Leakage-Resilient IBE/ABE with Optimal Leakage Rates from Lattices

Qiqi Lai[1,2], Feng-Hao Liu[3], and Zhedong Wang[4(✉)]

[1] School of Computer Science, Shaanxi Normal University, Xi'an, China
laiqq@snnu.edu.cn
[2] State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, China
[3] Florida Atlantic University, Boca Raton, FL, USA
fenghao.liu@fau.edu
[4] School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
wzdstill@sjtu.edu.cn

Abstract. We derive the first adaptively secure IBE and ABE for t-CNF, and selectively secure ABE for general circuits from lattices, with $1 - o(1)$ leakage rates, in the both relative leakage model and bounded retrieval model (BRM).

To achieve this, we first identify a new fine-grained security notion for ABE – partially adaptive/selective security, and instantiate this notion from LWE. Then, by using this notion, we design a new key compressing mechanism for identity-based/attributed-based weak hash proof system (IB/AB-wHPS) for various policy classes, achieving (1) succinct secret keys and (2) adaptive/selective security matching the existing non-leakage resilient lattice-based designs. Using the existing connection between weak hash proof system and leakage resilient encryption, the succinct-key IB/AB-wHPS can yield the desired leakage resilient IBE/ABE schemes with the optimal leakage rates in the relative leakage model. Finally, by further improving the prior analysis of the compatible locally computable extractors, we can achieve the optimal leakage rates in the BRM.

1 Introduction

Leakage-resilient cryptography aims to create crypto systems that maintain security even when partial information of the secret key is leaked. This line of studies is motivated by both theoretic curiosities and perhaps more importantly, real-world scenarios, where some secure crypto systems might be completely broken if some partial key leakage is given to the attackers. One famous example is the *side-channel attacks* where the adversary can obtain leakage from measuring some physical behavior of an implementation, e.g., [1,27]. Another source of leakage comes from imperfect erasure where the attacker can obtain partial information before the content is completely erased, e.g., the *cold boot attacks* [23]. On the other hand, leakage resilience can be used to achieve security for other more complicated systems. For example, in the design of non-malleable codes,

G. Hanaoka et al. (Eds.): PKC 2022, LNCS 13178, pp. 225–255, 2022.
https://doi.org/10.1007/978-3-030-97131-1_8

the work [17,26,31] leveraged leakage resilience to prove non-malleability. Therefore, leakage resilience has been an active research subject for the community, e.g., [4–6,8,16,25,33], to name a few.

Main Goal. As motivated above, we aim to determine how to derive encryption schemes with better leakage rates, stronger security, and more expressive access control functionalities. More specifically, our goal is to construct leakage resilient encryption schemes in both the relative leakage model and the bounded retrieval model (BRM) with (1) optimal leakage rates, i.e., $1 - o(1)$, (2) post-quantum security and (3) more fine-grained access control, i.e., IBE and ABE for various classes of policy functions.

The Leakage Models. Various leakage models have been studied in the literature, capturing information leaked to the adversary. This work focuses on a simple yet general model called the *bounded-leakage model* (also known as the *memory leakage model*), allowing the attacker to learn arbitrary information about the secret key sk, as long as the number of leaked bits is bounded by some parameter ℓ. This model has drawn a lot of attentions (e.g., [4,5,25,33]) for its elegance and simplicity, and can be used as a building block towards more sophisticated and realistic models, such as the continual leakage model [9,14] (see [25]). Thus, understanding this model is not only of theoretic interests but also a necessary step towards realizing security for broader physical attacks.

The bounded leakage model would require $\ell < |\mathsf{sk}|$, as otherwise, the attacker can trivially obtain the whole secret key, and thus no meaningful security can be attained. To further characterize this requirement, there are two important models studied in the literature that treat the relation between ℓ and sk in a different way: (1) *relative leakage model*, and (2) *bounded retrieval model* (BRM).

In the former, the secret key and public-key are chosen in the same way as a standard crypto system (not necessary leakage resilient), and then the leakage parameter ℓ would be determined. The latter model generalizes the former by considering ℓ as an independent parameter whose growth (essentially) only goes with $|\mathsf{sk}|$, but would barely affect the other parameters, such as the public-key size, encryption running time, and ciphertext size. Basically, both models can scale up ℓ to allow an arbitrarily long leakage. But their difference is that the former would require to scale up the security parameter and thus all the other parameters, while the latter would only scale up the secret-key size and keep the other parameters essentially the same. Thus, constructions in the BRM is more desirable yet more challenging.

