T}^* \leftarrow \mathcal{T}$, and compute (1) $\widetilde{T}^* := T_b \oplus \widehat{T}^*$ and (2) $Y^* = E_K(\widetilde{T}^*, X^*)$ and returns $(X^*, \widehat{T}^*, Y^*)$ to the ℓ-LR-CPA challenger. The ℓ-LR-CPA challenger then computes $Z^* = Y^* \oplus M_b$, and returns $(X^*, \widehat{T}^*, Z^*)$ to the ℓ-LR-CPA adversary.

Game 1. This game is similar to Game 0, except that the challenger of the ℓ-LR-twPRF choses the bit $b = 1$ in the challenge phase, and so the value Y^* is now random. More precisely, the challenger samples $X^* \leftarrow \mathcal{D}$, $\widehat{T}^* \leftarrow \mathcal{T}$, and compute (1) $\widetilde{T}^* := T_b \oplus \widehat{T}^*$ and (2) $Y^* \leftarrow \mathcal{R}$ and returns $(X^*, \widehat{T}^*, Y^*)$ to the ℓ-LR-CPA challenger. The ℓ-LR-CPA challenger then computes $Z^* = Y^* \oplus M_b$, and returns $(X^*, \widehat{T}^*, Z^*)$ to the ℓ-LR-CPA adversary.

Note that, at this point the bit b is now information theoretically hidden from the view of the ℓ-LR-CPA adversary, and hence the advantage of the adversary in this game is 0. The encryption and leakage queries in the learning phase are handled similarly as in Game 0. So Game 0 and Game 1 are indistinguishable. This completes the proof of this theorem. □

Remark 5. We note that the encryption is also resilient to leakage jointly from the secret key and also the *randomness* used to answer the encryption queries in the initial learning phase (but not in the challenge phase). This follows from the fact that this is a *public-coin* encryption scheme, since the randomness X and $\widehat{T}$ used to generate the ciphertexts is provided as part of the ciphertext in clear.

4.4 Achieving Leakage Resilience via Symmetric Tweakable weak Hash Proof System

In order to achieve leakage resilient tweakable weak-PRF, we introduce a new primitive called *Symmetric-key Tweakable weak hash proof system* (S-twHPS). Similar ideas were used in the construction of leakage-resilient wPRF by Hazay *et. al.* [19], who build a LR-wPRF from a symmetric-key weak HPS. We basically retain the high level properties of the symmetric-key weak HPS of [19], while introducing an additional dimension namely the tweak space and describing appropriate distributions over the tweak space. Namely, we introduce three more distributions corresponding to the uniform sampling, valid and invalid samplings of tweak from the tweak space as described below and modified the input indistinguishability and smoothness properties appropriately. For our purpose, we will view this hash proof system as a tweakable weak PRF (t-wPRF) family $\widetilde{\mathcal{E}} = \{E_K : \mathcal{T} \times \mathcal{D} \to \mathcal{R}\}_{K \in \mathcal{K}}$ with some special properties. We define six distributions according to the sampling possibilities. We require the inputs $X \leftarrow \mathcal{D}$ to be sampled *uniformly at random* which we refer to as $\mathsf{Dist}_0^{\mathcal{D}}$. Besides this, we also define two more distributions over the input domain of the t-wPRF denoted as $\mathsf{Dist}_1^{\mathcal{D}}$ (*valid*) and $\mathsf{Dist}_2^{\mathcal{D}}$ (*invalid*). Similarly we also define $\mathsf{Dist}_0^{\mathcal{T}}$ (corresponding to sampling $T \leftarrow \mathcal{T}$ uniformly at random), $\mathsf{Dist}_1^{\mathcal{T}}$ (*valid*) and $\mathsf{Dist}_2^{\mathcal{T}}$ (invalid). We say a joint distribution to be *like-joint distribution* when both the samples of the joint distribution are sampled from similar distributions (in our case either both are sampled *uniformly at random* or both sampled from *valid* or both from *invalid*). We require that samples from *like-joint distribution* of the input and the tweak are indistinguishable even given the secret key K. We refer to this property as the "*joint-input indistinguishability*" property. The second property we need the S-twHPS to satisfy is the "*smoothness*" property, which tells that conditioned on seeing many pairs $\{(X_i, T_i, E_K(T_i, X_i))\}$ for many different $X_i \leftarrow \mathsf{Dist}_1^{\mathcal{D}}$ (valid), $T_i \leftarrow \mathsf{Dist}_1^{\mathcal{T}}$ (valid) and a random choice of $X^* \leftarrow \mathsf{Dist}_2^{\mathcal{D}}$ (invalid), $T^* \leftarrow \mathsf{Dist}_2^{\mathcal{T}}$ (invalid) the output of $E_K(T^*, X^*)$ will be *truly random* and *independent*, where the randomness comes from the choice of a consistent secret key K. Notice that this implies that there must be many possible secret keys K that are consistent with the values $(X_i, T_i, E_K(T_i, X_i))$. The additional distributions $\mathsf{Dist}_1^{\mathcal{D}}$, $\mathsf{Dist}_2^{\mathcal{D}}$, $\mathsf{Dist}_1^{\mathcal{T}}$ and $\mathsf{Dist}_2^{\mathcal{T}}$ are only used in the context of the security definitions and proofs, and never in the actual schemes. In the actual scheme we will sample $X \leftarrow \mathcal{D}$ and $T \leftarrow \mathcal{T}$ uniformly at random (i.e., from $\mathsf{Dist}_0^{\mathcal{D}}$ and $\mathsf{Dist}_0^{\mathcal{T}}$ respectively). In the definition of S-twHPS we will also define a sampling key sampK which is needed in order to efficiently sample from the distributions $\mathsf{Dist}_1^{\mathcal{D}}$, $\mathsf{Dist}_2^{\mathcal{D}}$, $\mathsf{Dist}_1^{\mathcal{T}}$ and $\mathsf{Dist}_2^{\mathcal{T}}$. Formally, a S-twHPS is defined as follows:

Definition 5. *(Symmetric key tweakable weak HPS) Let $\mathcal{D}$, $\mathcal{K}$, $\mathcal{T}$ and $\mathcal{R}$, be efficient ensembles, and let $\widetilde{\mathcal{E}} = \{E_K : \mathcal{T} \times \mathcal{D} \rightarrow \mathcal{R}\}_{K \in \mathcal{K}}$ be some efficient function family with the following PPT algorithms:*

1. $\mathsf{sampK} \leftarrow \mathsf{SampGen}(K)$ takes an input $K \in \mathcal{K}$ and outputs a sampling key sampK.
2. $X \leftarrow \mathsf{Dist}_0^{\mathcal{D}}(\mathsf{sampK})$, $X \leftarrow \mathsf{Dist}_1^{\mathcal{D}}(\mathsf{sampK})$, $X \leftarrow \mathsf{Dist}_2^{\mathcal{D}}(\mathsf{sampK})$: Samples $X \in \mathcal{D}$ according to the distribution using the sampling key sampK. For sampling $X \leftarrow \mathsf{Dist}_0^{\mathcal{D}}$, it simply samples a uniformly random and independent $X \leftarrow \mathcal{D}$ and ignores the sampling key. For distributions $\mathsf{Dist}_1^{\mathcal{D}}$ and $\mathsf{Dist}_2^{\mathcal{D}}$ we need the sampling key sampK.
3. $T \leftarrow \mathsf{Dist}_0^{\mathcal{T}}(\mathsf{sampK})$, $T \leftarrow \mathsf{Dist}_1^{\mathcal{T}}(\mathsf{sampK})$, $T \leftarrow \mathsf{Dist}_2^{\mathcal{T}}(\mathsf{sampK})$: Samples $T \in \mathcal{T}$ according to the distribution using the sampling key sampK. For sampling $T \leftarrow \mathsf{Dist}_0^{\mathcal{T}}$, it simply samples a uniformly random and independent $T \leftarrow \mathcal{T}$ and ignores the sampling key. For distributions $\mathsf{Dist}_1^{\mathcal{T}}$ and $\mathsf{Dist}_2^{\mathcal{T}}$ we need the sampling key sampK.

We say the $\widetilde{\mathcal{E}}$ is symmetric-key tweakable wHPS (S-twHPS) if the following properties are satisfied:

- **Joint-Input Indistinguishability.** *For any polynomial $q = q(\lambda)$ and any choice of $(b_1, \ldots, b_q)$, $(b'_1, \ldots, b'_q) \in \{0, 1, 2\}^q$ the following distributions are computationally indistinguishable:*

$$(K, (T_1, X_1), (T_2, X_2), \ldots, (T_q, X_q) \approx_c (K, (T'_1, X'_1), (T'_2, X'_2) \ldots, T'_q, X'_q)$$

  where $K \leftarrow \mathcal{K}$, $\mathsf{sampK} \leftarrow \mathsf{SampGen}(K)$, $\{X_i \leftarrow \mathsf{Dist}_{b_i}^{\mathcal{D}}(\mathsf{sampK}), T_i \leftarrow \mathsf{Dist}_{b_i}^{\mathcal{T}}(\mathsf{sampK})\}$, and $\{X'_i \leftarrow \mathsf{Dist}_{b'_i}^{\mathcal{D}}(\mathsf{sampK}), T_i \leftarrow \mathsf{Dist}_{b'_i}^{\mathcal{T}}(\mathsf{sampK})\}$.

- **Smoothness**. *For any polynomial $q = q(\lambda)$, the following distributions are statistically indistinguishable:*

$$(((T_1, X_1), Y_1) \ldots, ((T_q, X_q), Y_q), ((X^*, T^*), Y^*))$$
$$\equiv (((T_1, X_1), Y_1), \ldots, ((T_q, X_q), Y_q), ((X^*, T^*), U_{\mathcal{Y}}))$$

  where $K \leftarrow \mathcal{K}$, $\mathsf{sampK} \leftarrow \mathsf{SampGen}(K)$, $\{X_i \leftarrow \mathsf{Dist}_1^{\mathcal{D}}(\mathsf{sampK})$, $T_i \leftarrow \mathsf{Dist}_1^{\mathcal{T}}(\mathsf{sampK}), Y_i = E_K(T_i, X_i)\}_{i \in [q]}$, $\{X^* \leftarrow \mathsf{Dist}_2^{\mathcal{D}}(\mathsf{sampK})$, $T^* \leftarrow \mathsf{Dist}_2^{\mathcal{T}}(\mathsf{sampK}), Y^* = E_K(T^*, X^*)\}$ and $U_{\mathcal{Y}} \leftarrow \mathcal{Y}$. In other words, Y^* is uniformly random and independent of the other elements, where the randomness comes from the choice of a consistent key K.

4.5 Constructing LR-twPRF from Symmetric-Key t-wHPS

In this section, we construct a leakage-resilient tweakable weak PRF from a symmetric-key tweakable weak HPS. Our construction and proof strategy follows closely along the lines of Naor and Segev [25] and Hazay *et. al.* [19].

Let $\mathcal{D}$, $\mathcal{R}$, $\mathcal{T}$, and $\mathcal{Z}$ be efficient ensembles such that $\widetilde{\mathcal{E}} = \{E_K : \mathcal{T} \times \mathcal{D} \to \mathcal{R}\}_{K \in \mathcal{K}}$ is a symmetric-key tweakable weak HPS (S-twHPS), and let $\mathsf{Ext}\colon \mathcal{R} \times \mathcal{S} \to \mathcal{Z}$ be a $((\log(|\mathcal{R}|) - \ell(\lambda)), \varepsilon(\lambda))$- extractor (see Def. 3), for some negligible function $\varepsilon(\lambda)$. Let us define the function family $\widetilde{\mathcal{E}}' = \{E'_K : \mathcal{T} \times \mathcal{D} \times \mathcal{S} \to \mathcal{Z}\}_{K \in \mathcal{K}}$ as follows: Compute $y = E_K(T, X)$ and define $E'_K(X, T, S) := \mathsf{Ext}(y; S)$, where $X \leftarrow \mathcal{D}, T \leftarrow \mathcal{T}$ and $S \leftarrow \mathcal{S}$.

Theorem 4. *If $\widetilde{\mathcal{E}} = \{E_K : \mathcal{T} \times \mathcal{D} \to \mathcal{R}\}_{K \in \mathcal{K}}$ is a symmetric-key tweakable weak HPS (*S-twPRF*), and $\mathsf{Ext}\colon \mathcal{R} \times \mathcal{S} \to \mathcal{Z}$ is a $((log(|\mathcal{R}|) - \ell(\lambda)), \varepsilon(\lambda))$- extractor, then $\widetilde{\mathcal{E}}'$ is a $\ell(\lambda)$-*LR-twPRF*.

Proof. We prove the theorem via a hybrid argument over several games defined below.

Game 0. This game corresponds to the t-wPRF security game (Definition 4.1) where the challenger uses the bit $b = 0$, meaning that the challenge tuple is *pseudo-random.* More precisely, all the output of the learning phase are answered as follows: Choose $(X_i, T_i, S_i) \leftarrow \mathcal{D} \times \mathcal{T} \times \mathcal{S}$ at random and compute $Z_i = E'_K(X_i, T_i, S_i) := \mathsf{Ext}(E_K(T_i, X_i); S_i)$. Finally, the tuple (X_i, T_i, S_i, Z_i) is returned as response to the adversary. Similarly, the challenge tuple is also constructed in a similar fashion by choosing $(X^*, T^*, S^*) \leftarrow \mathcal{D} \times \mathcal{T} \times \mathcal{S}$ at random and computing $Z^* = E'_K(X^*, T^*, S^*) := \mathsf{Ext}(E_K(T^*, X^*); S^*)$. The tuple (X^*, T^*, S^*, Z^*) is returned as the challenge tuple.

Game 1. In this game, we rely on the symmetric-key t-wHPS property of $\widetilde{\mathcal{E}}$ to change the distribution of all the $\{X_i, T_i\}_{i \in [q]}$ values during the learning stage to come from $\mathsf{Dist}_1^{\mathcal{D}}$ and $\mathsf{Dist}_1^{\mathcal{T}}$ respectively, and the value used in generating the challenge namely X^* and T^* to come from $\mathsf{Dist}_2^{\mathcal{D}}$ and $\mathsf{Dist}_2^{\mathcal{T}}$ respectively. More precisely, in the learning the challenger answers to all the queries by choosing $\{X_i \leftarrow \mathsf{Dist}_1^{\mathcal{D}}(\mathsf{sampK})\}_{i \in [q]}, \{T_i \leftarrow \mathsf{Dist}_1^{\mathcal{T}}(\mathsf{sampK})\}_{i \in [q]}$ $\{S_i \leftarrow \mathcal{S}\}_{i \in [q]}$, and computing $\{Z_i = E'_K(Y_i, S_i)\}_{i \in [q]}$ where $\{Y_i = E_K(T_i, X_i)\}_{i \in [q]}$. In the challenge phase, the challenger chooses $X^* \leftarrow \mathsf{Dist}_2^{\mathcal{D}}(\mathsf{sampK})$, $T^* \leftarrow \mathsf{Dist}_2^{\mathcal{T}}(\mathsf{sampK})$, $S^* \leftarrow \mathcal{S}$, and computes $Z^* = E'_K(Y^*, S^*)$, where again $Y^* = E_K(T^*, X^*)$.