Leakage rate, i.e., the ratio $\ell/|\mathsf{sk}|$, is an important measure of efficiency for crypto systems in these two models. Particularly, rate $1-o(1)$ is the best we can hope for – in order to tolerate ℓ bits of leakage, the system only needs to scale $|\mathsf{sk}|$ slightly larger than ℓ, optimizing the security/efficiency tradeoff.

Current State of the Arts and Challenges. We first notice that for the pre-quantum settings, leakage resilience can be achieved via the beautiful framework – *dual system encryption*, even for IBE/ABE and with optimal leakage rates,

e.g., [28]. However, current instantiations of the dual system encryption are all group-based [11,20,28,29,41,42], and thus cannot defend against quantum algorithms. It is an interesting yet extremely challenging open question how to instantiate a dual system from a post-quantum candidate, such as LWE or LPN.

For post-quantum leakage resilient encryption schemes, we notice that there are some limitations of the current techniques in achieving the optimal leakage rate beyond the basic PKE. In prior work, there have been constructed LWE/LPN-based PKE schemes with leakage rates $1-o(1)$, e.g., [10,13], but their ideas do not generalize to more advanced settings, such as IBE and ABE. In a subsequent work, Hazay et al. [25] proposed a unified framework, showing that (1) PKE implies leakage resilient PKE in the relative leakage model, and (2) IBE implies leakage resilient PKE/IBE in the BRM. Moreover, the leakage resilient IBE achieves the same level of adaptive/selective security as that of the underlying IBE. Their idea can be generalized to construct leakage resilient ABE, but this approach inherently yields a very low leakage rate (i.e., $1/O(\lambda)$).

A recent work [35] somewhat mitigated this issue by improving the leakage rates, yet at the cost of weaker security guarantees for the post-quantum instantiations. Particularly, they construct LWE-based leakage resilient IBE schemes in both the relative leakage model and the BRM, achieving $1 - o(1)$ leakage rate in the former and $1 - O(1)$ (for any arbitrarily small constant) in the latter. Their improvement relies on a novel *key-compression mechanism* that shortens the secret key length required in the framework of Hazay et al. [25]. Due to some technical limitation in the mechanism, their IBE scheme however, can only achieve the selective security. From these works [25,35], we see a tradeoff between security and leakage rate, i.e., either we have an adaptively secure IBE with a low leakage rate, or a selectively secure IBE with a higher leakage rate.

Main Question. In this work, we aim to further determine whether the tradeoff between (selective/adaptive) security and leakage rates as above is inherent. Particularly, we ask the following:

> *Can we achieve the optimal leakage rate ($1-o(1)$) for* IBE *(and* ABE *) in both relative and bounded retrieval models with security matching existing non-leakage resilient* IBE *(*ABE *), under* LWE*?*

1.1 Our Contributions

In this work, we give positive answers in many settings of the main question. Our central idea is a refinement of the framework of [25,35] by designing a new key compression mechanism from ABE *with succinct keys*. Below we describe our contributions in more details.

- As a warm-up, we propose a new leakage model for ABE that incorporates parameters ℓ and ω, where ℓ is the number of bits allowed to leak per key and ω is the number of keys the adversary can leak. We note that for PKE and IBE, there is only one possible secret key corresponding to the challenge id.

In this case, it is without loss of generality to just consider $\omega = 1$. However, for the ABE setting, there could be many possible secret keys corresponding to the challenge attribute, so specifying ω is natural and necessary in the leakage model. We call a scheme (ℓ, ω)-leakage resilient if the scheme can tolerate leakage on ω keys, each within ℓ bits.

- Next, we design improved instantiations of attribute-based weak hash proof system (AB-wHPS), which generalizes (identity-based) weak hash proof system [5,25] by associating each ciphertext with an attribute and each secret key with a policy function. Particularly, we construct lattice-based AB-wHPS from ABE for various function classes, achieving two important new features: (1) succinct secret keys, i.e., the secret key length is $|f| + o(|f|)$ where f is the policy function, and (2) security matching currently the best known lattice-based ABE schemes (not necessarily leakage resilient). More specifically, we construct adaptively secure AB-wHPS for the class of comparison functions (which is the IB-wHPS) and the class t-CNF*[1], and selectively secure AB-wHPS for general circuits.
- By using AB-wHPS for class $\mathcal{F}$ with *succinct keys*, we are able to construct $(\ell, 1)$-leakage resilient ABE for $\mathcal{F}$, with leakage rate $\ell/|\mathsf{sk}| = (1 - o(1))$ in the relative leakage model.
  We view AB-wHPS with succinct key as an improved key compression mechanism from prior works [25,35] in the following two aspects: (1) AB-wHPS has better expressibility of policy function (the prior work [35] can only express the comparison function), and (2) we can derive adaptively secure AB-wHPS with succinct keys for classes which we have adaptively secure (non-leakage resilient) ABE. Prior to our work, for lattice-based schemes, we only had either a selectively secure IB-wHPS with succinct secret keys [35] or an adaptively secure IB-wHPS with non-succinct keys [25].
- From our AB-wHPS, we can further derive $(\ell, 1)$-leakage resilient ABE in the BRM, via an amplification and a connection with locally computable extractors as pointed out by [25]. However, prior compatible locally computable extractors [5] can only achieve $1 - O(1)$ leakage rate for an arbitrarily small constant. To achieve $1 - o(1)$ leakage rate, we improve the prior analysis [5] by refining their proof technique via the framework of Vadhan [40].
- Finally, we present a bootstrapping mechanism that generalizes our prior $(\ell, 1)$-leakage resilient ABE schemes to (ℓ, ω)-leakage resilient schemes for any bounded polynomial ω, in both relative leakage model and bounded retrieval model. The resulting leakage rate is still optimal (i.e., $1 - o(1)$) against block leakage functions, a slightly more restricted class.