We argue that Game 0 and 1 are computationally indistinguishable by the *joint-input indistinguishability* property of the underlying symmetric-key tweakable wHPS (S-twHPS). Let q be the total number of t-wPRF queries that the adversary $\mathcal{A}$ makes during the learning stage. The reduction algorithm takes a tuple of the form $(K, (T_1, X_1), (T_2, X_2), \ldots, (T_q, X_q), (T^*, X^*))$ as input. This is used to simulate all the leakage queries and the t-wPRF queries made by $\mathcal{A}$ and also to form the challenge ciphertext. If $\{X_i, T_i\}_{i \in [q]}$, and the X^* are chosen uniformly at random (i.e., from $\mathsf{Dist}_0^{\mathcal{D}}$ and $\mathsf{Dist}_0^{\mathcal{T}}$) then this perfectly simulates Game 0 and if they are chosen via $\{X_i \leftarrow \mathsf{Dist}_1^{\mathcal{D}}(\mathsf{sampK})\}_{i \in [q]}, \{T_i \leftarrow \mathsf{Dist}_1^{\mathcal{T}}(\mathsf{sampK})\}_{i \in [q]}$ and $X^* \leftarrow \mathsf{Dist}_2^{\mathcal{D}}(\mathsf{sampK})$, $T^* \leftarrow \mathsf{Dist}_2^{\mathcal{T}}(\mathsf{sampK})$ then this perfectly simulates Game 1. Note that the input indistinguishable property holds even if the entire secret key is available to the adversary, and hence certainly holds good when given bounded leakage on the secret key.

Game 2. In this game, the challenger further modifies the challenge tuple generation. In particular, the challenge is generated by sampling $X^* \leftarrow \mathsf{Dist}_2^{\mathcal{D}}(\mathsf{sampK})$, $T^* \leftarrow \mathsf{Dist}_2^{\mathcal{T}}(\mathsf{sampK})$ as before, but the value of Z^* is sampled uniformly at random and independently, i.e., $Z^* \leftarrow \mathcal{Z}$. We argue that this change is indistinguishable from the view point of an adversary relying the smoothness property and property of the strong average case extractor. Note that in the learning phase, the values of X_i and T_i are sampled from valid distributions, namely, $\mathsf{Dist}_1^{\mathcal{D}}$ and $\mathsf{Dist}_1^{\mathcal{T}}$ respectively. The challenge is generated by choosing X^* and T^* from invalid distributions, namely, $\mathsf{Dist}_2^{\mathcal{D}}$ and $\mathsf{Dist}_2^{\mathcal{T}}$ respectively. By the *smoothness* property of the underlying S-twHPS, the value $Y^* = E_K(T^*, X^*)$ is uniformly random even given X^*, T^* and all of the wPRF query responses that the attacker sees in the learning stage. Let us denote the information learned by the adversary in the learning phase by $aux = (\{X_i, T_i, S_i, Z_i = E'_K(X_i, T_i, S_i)\}_{i \in [q]})$. So, by the smoothness property, $\widetilde{\mathrm{H}}_\infty(Y^*|X^*, T^*, aux) = \log(|\mathcal{R}|)$. Let $L = \{0,1\}^{l(\lambda)}$ denote the response to the leakage query made by the adversary $\mathcal{A}$ during the learning phase to the leakage oracle $\mathcal{O}_K^{\ell}$. [5] Therefore, by the chain-rule of average min-entropy we have:

$$\widetilde{\mathrm{H}}_\infty(Y^*|X^*, T^*, aux, L) \geq \widetilde{\mathrm{H}}_\infty(Y^*|X^*, T^*, aux) - \ell(\lambda) = \log(|\mathcal{R}| - \ell(\lambda))$$

Finally, we apply the $(\log(|\mathcal{R}|) - \ell(\lambda), \varepsilon(\lambda))$- extractor to then output of S-twHPS, which transforms it into a uniformly random value. So by the security property of the extractor we get:

$$(X^*, T^*, S^*, aux, L, Z^* = E'_K(X^*, T^*, S^*)) \approx_s (X^*, T^*, S^*, aux, L, Z^* \leftarrow \mathcal{Z})$$

We note that the seed of the extractor is choosen uniformly at random and independent of the auxiliary values aux learnt by $\mathcal{A}$ in the learning phase and also the leakage L. Therefore, even conditioned on everything the attacker sees in the learning stage, and on the challenge input (X^*, T^*, S^*) the value $Z^* = E'_K((X^*, T^*, S^*)) = \mathsf{Ext}(Y^*; S^*)$ is statistically indistinguishable from uniform. So the change from Game 1 to Game 2 is indistinguishable to an adversary except with negligible probability.

Game 3. In this game the challenger chooses the bit $b = 1$ (recall that the bit $b = 1$ corresponds to receiving the output from a truly random function) in the challenge phase. In addition, the challenger switches back the distributions to same as Game 0. In particular, now in the learning phase, the challenger samples $\{X_i \leftarrow \mathsf{Dist}_0^{\mathcal{D}}(\mathsf{sampK})\}_{i\in[q]}, \{T_i \leftarrow \mathsf{Dist}_0^{\mathcal{T}}(\mathsf{sampK})\}_{i\in[q]}\ X^* \leftarrow \mathsf{Dist}_0^{\mathcal{D}}(\mathsf{sampK}), T^* \leftarrow \mathsf{Dist}_0^{\mathcal{T}}(\mathsf{sampK})$. This corresponds to randomly sampling these values according to respective distributions. The responses to the learning phase is computed as usual, namely $\{Z_i = E'_K(X_i, T_i, S_i)\}_{i\in[q]}$, and the challenge is still chosen uniformly at random, i.e., $Z^* \leftarrow \mathcal{Z}$ as in Game 2. We argue that Game 2 and Game 3 are indistinguishable by the *input indistinguishability* property of the underlying S-twHPS similar to the way we argued indistinguishability between Games 0 and 1. This completes the proof of the theorem. $\square$

4.6 Constructing Symmetric-key Tweakable weak HPS

We have seen how to construct a leakage resilient tweakable weak PRF (LR-twPRF) from Symmetric-key twHPS. In this section, we show how to actually construct the Symmetric-Key Tweakable weak HPS (S-twHPS), with the desired properties we needed for our construction of LR-twPRF. For constructing the S-twHPS, we need a tweakable weak PRF (t-wPRF) and a standard CPA-secure symmetric-key encryption scheme.

Construction: Let $n = n(\lambda)$ be some polynomial, and let $\mathcal{E} = \{E_K : \mathcal{T} \times \mathcal{D} \to \mathbb{Z}_n\}_{K\in\mathcal{K}}$ be a standard t-wPRF family. Further, let $\mathcal{F}_{weak} = \{F_K : \mathcal{D} \to \mathbb{Z}_n\}_{K\in\mathcal{K}}$ be a weak-PRF family and $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a CPA-secure standard symmetric-key encryption scheme, where the encryption and decryption functions are specified as follows:

[5] Recall that adaptive access to the leakage oracle by $\mathcal{A}$ is equivalent to a *single* call to the leakage oracle $\mathcal{O}_K^{\ell}$ at the end of learning phase.

1. $\mathsf{Enc}_K(M)$: Choose a random $x \leftarrow \mathcal{D}$, and compute $C = (x, F_k(x) + M)$, where the addition is performed in $\mathbb{Z}_n$.
2. $\mathsf{Dec}_K(C = (x, y))$: Output $M = y - F_K(x)$

The security of this encryption is straightforward to prove. In the security proof, we replace the value $y = F_K(x)$ with a uniformly random and independent value and argue that the change is indistinguishable to an adversary. Note that for this encryption scheme the message space $\mathcal{M} = \mathbb{Z}_n$ and the ciphertext space $\mathcal{C} = \mathcal{D} \times \mathbb{Z}_n$. Another useful property of this encryption scheme is that we can *obliviously* sample $c \leftarrow C$ without knowing the secret key K, and this induces the same distribution as encrypting a random $m \leftarrow \mathbb{Z}_n$. Now, given the t-wPRF family $\mathcal{E}$, and a CPA-secure encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, we construct the S-twHPS as follows:

$$\widetilde{\mathcal{E}}_{\mathrm{S-twHPS}} = \{E_K : \mathcal{C}^n \times \mathcal{T}^n \to \mathbb{Z}_n\}_{K \in ([n], \mathcal{K})} \text{ where}$$
$$E_{K=(s,k)}(X = (C_1, T_1), \ldots, (C_n, T_n)) = \mathsf{Dec}_k(T_s, C_s)$$

Notice that we can efficiently sample uniformly random inputs from the domain $\mathcal{C}^n$ and $\mathcal{T}^n$ of E_K (without knowing K), which corresponds to sampling from the distributions $\mathsf{Dist}_0^{\mathcal{D}}$ and $\mathsf{Dist}_0^{\mathcal{T}}$ respectively. In this construction the tweak space $\mathcal{T}$ is also $\mathbb{Z}_n$. We define the additional algorithms needed for the definition of S -twHPS as follows:

- $\mathsf{sampK} \leftarrow \mathsf{SampGen}(K)$: Parse $K = (s, k)$. Choose $(n-1)$ other values $\{k_i \leftarrow \mathcal{K}\}_{i \in [n] \setminus t}$, set $k_s = k$. Set $\mathsf{SampK} := (k_1, \ldots, k_n)$.
- $T \leftarrow \mathsf{Dist}_1^{\mathcal{T}}(\mathsf{sampK})(Valid)$: Choose $t \leftarrow \mathbb{Z}_n$, compute $\{\hat{t}_i = \mathsf{Enc}_{k_i}(t)\}_{i \in [n]}$. Output $T = (\hat{t}_1, \ldots, \hat{t}_n)$.
- $T \leftarrow \mathsf{Dist}_2^{\mathcal{T}}(\mathsf{sampK})(Invalid)$: Choose $t \leftarrow \mathbb{Z}_n$, compute $\{\hat{t}_i = \mathsf{Enc}_{k_i}(t+i)\}_{i \in [n]}$. Output $T = (\hat{t}_1, \ldots, \hat{t}_n)$.
- $X \leftarrow \mathsf{Dist}_1^{\mathcal{D}}(\mathsf{sampK})(Valid)$: Choose $r \leftarrow \mathbb{Z}_n$, compute $\{C_i = \mathsf{Enc}_{k_i}(\hat{t}_i, r)\}_{i \in [n]}$, where the $\hat{t}_i$ values used as (encrypted) tweak are sampled accordingly to $\mathsf{Dist}_1^{\mathcal{T}}$ (valid). Output $X = ((C_1, \hat{t}_1), \ldots, (C_n, \hat{t}_n))$.
- $X \leftarrow \mathsf{Dist}_2^{\mathcal{D}}(\mathsf{sampK})(Invalid)$: Choose $r \leftarrow \mathbb{Z}_n$, compute $\{C_i = \mathsf{Enc}_{k_i}(\hat{t}_i, r + i)\}_{i \in [n]}$, where the $\hat{t}_i$ values used as (encrypted) tweak are sampled accordingly to $\mathsf{Dist}_2^{\mathcal{T}}$ (invalid). Output $X = ((C_1, \hat{t}_1), \ldots, (C_n, \hat{t}_n))$.

Note that for a *valid* T, all the encrypted tweaks $\hat{t}_i$ decrypt to the *same* value t, and for an *invalid* T, the values $\hat{t}_i$ decrypt to *different* values $(t+i)$. For a *valid* X, all the values decrypt to the *same* value r under the valid tweak $\hat{t}_i$ (all of which hides the same tweak t), whereas for an *invalid* X, all the values decrypt to *different* values $(r+i)$ under invalid tweaks $\hat{t}_i$ (all of which hide different tweaks $(t+i)$). Firstly, it is easy to see that the distributions $\mathsf{Dist}_1^{\mathcal{T}}$ and $\mathsf{Dist}_2^{\mathcal{T}}$ are indistinguishable even given the secret key (s, k). This follows from the fact that the value $\hat{t}_i$ is uniform on its own, and we cannot distinguish $\hat{t}_i$ for $i \notin s$ from uniform by the security of the wPRF. Now we have to argue that $((t, k), (\hat{t}_1, C_1), (\hat{t}_2, C_2), \ldots, (\hat{t}_n, C_n)) \approx_c ((t, k), (\hat{t}'_1, C'_1), (\hat{t}'_2, C'_2) \ldots, (\hat{t}'_n, C'_n))$, i.e., these two distributions are indistinguishable, even given the secret key (s, k), where $\{C_i \leftarrow \mathsf{Dist}_{b_i}^{\mathcal{D}}(\mathsf{sampK}), \hat{t}_i \leftarrow \mathsf{Dist}_{b_i}^{\mathcal{T}}(\mathsf{sampK})\}$, and $\{C'_i \leftarrow \mathsf{Dist}_{b'_i}^{\mathcal{D}}(\mathsf{sampK}), \hat{t}'_i \leftarrow \mathsf{Dist}_{b'_i}^{\mathcal{T}}(\mathsf{sampK})\}$ and $(b_i, b'_i) \in \{0, 1, 2\}^n$. This follows from the fact that the value $(C_s, \hat{t}_s)$ is uniform on its own, and for all indices $i \notin s$, the values $(C_i, \hat{t}_i)$ are indistinguishable from uniform by the security of the underlying t-wPRF and wPRF respectively. In particular, the indistinguishability of the values $\hat{t}_i$ for $i \notin s$ relies on the CPA-security of the symmetric-key encryption and the indistinguishability of the values C_i for $i \notin s$ relies on the CPA-security of the tweakable symmetric-key encryption. This proves the first property, namely, the *joint-input indistinguishability* property of the S-twHPS.