1.2 Overview of Our Techniques

Our central insight is a new key-compression mechanism for the framework in [25]. To illustrate our new idea, we first briefly review the prior framework [25]

[1] This is the dual class of t-CNF where the function is an assignment x and attribute is a description of t-CNF. We use the dual class as we are working on Key-policy ABE while the prior work [38] worked on Ciphertext-policy ABE.

and point out the barrier of their leakage rates. Then we will describe our new ideas for the improvement.

(Weak) Hash Proof System. A hash proof system can be described as a key encapsulation mechanism that consists of four algorithms $(\mathsf{Setup}, \mathsf{Encap}, \mathsf{Encap}^*, \mathsf{Decap})$: (1) Setup outputs a key pair $(\mathsf{pk}, \mathsf{sk})$, (2) $\mathsf{Encap}(\mathsf{pk})$ outputs a pair (CT, k) where k is a key encapsulated in a "valid" ciphertext CT, (3) $\mathsf{Encap}^*(\mathsf{pk})$ outputs an "invalid" ciphertext CT^*, and (4) $\mathsf{Decap}(\mathsf{sk}, \mathsf{CT})$ outputs a key k'. A (weak) hash proof system requires the following:

- **Correctness.** For a valid ciphertext CT, Decap always outputs the encapsulated key $k' = k$, i.e., $\mathsf{Decap}(\mathsf{sk}, \mathsf{CT}) = k$, where $(\mathsf{CT}, k) \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk})$.
- **Ciphertext Indistinguishability.** Valid ciphertexts and invalid ciphertexts are computationally indistinguishable, *even given the secret key*. This condition is essential for achieving leakage resilience [5,33].
- **Universality.** The decapsulation of an invalid ciphertext has information entropy, even for unbounded adversaries. Here, the randomness of invalid decapsulation comes from randomness in generating secret keys. A weak HPS (wHPS) only requires this property to hold for a random invalid ciphertext, i.e. $\mathsf{CT}^* \xleftarrow{\$} \mathsf{Encap}^*(\mathsf{pk})$, while a full-fledged HPS requires this to hold for any invalid ciphertext.

As noted in prior work [5], a wHPS already suffices to achieve leakage resilience, though it is not sufficient for the CCA2 security, for which the HPS was originally intended to design [12]. Roughly speaking, the leakage resilient scheme derived from wHPS [5,25,33] can tolerate $\ell \approx |k| - \lambda$ bits of leakage, i.e., the length of encapsulated key minus security parameter, and thus the leakage rate of the derived encryption scheme would be $\ell/|\mathsf{wHPS.sk}| \approx \frac{|k|-\lambda}{|\mathsf{wHPS.sk}|}$.

Moreover, the idea can be generalized to IB-wHPS and AB-wHPS where an additional id or attribute $\boldsymbol{x}$ is associated with the ciphertext, and id or a policy function f is associated with the secret key. In the same way [25], IB-wHPS and AB-wHPS suffice to derive leakage resilient IBE and ABE.

wHPS from Any PKE and Generalizations [25]**.** While there were several instantiations of wHPS from specific assumptions [5,33], Hazay et al. [25] showed somewhat surprisingly, any PKE implies wHPS. Their construction [25] can be thought as the following two steps: (1) construct a basic wHPS that only outputs 1 bit (or $\log \lambda$-bits), (2) amplify the output of the wHPS via parallel repetition. As pointed out in the work [25], parallel repetition might not amplify HPS in general, yet it does for wHPS as required in the application of leakage resilience.