Now, we need to argue that the second property, namely, the *smoothness* property of the S-twHPS. Recall that, the smoothness requirement tells that given arbitrarily many $\mathsf{Valid} = \{X_i \leftarrow \mathsf{Dist}_1^{\mathcal{D}}(\mathsf{sampK}), T_i \leftarrow \mathsf{Dist}_1^{\mathcal{T}}(\mathsf{sampK}), Y_i = E_K(T_i, X_i)\}$ values, and a challenge tuple $\mathsf{Invalid} = \{X^* \leftarrow \mathsf{Dist}_2^{\mathcal{D}}(\mathsf{sampK}), T^* \leftarrow \mathsf{Dist}_1^{\mathcal{T}}(\mathsf{sampK})\}$, the value $Y^* = E_K(T^*, X^*)$ is (perfectly) uniformly random and independent of Valid. First we observe that given many values $\{X_i, T_i, E_K(T_i, C_i)\}$ where X_i and T_i are valid, we learn nothing (information theoretically) about the secret index s contained in $K = (s, k)$. For a random invalid $T^* \leftarrow \mathsf{Dist}_2^{\mathcal{T}}(\mathsf{sampK})$, $E_K(T^*) = \mathsf{Dec}_{k_s}(\hat{T}_s) = (t+s)$ is truly random and independent. Now a random invalid $X^* \leftarrow \mathsf{Dist}_2^{\mathcal{D}}(\mathsf{sampK})$, $T^* \leftarrow \mathsf{Dist}_2^{\mathcal{T}}(\mathsf{sampK})$, we have $E_K(T^*, X^*) = \mathsf{Dec}_{k_s}(\hat{T}_s, C_s) = r + s$ is also truly random and independent of Valid. This proves the smoothness property. □

4.7 Output Amplification via Parallel Repetition

In the previous construction, we showed how to construct symmetric-key t-wHPS for small output domain namely *polynomial* size output domain, which implies that the output will have at most $O(\log n)$ entropy associated with it. This cannot be used to extract even a single bit. So here, we amplify the output domain of the S-twHPS via parallel repetition using independent copies of the scheme concatenated together.

Theorem 5. *If $\widetilde{\mathcal{E}} = \{E_k : \mathcal{T} \times \mathcal{D} \rightarrow \mathcal{R}\}_{k \in \mathcal{K}}$ is a symmetric-key tweakable wHPS and let $m = m(\lambda)$ be some arbitrary polynomial. Define $\widetilde{\mathcal{E}}^m = \{E_K : \mathcal{T}^m \times \mathcal{D}^m \rightarrow \mathcal{R}^m\}_{K \in \mathcal{K}^m}$ via*

$$E_{(k_1,\ldots,k_m)}((T_1, X_1), \ldots, \ (T_m, X_m)) \stackrel{\text{def}}{=} (E_{k_1}(T_1, X_1), \ldots, E_{k_m}(T_1, X_m)).$$

Then $\widetilde{\mathcal{E}}^m$ is also a symmetric-key tweakable wHPS, whose output is amplified by a factor of m.

Proof. Before proceeding with the proof, let us redefine all the algorithms of the modified S-twHPS.

- $\mathsf{sampK} \leftarrow \mathsf{SampGen}^m(K)$: Parse $K = (k_1, \ldots, k_m)$, and sample $\{\mathsf{SampK}_i \leftarrow \mathsf{SampGen}(k_i)\}_{i \in [m]}$. Set $\mathsf{SampK} := (\mathsf{SampK}_1, \ldots, \mathsf{SampK}_m)$.
- $T \leftarrow \mathsf{Dist}_{b \in \{1,2\}}^{\mathcal{T},m}(\mathsf{sampK})$: Sample $\{T_i \leftarrow \mathsf{Dist}_b^{\mathcal{T}}(\mathsf{sampK}_i)\}_{i \in [m]}$. Output $T = (T_1, \ldots, T_m)$
- $X \leftarrow \mathsf{Dist}_{b \in \{1,2\}}^{\mathcal{D},m}(\mathsf{sampK})$: Sample $\{X_i \leftarrow \mathsf{Dist}_b^{\mathcal{D}}(\mathsf{sampK}_i)\}_{i \in [m]}$. Output $X = (X_1, \ldots, X_m)$

Now we want to show that the joint-input indistinguishability and smoothness properties holds for $\widetilde{\mathcal{E}}^m$ with the above algorithms.

1. *Joint-Input Indistinguishability.* Before proving this, let us introduce some notations for clarity. $\forall i = 1$ to q, let $T_i = \{T_{i,1}, \ldots, T_{i,m}\}$ and $X_i = \{X_{i,1}, \ldots, X_{i,m}\}$. Let us define the relation $S_i \subseteq T_i \times X_i$ such that:
   $S_i = \{(T_{i,j}, X_{i,j}) | T_{i,j} \in T_i, \ X_{i,j} \in X_i, \ j = 1, 2, \ldots, m\}$. For the modified S-twHPS $\widetilde{\mathcal{E}}^m$, for any polynomial $q = q(\lambda)$, input indistinguishability requires the following computational indistinguishability property to hold:

$$(K = (k_1, \ldots, k_m), (S_1, S_2, \ldots, S_q)) \approx_c (K = (k'_1, \ldots, k'_m), (S'_1, S'_2, \ldots, S'_q))$$

where $K \leftarrow \mathcal{K}^m$, $\mathsf{sampK} \leftarrow \mathsf{SampGen}^m(K)$, $\{X_i \leftarrow \mathsf{Dist}^{\mathcal{D}}_{b_i}(\mathsf{sampK})\}_{i\in[q]}$, $\{T_i \leftarrow \mathsf{Dist}^{\mathcal{T}}_{b_i}(\mathsf{sampK})\}_{i\in[q]}$, $\{X'_i \leftarrow \mathsf{Dist}^{\mathcal{D}}_{b'_i}(\mathsf{sampK})\}_{i\in[q]}$, $\{T'_i \leftarrow \mathsf{Dist}^{\mathcal{T}}_{b'_i}(\mathsf{sampK})\}_{i\in[q]}$, and $(b_1,\ldots,b_q),(b'_1,\ldots,b'_q) \in \{0,1,2\}^q$. This follows via a sequence of m *hybrid* steps, where at each step, instead of sampling $\{T_{i,j}, X_{i,j} \leftarrow \mathsf{Dist}^{\mathcal{D}}_{b_1} \times \mathsf{Dist}^{\mathcal{T}}_{b_1}\}_{(j\in[m],i\in[q])}$, we sample $\{T'_{i,j}, X'_{i,j} \leftarrow \mathsf{Dist}^{\mathcal{D}}_{b'_1} \times \mathsf{Dist}^{\mathcal{T}}_{b'_1}\}_{(j\in[m],i\in[q])}$ and rely on the joint-input indistinguishability property of the "small" scheme in position $j \in [m]$ to argue joint-input indistinguishability.

2. *Smoothness.* For the modified S-twHPS $\widetilde{\mathcal{E}}^m$, smoothness requires, for any polynomial $q = q(\lambda)$, we need the following statistical equivalence property to hold:

$$(((T_1,X_1),Y_1)\ldots,((T_q,X_q),Y_q),\overrightarrow{X}^*|_m,\overrightarrow{T}^*|_m,\overrightarrow{Y}^*|_m)$$

$$\equiv (((T_1,X_1),Y_1)\ldots,((T_q,X_q),Y_q),\overrightarrow{X}^*|_m,\overrightarrow{T}^*|_m,\overrightarrow{U_{\mathcal{R}}}^*|_m)$$

where $K \leftarrow \mathcal{K}^m$, $\mathsf{sampK} \leftarrow \mathsf{SampGen}^m(K)$, $\{X_i \leftarrow \mathsf{Dist}^{\mathcal{D}}_1(\mathsf{sampK})\}_{i\in[m]}$, $\{T_i \leftarrow \mathsf{Dist}^{\mathcal{T}}_1(\mathsf{sampK})\}_{i\in[m]}$, $\{X^* \leftarrow \mathsf{Dist}^{\mathcal{D}}_2(\mathsf{sampK})\}$, $\{T^* \leftarrow \mathsf{Dist}^{\mathcal{T}}_2(\mathsf{sampK})\}$, $\{Y_i = E_K(T_i,X_i)\}_{i\in[q]}$, $Y^* = F_K(X^*)$ and $\overrightarrow{U} \leftarrow \mathcal{R}^m$. This also follows by a sequence of m *hybrid* steps, where at each step, for each $j \in [m]$, instead of computing $Y^*_j = E_{k_j}(T^*_j, X^*_j)$,we switch to $U_j \leftarrow \mathcal{R}$, and rely on the smoothness property of the "small" scheme to argue smoothness of the modified scheme.