The basic wHPS is simple: given any $\mathsf{PKE} = (\mathsf{Enc}, \mathsf{Dec})$, the $\mathsf{wHPS.pk}$ consists of two public keys $\mathsf{pk}_0, \mathsf{pk}_1$ from PKE, and $\mathsf{wHPS.sk}$ is (b, sk_b) for a random bit b where sk_b corresponds to pk_b. The Encap algorithm outputs a valid ciphertext $\mathsf{CT} = (\mathsf{Enc}_{\mathsf{pk}_0}(k), \mathsf{Enc}_{\mathsf{pk}_1}(k))$ to encapsulate a uniformly random key $k \in \{0,1\}$. The Encap^* algorithm outputs an invalid ciphertext $\mathsf{CT}^* = (\mathsf{Enc}_{\mathsf{pk}_0}(k), \mathsf{Enc}_{\mathsf{pk}_1}(1-k))$ for a uniformly random bit k. With a parallel repetition of n times, i.e.,

$\mathsf{wHPS}_{\|}.\mathsf{pk} := \{\mathsf{pk}_{i,0}, \mathsf{pk}_{i,1}\}_{i\in[n]}$ and $\mathsf{wHPS}_{\|}.\mathsf{sk} := \{(i, b_i), \mathsf{sk}_{i,b_i}\}_{i\in[n]}$, we can get a $\mathsf{wHPS}_{\|}$ with $|k| = n$ for an arbitrarily large $n \gg \lambda$, and thus a leakage resilient encryption that tolerates $\ell = n - \lambda \approx n - o(|\mathsf{wHPS}_{\|}.\mathsf{sk}|)$.

Naturally, this elegant approach can be generalized to construct IB-wHPS and AB-wHPS for class $\mathcal{F}$ from any IBE and ABE for $\mathcal{F}$, and the (adaptive/selective) security of the IB-wHPS and AB-wHPS matches the underlying IBE and ABE. Therefore, this framework provides a powerful way to design leakage resilient IBE and ABE from any IBE and ABE that can tolerate an arbitrarily large leakage ℓ.

Technical Challenges from Prior Work. This technique of [25] achieves almost everything one would desire, except for the leakage rate. The main reason comes from the secret key size of $\mathsf{wHPS}_{\|}$, which is also scaled up by the parallel repetition, resulting in a low leakage rate as $\frac{\ell}{|\mathsf{wHPS}_{\|}.\mathsf{sk}|} = \frac{n-o(|\mathsf{wHPS}_{\|}.\mathsf{sk}|)}{|\mathsf{wHPS}_{\|}.\mathsf{sk}|} \approx \frac{n-o(n|\mathsf{PKE.sk}|)}{n|\mathsf{PKE.sk}|} \approx \frac{1}{|\mathsf{PKE.sk}|}$. To further improve the rate, it suffices to decrease $|\mathsf{wHPS.sk}|$ as observed by [35]. In particular, if we can shrink the secret key size of the wHPS to roughly $|\mathsf{wHPS}_{\|}.\mathsf{sk}| \approx n + |\mathsf{PKE.sk}|$, then the leakage rate would be $\frac{n-o(|\mathsf{wHPS}_{\|}.\mathsf{sk}|)}{|\mathsf{wHPS}_{\|}.\mathsf{sk}|} \approx \frac{n-o(n+|\mathsf{PKE.sk}|)}{n+|\mathsf{PKE.sk}|} \approx 1 - o(1)$, for sufficiently large n. Therefore, now the goal becomes to design a compact form of $\mathsf{wHPS}_{\|}.\mathsf{sk}$ that can encode n possible keys in a succinct way.

The work [35] achieved this goal and the more general IB-wHPS by proposing a novel key compression mechanism from a new primitive called *multi*-IBE. Then they instantiated the required multi-IBE from *inner-product encryption* (IPE) [3, 11, 42] with succinct keys. However, for lattice-based IPE schemes [3], only the selective security can be achieved under currently known techniques. Thus, the work [35] can only derive selectively secure leakage resilient IBE from lattices.

At this point, we summarize two limitations from the prior key compression mechanism [35]: (1) the approach is tied to IBE/IB-HPS, and it is unclear whether we can further generalize the technique for further expressive policies, i.e., ABE; (2) the lattice-based instantiations are only selectively secure under currently known techniques. Below we show our new ideas to break these limitations.

Our New Key Compression Mechanism. We first present a new key compression mechanism that can be generalized to more expressive policy functions, i.e., ABE. To illustrate our core insight, we first describe how to use the technique of key-policy (KP)-ABE to encode $\mathsf{wHPS}_{\|}.\mathsf{sk}$ succinctly. The idea can be naturally generalized to compress IB-wHPS and AB-wHPS. To facilitate further discussions, we first recall the concept of KP-ABE.

In a KP-ABE scheme, a secret key is associated with a policy function $f : \{0,1\}^* \to \{0,1\}$, and a ciphertext is associated with an attribute $\boldsymbol{x}$. The secret key can decrypt and recover the encrypted message if and only if $f(\boldsymbol{x}) = 1$.

Now we explain our key compression mechanism. Let us describe the format of a valid ciphertext of $\mathsf{wHPS}_{\|}$ as $\mathsf{CT} := \left\{\mathsf{Enc}_{\mathsf{pk}_{i,0}}(k_i), \mathsf{Enc}_{\mathsf{pk}_{i,1}}(k_i)\right\}_{i\in[n]}$, and a secret key is of the form $\{(i, b_i), \mathsf{sk}_{i,b_i}\}_{i\in[n]}$. From another angle looking

at the ciphertext, we can view the indices (i, b)'s as attributes in an ABE, i.e. $\mathsf{CT} := \{\mathsf{ABE.Enc}(\mathsf{mpk}, (i, 0), k_i), \mathsf{ABE.Enc}(\mathsf{mpk}, (i, 1), k_i)\}_{i\in[n]}$. Then we can use a single ABE secret key to encode the set of keys $\{(i, b_i), \mathsf{sk}_{i,b_i}\}_{i\in[n]}$ as follows. Let $\boldsymbol{b} = (b_1, b_2, \ldots, b_n) \in \{0, 1\}^n$ be a binary vector, and define the following policy function $g_{\boldsymbol{b}}(i, z) = 1$ iff $b_i = z$ for each $i \in [n]$. In this way, only this set of attributes $\{(i, b_i)\}_{i\in[n]}$ satisfies the policy function $g_{\boldsymbol{b}}$, so the ABE decryption algorithm with $\mathsf{sk}_{g_{\boldsymbol{b}}}$ can successfully recover the encrypted messages from $\{\mathsf{ABE.Enc}(\mathsf{mpk}, (i, b_i), k_i)\}_{i\in[n]}$. The other part of the ciphertext, i.e., $\{\mathsf{ABE.Enc}(\mathsf{mpk}, (i, 1 - b_i), k_i)\}_{i\in[n]}$ is hidden by the security of the ABE. This approach can be naturally extended to the setting of IB-wHPS and AB-wHPS by adding an additional string $\boldsymbol{x} \in \{0, 1\}^*$ (either an ID or general attribute) to the existing attributes as above, resulting in ciphertexts of the form $\mathsf{CT} := \{\mathsf{ABE.Enc}(\mathsf{mpk}, (\boldsymbol{x}, i, 0), k_i), \mathsf{ABE.Enc}(\mathsf{mpk}, (\boldsymbol{x}, i, 1), k_i)\}_{i\in[n]}$. It is not hard to check these designs satisfy the requirements of (IB/AB)-wHPS.

Here we can conclude: (1) $\mathsf{sk}_{g_{\boldsymbol{b}}}$ is functionally equivalent to the set of secret keys $\{(i, b_i), \mathsf{sk}_{i,b_i}\}_{i\in[n]}$, and (2) as long as $\mathsf{sk}_{g_{\boldsymbol{b}}}$ has a succinct representation, i.e., $|\mathsf{sk}_{g_{\boldsymbol{b}}}|$ only depends on the depth but not the size of the function $g_{\boldsymbol{b}}$ when $g_{\boldsymbol{b}}$ is given, we can achieve the optimal leakage rate. We can instantiate the desired ABE by the lattice-based schemes [7,22], and consequently derive a PKE/IBE/ABE with the optimal rate in the relative leakage model.

Adaptive Security for Various Function Classes. A careful reader may already observe that the underlying ABE schemes of [7,22] do not achieve adaptive security, and neither do the IB-wHPS and AB-wHPS as constructed above. Moreover, it seems that lattice-based ABE that supports the computation $g_{\boldsymbol{b}}(\cdot)$ with succinct keys (e.g., general circuits [7,22]) can only achieve selective security. Thus, existing techniques plus the above approach do not suffice for our goal on adaptive security.

To overcome the limitation, we further observe that our constructions of IB-wHPS and AB-wHPS above actually *do not* require the full adaptive security of the whole attribute $(\boldsymbol{x}, (i, b))$ from the underlying ABE. We only need the selective security over the second part (i, b), as this part is generated by the honest key generation algorithm, instead of being challenged by the adversary.