5 The Master Theorem.

So we have seen how to construct a symmetric-key tweakable wHPS starting from tweakable wPRF (t-wPRF) and a (standard) CPA-secure symmetric-key encryption scheme as shown in section 4.6. We also showed how to construct LR-tweakable weak PRF from S-twHPS in section 4.5. Putting this altogether as shown in the diagram at the end of section 2, we get the following:

Theorem 6. *Assuming the existence of one-way functions, there exists $\ell(\lambda)$-*LR-twPRF *and $\ell(\lambda)$-leakage-resilient CPA-secure tweakable encryption scheme. Further, assuming the existence the existence of tweakable weak PRFs with key size $\gamma(\lambda)$, the above schemes can achieve any leakage rate $\alpha(\lambda) = O(\frac{\log\lambda}{\gamma(\lambda)})$*

Proof. Let $n = n(\lambda)$ be a polynomial parameter in λ, which is a power of 2. (We can interpret $n = |\{0,1\}^{O(\log\lambda)}|$). Using our construction of Symmetric-key tweakable wHPS (S-twHPS) with parameter $n = n(\lambda)$ (representing the small output domain size), and using parallel repetition with parameter $m = m(\lambda)$ (representing the repetition factor), the key size of the our modified S-twHPS is $m(\gamma(\lambda) + \log n)$ (to specify the index s requires extra $\log n$ bits). Note that $m(\gamma(\lambda) + \log n) < 2m\gamma$, since for a secure encryption scheme $\gamma > \log n$. Now using the construction of leakage-resilient tweakable wPRF as shown in Section 4.5 and using universal hash functions (pairwise independent hash functions) as the extractor with output size λ-bit, we obtain a $\ell(\lambda)$-LR-twPRF with leakage bound $\ell(\lambda) = m\log n - 2\lambda$. So we get a relative leakage rate $\alpha \approx O(\log n/\gamma) \approx O(\log(\lambda)/\gamma(\lambda))$. So by appropriately choosing the parameters n and m, we obtain the claim made in the theorem. More precisely, the parameter n influences the leakage rate, whereas the parameter m influences the overall leakage bound or amount, and it can be made as large as possible by increasing the rounds of parallel repetition.

6 Extensions

In this section we show how our construction of symmetric-key tweakable encryption can be generalized considering more general settings than the *length-bounded* leakage.

6.1 Entropy-bounded Leakage-Resilient Tweakable Encryption.

The notion of entropy-bounded leakage was first suggested by Naor and Segev [25]. This model does not restrict the output length of the leakage function to some ℓ bits; rather the length of the leakage function can even exceed the length of the secret key, but the secret key should have a reasonable amount of *min-entropy* left in it even given this leakage. So the secret key should still be unpredictable given the leakage. There are various definitions used to quantify this requirement (see [5, 9, 25]). We consider the definition give in [9]. This was also used as a working definition in [19] to show generalizations of their construction in the entropy-bounded leakage model. Dodis *et. al.* [9] considered entropy loss with respect to an uniform distribution and showed that the entropy loss over an uniform distribution is an upper bound on the entropy loss over any arbitrary distribution.
More precisely, they defined a function $f : \{0,1\}^* \to \{0,1\}^*$ to be ℓ-leaky, if $\forall n \in \mathbb{N}$, we have $\widetilde{H}_\infty(U_n|f(U_n)) \geq n - \ell$, where, U_n is the uniform distribution over $\{0,1\}^n$. Now, a length-bounded leakage function $f : \{0,1\}^* \to \{0,1\}^\ell$ is also ℓ-leaky. Using this observation, Dodis *et. al* showed that if a function is ℓ-leaky, then it decreases the entropy of every distribution by at most ℓ bits. We modify the security model by allowing the adversary to choose at each step a ℓ_i-leaky function as leakage function and the total leakiness is bounded by some leakiness parameter $\ell \leq \sum_i \ell_i$. We note that this leakiness condition may not be efficiently verifiable for the challenger in the security game, since the checking the amount of leakiness for a function in general may not be possible. So instead we require the adversary to satisfy this condition and define the security game with respect to the class of adversary that respects the above conditions. More precisely, the security game of $\ell(= \ell(\lambda))$-leaky tweakable encryption is defined as follows. Let us denote the class $\mathcal{F}$ as the class of ℓ-leaky functions.

1. **Initialization:** The challenger chooses a uniformly random $K \leftarrow \mathsf{KeyGen}(1^\lambda)$ and proceeds with the game as follows.
2. **Learning Stage:** The attacker $\mathcal{A}^{\mathcal{O}_K^\ell, \mathsf{Enc}_K(.,.)}$ gets oracle access to the leakage oracle $\mathcal{O}_K^\ell$. The adversary can adaptively submit functions $f_i \in \mathcal{F}$ to the leakage oracle $\mathcal{O}_K^\ell$, each of which is ℓ_i-leaky, where $\ell_i \leq \ell$, and it should hold that $\sum_i \ell_i \leq \ell$. This allows the adversary to reduce the min-entropy of the secret key K by at most ℓ bits. Besides the adversary can also query the encryption oracle $\mathsf{Enc}_K(.,.)$ as usual. The adversary queries the oracle with tuples $(T_i, M_i)_{i\in[q]}$ for some polynomial $q = q(\lambda)$, $T_i \in \mathcal{T}$ and $M_i \in \mathcal{M}$, and receives as output $\{C_i \leftarrow \mathsf{Enc}_K(T_i, M_i)\}_{i\in[q]}$ and $C_i \in \mathcal{C}$.
3. **Challenge Stage:** In this stage, the adversary submits two message-tweak pair (M_0, T_0) and (M_1, T_1), where $|M_0| = |M_1|$. The challenger chooses a random bit $b \leftarrow \{0,1\}$, compute $C_b = \mathsf{Enc}_K(M_b, T_b)$ and sends C_b to the adversary. The adversary $\mathcal{A}$ then outputs a bit b'.

We define the advantage of the attacker $\mathcal{A}$ as $Adv_{\mathcal{A}}^{\text{ELR-t-CPA}}(\lambda) = |\Pr[b' = b] - \frac{1}{2}|$, where ELR stands for entropic leakage-resilient.

It is very easy to see that under this modified definition our construction of leakage resilient tweakable holds. The only change happens in the proof of security where instead of length bounded leakage we use this notion of entropy loss of leakage.

6.2 After-the-Fact Leakage-resilient Tweakable Encryption.

In our definition of leakage-resilient (LR) tweakable encryption, we only allowed access to the leakage oracle to the adversary in the learning phase, but not in the challenge or post-challenge phase. The reason is that the adversary can encode the decryption function along with the two challenge messages and the challenge ciphertext as the leakage function to leak the bit b that we are trying to hide. This requires a meaningful formulation of the security model and definitions if we allow access to the leakage oracle to the adversary even after the challenge phase referred to as *After-the-Fact* leakage. Halevi and Lin [17] formulated a new notion of *entropic leakage-resilient* public key encryption which captures the intuition that as long as the entropy of the encrypted message is higher than the amount of leakage, the message still has some (pseudo) entropy left (even if this leakage was obtained after seeing the ciphertext). We port this intuition and definitions to the case of tweakable encryption.

Our security definition of After-the-fact leakage-resilient tweakable encryption consists of two games: one "*real*" game and another "*simulated*" game. Both games depend on several parameters: k is the a-priori min-entropy of the message, and ℓ_{pre}; ℓ_{post} control the amount of leakage in various parts of the games (namely the pre and post-challenge-ciphertext leakage bounds). Therefore, the leakage oracle $\mathcal{O}_K^{\ell_{pre},\ell_{post}}(.)$ is now parametrized by the secret key K and the two leakage parameters ℓ_{pre} and ℓ_{post}. All of these parameters are of course functions of the security parameter λ. For simplicity we will assume the message m is a uniform random k-bit string, but in general it may also come from arbitrary high min-entropy source $\mathcal{M}$ with min-entropy at least k.

Real Game. Given the parameters $(\lambda, \ell_{pre}, \ell_{post})$, and the tweakable encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, the real game is defined as follows:

1. **Initialization:** The challenger chooses a uniformly random $K \leftarrow \mathsf{KeyGen}(1^\lambda)$, and a uniform random message $M \leftarrow U_k$ and proceeds with the game as follows.
2. **Learning Stage:** This learning phase corresponds to the pre-challenge phase. The attacker $\mathcal{A}^{\mathcal{O}_K^{\ell_{pre},\ell_{post}}, \mathsf{Enc}_K(.,.)}$ gets oracle access to the leakage oracle $\mathcal{O}_K^{\ell_{pre},\ell_{post}}(.)$ which allows him/her to learn at most ℓ_{pre} bits of information about the secret key K. Besides the adversary can also query the encryption oracle $\mathsf{Enc}_K(.,.)$. The adversary queries the oracle with tuples $(T_i, M_i)_{i\in[q]}$ for some polynomial $q = q(\lambda)$, $T_i \in \mathcal{T}$ and $M_i \in \mathcal{M}$, and receives as output $\{C_i \leftarrow \mathsf{Enc}_K(T_i, M_i)\}_{i\in[q]}$ and $C_i \in \mathcal{C}$.
3. **Challenge Stage:** In this stage, the challenger chooses a random tweak $T \in \mathcal{T}$ and encrypts the message M under the tweak T and sends the ciphertext $C = (T, \mathsf{Enc}_K(T, M))$ to the adversary.
4. **Learning Stage:** This phase corresponds to the post challenge phase. The attacker $\mathcal{A}^{\mathcal{O}_K^{\ell_{pre},\ell_{post}(.)}}$ gets oracle access to the leakage oracle $\mathcal{O}_K^{\ell_{pre},l_{post}}(.)$ which allows him/her to learn at most ℓ_{post} bits of information about the secret key K.

We let $\mathsf{View}_{\mathcal{A}}^{\mathsf{rl}}(\Pi) = (randomness; T, f_{pre}(K); C; f_{post}(K))$ be the random variable describing the view of the adversary $\mathcal{A}$ in the game above, and by M^{rl} we denote the message that was chosen at the onset of this game. When we write the tuple $(M^{\mathsf{rl}}, \mathsf{View}_{\mathcal{A}}^{\mathsf{rl}})$, we mean the joint distribution of message M^{rl} and $\mathcal{A}$'s view in the real game with M^{rl}.

Simulated Game. In the simulated game we replace the challenger from above by a simulator Sim that interacts with $\mathcal{A}$ in any way that it sees fit. Sim gets a uniformly chosen message M^{sim} as input, and it needs to simulate the interaction conditioned on this M^{sim}. The view of $\mathcal{A}$ when interacting with $\mathcal{S}$ is denoted $\mathsf{View}^{\mathsf{sim}}_{\mathcal{A}}(\Pi)$. We say that Π is entropic leakage-resilient or After-the-Fact leakage-resilient if the views of the adversary in the two games namely, $\mathsf{View}^{\mathsf{rl}}_{\mathcal{A}}(\Pi)$ and $\mathsf{View}^{\mathsf{sim}}_{\mathcal{A}}(\Pi)$ are indistinguishable even given the message M, and the message M^{sim} has high min-entropy given $\mathsf{View}^{\mathsf{sim}}_{\mathcal{A}}(\Pi)$.

Definition 6. *Let λ, ℓ_{pre}, ℓ_{post} be parameters as above. A tweakable encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is $(\ell_{pre}, \ell_{post})$-after-the-fact leakage-resilient with respect to these parameters if there exists a simulator Sim, such that, for every PPT adversary $\mathcal{A}$ the following two conditions hold:*

- $(M^{rl}, \mathsf{View}^{rl}_{\mathcal{A}}(\Pi)) \approx_c (M^{sim}, \mathsf{View}^{sim}_{\mathcal{A}}(\Pi))$
- The average min-entropy of M^{sim} given $\mathsf{View}^{\mathsf{sim}}_{\mathcal{A}}(\Pi)$ is

$$\widetilde{\mathrm{H}}_{\infty}(M^{sim} \mid \mathsf{View}^{sim}_{\mathcal{A}}(\Pi)) \geq k - \ell_{post}$$

Construction of After-the-Fact LR-Tweakable encryption. Our construction of After-the-Fact LR-Tweakable encryption is essentially the same construction of LR-tweakable encryption from LR-tweakable weak PRFs from Section 4.3. Assume $\widetilde{\mathcal{E}} = \{E : \mathcal{K} \times \mathcal{T} \times \mathcal{D} \to \mathcal{R}\}$ is ℓ-LR-twPRF family, where the output domain $\mathcal{Y}$ and the tweak space $\mathcal{T}$ forms an additive group (e.g., bit-strings under XOR).

1. $\mathsf{KeyGen}(1^\lambda)$: Sample a uniformly random and independent $K \leftarrow \mathcal{K}$
2. $\mathsf{Enc}(M, K, T)$: The encryption algorithm takes as input $K \in \mathcal{K}$, message $M \in \mathcal{R}$, and a tweak $T \in \mathcal{T}$, samples $X \leftarrow \mathcal{D}$, $\hat{T} \leftarrow \mathcal{T}$ and computes
$$1)\ \widetilde{T} := T \oplus \widehat{T}, \ \text{ and } \ 2)\ \mathsf{Enc}_K(T, M) = (X, \widehat{T}, E_K(\widetilde{T}, X) \oplus M)$$
3. $\mathsf{Dec}(K, X, T, \widehat{T}, C)$: The decryption algorithm takes in the secret key K and the tuple (X, T, C) and computes $\widetilde{T} := T \oplus \widehat{T}$ and $M = C \oplus E_K(\widetilde{T}, X)$