With this insight, we define a more fine-grained security notion that considers partially adaptive/selective security over partitioned attributes $(\boldsymbol{x}, (i, b))$. Intuitively, if the underlying ABE is adaptively (or selectively) secure over $\boldsymbol{x}$ and *selective secure* over (i, b), then we can prove the AB-wHPS is adaptively (resp. selectively) secure. Furthermore we instantiate the required partially adaptive-selective ABE for various function classes. As a result, we obtain an adaptively secure IB-wHPS and AB-wHPS for t-CNF*, and selectively secure AB-wHPS for general circuits. This matches the function classes for which we know how to construct adaptively secure ABE without leakage.

Application. Our AB-wHPS with succinct keys immediately yields a $(\ell, 1)$-leakage resilient ABE with leakage rate $1 - o(1)$ in the relative leakage model, followed from the framework [25]. More specifically, by using our adaptively secure AB-wHPS for the comparison function (i.e., IB-wHPS) and the t-CNF* functions, we get leakage resilient and adaptively secure ABE for these classes with optimal leakage rates. Additionally, we can have selectively secure leakage resilient ABE for general circuits, with leakage rate $1 - o(1)$.

Extension I. As pointed out by [25], we can further derive $(\ell, 1)$-leakage resilient ABE in the BRM from AB-wHPS, via an amplification and a connection with locally computable extractors [40]. However, the analysis from prior compatible locally computable extractors only yields $1 - O(1)$ rate for the leakage resilient encryption scheme. It was left as an interesting open question by [35] how to improve the analysis of the extractor. We solve this open question by improving the analysis of the sampler [5] required by the general construction of Vadhan [40]. With our improved analysis, we are able to achieve $1 - o(1)$ leakage rate in the BRM.

Extension II. Finally, we show how to derive (ℓ, ω)-leakage resilient ABE with the optimal leakage rate in the block leakage setting for both relative model and BRM, for any bounded polynomial ω. Inspired by the work [21], we derive a new bootstrapping mechanism by connecting secret sharing with our AB-wHPS. We leave it as an interesting open question how to achieve leakage resilient ABE even for an unbounded polynomial ω.

1.3 Other Related Work

AB-wHPS has been studied to construct leakage resilient ABE schemes in [43,44]. Particularly, in [43], the authors focus on AB-wHPS supporting linear secret sharing schemes as the policy function class, from the pre-quantum decisional bilinear Diffie-Hellman assumption. The work in [44] constructed an AB-wHPS from a post-quantum, i.e., LWE, assumption. However, the constructions only achieve selective security for linear secret sharing schemes. And both of these related work only consider security in the relative leakage model. Compared with the prior works, our design/analysis approach is more modular, supporting broader function classes and/or stronger (adaptive) security.

2 Preliminaries

We use several standard mathematical notations, whose detailed descriptions are deferred to the full version of this paper, due to space limit.

2.1 Attribute-Based Encryption (ABE)

Definition 2.1 (ABE [37]). *An attribute-based encryption (*ABE*) scheme for a function class* $\mathcal{F}_\lambda = \{f : \mathcal{X}_\lambda \to \{0,1\}\}$ *consists of four algorithms* $\mathsf{ABE}.\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$ *as follows.*

- **Setup.** $\mathsf{ABE.Setup}(1^\lambda)$ *takes a security parameter* λ *as input, and generates a pair of master public key and master secret key* $(\mathsf{mpk}, \mathsf{msk})$*, where* mpk *contains the attribute space* $\mathcal{X}_\lambda$*, message space* $\mathcal{M}$ *and ciphertext space* $\mathcal{CT}$*.*
- **Key generation.** $\mathsf{ABE.KeyGen}(f, \mathsf{msk})$ *takes as input a function* $f \in \mathcal{F}_\lambda$ *and the master secret key* msk*, and generates a secret key* (f, sk_f)*. Without loss of generality, we think the secret key contains two parts, the function description* f*, and an extra* sk_f*. The secret key is succinct if* $|\mathsf{sk}_f| = o(|f|)$*. When the context is clear, we often omit the description of* f*.*
- **Encryption.** $\mathsf{ABE.Enc}(\mathsf{mpk}, \boldsymbol{x}, \mu)$ *takes as input the master public key* mpk*, an attribute* $\boldsymbol{x} \in \mathcal{X}_\lambda$ *and a message* $\mu \in \mathcal{M}$*, and outputs a ciphertext* $\mathsf{ct} \in \mathcal{CT}$*.*
- **Decryption.** $\mathsf{ABE.Dec}(\mathsf{sk}_f, \mathsf{ct})$ *takes as input a secret key* sk_f *and a ciphertext* c*, and outputs* $\mu \in \mathcal{M}$ *if* $f(\boldsymbol{x}) = 1$ *and* $\perp$ *if* $f(\boldsymbol{x}) = 0$*, where* $\boldsymbol{x}$ *is the corresponding attribute used to generate* ct*.*