Let us denote the challenge ciphertext as $C = (X^*, \widehat{T}^*, \psi)$, where $\psi = E_K(\widetilde{T}, X) \oplus M$. The main idea of the proof is also similar to the proof idea of this construction. Instead of randomly sampling $X^* \leftarrow \mathcal{D}$ and $\widetilde{T}^* \leftarrow \mathcal{T}$ (corresponding to $\mathsf{Dist}^{\mathcal{D}}_0$ and $\mathsf{Dist}^{\mathcal{T}}_0$ respectively) as in the real game, the simulator samples $X^* \leftarrow \mathsf{Dist}^{\mathcal{D}}_2(\mathsf{sampK})$ (invalid), $\widetilde{T}^* \leftarrow \mathsf{Dist}^{\mathcal{T}}_2(\mathsf{sampK})$ (invalid), while all the $(X_i, \widetilde{T}_i)$ values are sampled from valid distributions ($\mathsf{Dist}^{\mathcal{D}}_1(\mathsf{sampK})$ and $\mathsf{Dist}^{\mathcal{T}}_1(\mathsf{sampK})$ respectively). Now, by the *input indistinguishability* property of the underlying symmetric-key tweakable wHPS, these two distributions are indistinguishable even given the message m. This shows the indistinguishability of the real game and the simulated game. In the simulated game, since $X^* \leftarrow \mathsf{Dist}^{\mathcal{D}}_2(\mathsf{sampK})$ and $\widetilde{T}^* \leftarrow \mathsf{Dist}^{\mathcal{T}}_2(\mathsf{sampK})$, the value $Y = E_K(\widetilde{T}^*, X^*)$ in the challenge ciphertext is $\varepsilon(\lambda)$ close to the uniform distribution over $(\mathsf{log}(|\mathcal{R}|) - \ell(\lambda))$ bits, where ℓ denotes the maximum pre-challenge leakage bound the adversary can learn. This follows from the *smoothness* property of the underlying symmetric-key twHPS and from the property of the $(\log(|\mathcal{R}|) - \ell(\lambda), \varepsilon(\lambda))$- extractor $\mathsf{Ext} : \mathcal{R} \times \mathcal{S} \to \mathcal{Z}$ used in constructing the leakage-resilient t-wPRF as shown in Section 4.5. Thus, the min-entropy of m is ε-close to a uniform distribution over $\mathcal{Z}$. Further, since the post-challenge leakage is bounded by ℓ_{post} bits, the min-entropy of m is reduced by at most this much. So, it retains roughly $(\mathsf{log}(|\mathcal{Z}|) - \ell_{post})$ bits of entropy. This shows that our construction of leakage-resilient tweakable encryption is also resilient to After-the-Fact leakage.

7 Conclusion

Our work initiates the study of leakage-resilient (LR) tweakable encryption schemes from minimal assumptions. The construction that we show has a leakage rate of $O(\frac{\log\lambda}{\gamma(\lambda)})$, where $\gamma(\lambda)$ is the size of the secret key. During the process of the construction, several new primitives have been proposed. One such primitive we introduced was the symmetric-key tweakable weak hash proof system, which may be of independent interest. It will be interesting to explore implementation aspects of this scheme so as to bridge the gap between theory and practice. From the theoretical perspective, it will be interesting to explore further constructions of LR-tweakable encryptions in more generalized leakage settings like the continuous memory leakage model [9] or auxiliary input model [11]. Another possible extension of the current scheme would be to improve the overall leakage rate. Finally, we also leave open the constructions of leakage-resilient format preserving encryption and full disk encryption from leakage-resilient tweakable encryption.

References

1. Michel Abdalla, Sonia Belaïd, and Pierre-Alain Fouque. Leakage-resilient symmetric encryption via re-keying. In *Cryptographic Hardware and Embedded Systems-CHES 2013*, pages 471–488. Springer, 2013.
2. Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *Theory of Cryptography*, pages 474–495. Springer, 2009.
3. Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In *Advances in Cryptology–EUROCRYPT 2010*, pages 113–134. Springer, 2010.
4. Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *Advances in Cryptology-CRYPTO 2009*, pages 36–54. Springer, 2009.
5. Zvika Brakerski and Yael Tauman Kalai. A parallel repetition theorem for leakage resilience. In *Theory of Cryptography*, pages 248–265. Springer, 2012.
6. Zvika Brakerski, Yael Tauman Kalai, Jonathan Katz, and Vinod Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 501–510. IEEE, 2010.
7. Sherman SM Chow, Yevgeniy Dodis, Yannis Rouselakis, and Brent Waters. Practical leakage-resilient identity-based encryption from simple assumptions. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 152–161. ACM, 2010.
8. Whitfield Diffie and Martin E Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.
9. Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Cryptography against continuous memory attacks. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 511–520. IEEE, 2010.
10. Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Efficient public-key cryptography in the presence of key leakage. In *Advances in Cryptology-ASIACRYPT 2010*, pages 613–631. Springer, 2010.
11. Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 621–630. ACM, 2009.
12. Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM journal on computing*, 38(1):97–139, 2008.

13. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 293–302. IEEE, 2008.
14. Sebastian Faust, Krzysztof Pietrzak, and Joachim Schipper. Practical leakage-resilient symmetric cryptography. In *Cryptographic Hardware and Embedded Systems–CHES 2012*, pages 213–232. Springer, 2012.
15. Suvadeep Hajra, Chester Rebeiro, Shivam Bhasin, Gaurav Bajaj, Sahil Sharma, Sylvain Guilley, and Debdeep Mukhopadhyay. Drecon: Dpa resistant encryption by construction. In *International Conference on Cryptology in Africa*, pages 420–439. Springer, 2014.
16. J Alex Halderman, Seth D Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A Calandrino, Ariel J Feldman, Jacob Appelbaum, and Edward W Felten. Lest we remember: cold-boot attacks on encryption keys. *Communications of the ACM*, 52(5):91–98, 2009.
17. Shai Halevi and Huijia Lin. After-the-fact leakage in public-key encryption. In *Theory of Cryptography*, pages 107–124. Springer, 2011.
18. Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In *Advances in Cryptology-CRYPTO 2003*, pages 482–499. Springer, 2003.
19. Carmit Hazay, Adriana López-Alt, Hoeteck Wee, and Daniel Wichs. Leakage-resilient cryptography from minimal assumptions. In *Advances in Cryptology–EUROCRYPT 2013*, pages 160–176. Springer, 2013.
20. Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *Advances in Cryptology–ASIACRYPT 2009*, pages 703–720. Springer, 2009.
21. Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in CryptologyCRYPTO99*, pages 388–397. Springer, 1999.
22. Paul C Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in CryptologyCRYPTO96*, pages 104–113. Springer, 1996.
23. Moses Liskov, Ronald L Rivest, and David Wagner. Tweakable block ciphers. In *Advances in CryptologyCRYPTO 2002*, pages 31–46. Springer, 2002.
24. Silvio Micali and Leonid Reyzin. Physically observable cryptography. In *Theory of Cryptography*, pages 278–296. Springer, 2004.
25. Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. *SIAM Journal on Computing*, 41(4):772–814, 2012.
26. Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.
27. Sikhar Patranabis, Debapriya Basu Roy, and Debdeep Mukhopadhyay. Using tweaks to design fault resistant ciphers. In *VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), 2016 29th International Conference on*, pages 585–586. IEEE, 2016.
28. Krzysztof Pietrzak. A leakage-resilient mode of operation. In *Advances in Cryptology-EUROCRYPT 2009*, pages 462–482. Springer, 2009.
29. Emmanuel Prouff and Patrick Schaumont. *Cryptographic Hardware and Embedded Systems–CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012, Proceedings*, volume 7428. Springer, 2012.
30. Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes ocb and pmac. In *Advances in Cryptology-ASIACRYPT 2004*, pages 16–31. Springer, 2004.