Correctness. We require that for all $f \in \mathcal{F}$, $\boldsymbol{x} \in \mathcal{X}_\lambda$, $\mu \in \mathcal{M}$, for correctly generated $(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{ABE.Setup}(1^\lambda)$, $\mathsf{sk}_f \xleftarrow{\$} \mathsf{ABE.KeyGen}(\mathsf{msk}, f)$ and $\mathsf{ct} \xleftarrow{\$} \mathsf{ABE.Enc}(\mathsf{mpk}, \boldsymbol{x}, \mu)$, it holds that

- if $f(\boldsymbol{x}) = 1$, $\Pr[\mathsf{ABE.Dec}(\mathsf{sk}_f, \mathsf{ct}) = \mu] \geq 1 - \mathsf{negl}(\lambda)$.
- if $f(\boldsymbol{x}) = 0$, $\Pr[\mathsf{ABE.Dec}(\mathsf{sk}_f, \mathsf{ct}) = \perp] \geq 1 - \mathsf{negl}(\lambda)$.

Leakage Resilience in the Relative Leakage Model

Next, we give the formal definition of leakage-resilient key-policy ABE.

Definition 2.2 (Leakage-Resilient ABE). *A leakage-resilient* ABE *with attribute space* $\mathcal{X}_\lambda$ *for a class of functions* $\mathcal{F}_\lambda = \{f : \mathcal{X}_\lambda \to \{0,1\}\}$ *in the relative leakage model consists of four algorithms* $\mathsf{ABE}.\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$*, which are parameterized by a security parameter* λ *and leakage parameters* ℓ, ω*. In particular,* (ℓ, ω)*-leakage-resilient security can be defined by the following experiment.*

Experiment $\mathbf{Exp}^{\mathsf{LR}}_{\mathsf{ABE},\mathcal{A}}(\lambda,\ell,\omega)$

Attribute Challenge: *In the setting of selective case, $\mathcal{A}$ chooses an challenge attribute $\boldsymbol{x}^* \in \mathcal{X}_\lambda$ before the Setup stage and sends it to $\mathcal{C}$; In the setting of adaptive case, $\mathcal{A}$ chooses an challenge $\boldsymbol{x}^* \in \mathcal{X}_\lambda$ in the challenge stage, and sends it to $\mathcal{C}$.*

Test Stage 1: *$\mathcal{A}$ adaptively queries the challenger $\mathcal{C}$ with function $f \in \mathcal{F}_\lambda$. For each query, $\mathcal{C}$ responds with (f, sk_f) if $f(\boldsymbol{x}^*) \neq 1$ and $\perp$ otherwise.*

ω-Leakage Queries Stage: *$\mathcal{A}$ adaptively queries the challenger $\mathcal{C}$ with q pairs (f_i, h_i) for $i \in [\omega]$, where f_i is a policy function such that $f_i(\boldsymbol{x}^*) = 1$, and $h_i : \{0,1\}^* \to \{0,1\}^\ell$ is a leakage function. The adversary gets $h_i(\mathsf{sk}_{f_i})$ from $\mathcal{C}$.*

Challenge Stage: *$\mathcal{A}$ chooses two messages μ_0, $\mu_1 \in \mathcal{M}$ and sends them to $\mathcal{C}$. Then $\mathcal{C}$ chooses $b \xleftarrow{\$} \{0,1\}$ and computes $\mathsf{ct}_b \xleftarrow{\$} \mathsf{ABE.Enc}(\mathsf{mpk}, \boldsymbol{x}^*, \mu_b)$. Finally, $\mathcal{C}$ returns ct_b to $\mathcal{A}$.*

Test Stage 2: *$\mathcal{A}$ adaptively queries the challenger $\mathcal{C}$ with function $f \in \mathcal{F}_\lambda$. Then $\mathcal{C}$ responds with $(f, \mathsf{sk}_{\mathsf{id},f})$ if $f(\boldsymbol{x}^*) \neq 1$ and $\perp$ otherwise.*

Output: *The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.*

We define the advantage of $\mathcal{A}$ in the above experiment[2] *to be*

$$\mathbf{Adv}^{\mathsf{LR}}_{\mathsf{ABE},\mathcal{A}}(\lambda,\ell,\omega) = |\Pr[b = b'] - 1/2|\,.$$

The scheme is (ℓ,ω)-leakage resilient if for any PPT *adversary $\mathcal{A}$, we have $\mathbf{Adv}^{\mathsf{LR}}_{\mathsf{ABE},\mathcal{A}}(\lambda,\ell,\omega) \leq \mathsf{negl}(\lambda)$, and the leakage rate of this* ABE *is $\frac{\ell}{|\mathsf{sk}|}$.*

Furthermore, the scheme is abbreviated as ℓ-leakage resilient if $\omega = 1$ in the above experiment.

Remark 2.3. *We use the parameter ω to denote the number of different challenge keys that can be conducted leakage queries. For* PKE *and* IBE*, we have $\omega = 1$ as for these two settings, there is a unique challenge key corresponding to the challenge attribute. For the more general* ABE*, there might be many different "1"-keys corresponding to the challenge attribute. Thus, this parameter ω would be an important specification for the leakage resilient* ABE*.*

Remark 2.4. *In our security model, the adversary can obtain leakage on ω secret keys adaptively one after another. The secret keys would then form a block-source under the leakage.*[3] *We note that it is possible to generalize the model where the leakage function takes inputs all the ω secret keys. In this work, we focus mainly on the block-source setting, as it already captures many useful scenarios.*

[2] Notice that in the above experiment $\mathbf{Exp}^{\mathsf{LR}}_{\mathsf{ABE},\mathcal{A}}(\lambda,\ell,\omega)$, we allow the adversary to interleave key queries in *Test Stage 1* and leakage queries in *ω-Leakage queries Stage*, in an arbitrary way.

[3] For the case that $\mathsf{sk} := S = (S_1,\ldots,S_m)$ is an $m \times e$ block source as in [39], we define leakage functions $f_i : \{0,1\}^* \to \{0,1\}^\ell$ independently for each block S_i with all $i \in [m]$. We say $(f_1,\ldots,f_m)$ are block leakage functions, if the min-entropy of S_i is still large enough even given leakage $(f_1(S_1),\ldots,f_{i-1}(S_{i-1}))$ for any $i \in [m]$. Clearly, when $m = 1$, this is the trivial case in Definition 2.2. Here, we call $\frac{m\ell}{|\mathsf{sk}|}$ the block leakage rate of the corresponding scheme.

Leakage Resilience in the BRM.
Below, we generalize to the setting of ABE the definition of leakage-resilience in the BRM by Alwen et al. [5].

Definition 2.5. (ABE in the BRM). *An* ABE *for attribute space* $\mathcal{X}_\lambda$ *and policy function class* $\mathcal{F} := \{\mathcal{X}_\lambda \to \{0,1\}\}$ *is* (ℓ, ω)*-leakage resilient in the* BRM *if its master public-key size, ciphertext size, encryption time and decryption time (and the number of secret-key bits used by decryption) are independent of the leakage-bound* ℓ*. Besides, in the leakage resilient experiment, the adversary is allowed to conduct key leakage attacks on* ω *secret keys corresponding to the challenge attribute. More formally, there exist polynomials* $\mathsf{mpksize}, \mathsf{ctsize}, \mathsf{encT}, \mathsf{decT}$*, such that, for any polynomial* ℓ *and any* $(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{ABE.Setup}(1^\lambda, 1^{\ell(\lambda)})$*,* $\boldsymbol{x} \in \mathcal{X}_\lambda$*,* $\mu \in \mathcal{M}$*,* $\mathsf{ct} \xleftarrow{\$} \mathsf{ABE.Enc}(\mathsf{mpk}, \boldsymbol{x}, \mu)$*, the scheme satisfies:*

1. *Master public-key size is* $|\mathsf{mpk}| \leq O(\mathsf{mpksize}(\lambda))$*, ciphertext size is* $|\mathsf{ct}| \leq O(\mathsf{ctsize}(\lambda, |\mu|))$*.*
2. *Run-time of* $\mathsf{ABE.Enc}(\mu, \mathsf{pk})$ *is bounded by* $O(\mathsf{encT}(\lambda, |\mu|))$*.*
3. *Run-time of* $\mathsf{ABE.Dec}(\mathsf{ct}, \mathsf{sk}_f)$ *and the number of bits of* sk_f *used in this decryption bounded by* $O(\mathsf{decT}(\lambda, |\mu|))$*, where* $\mathsf{sk}_f \xleftarrow{\$} \mathsf{ABE.KeyGen}(\mathsf{msk}, f)$ *with* $f \in \mathcal{F}$ *such that* $f(\boldsymbol{x}) = 1$*. Here we assume that the secret key* sk_f *is stored in a random access memory (RAM), and the decryption algorithm* $\mathsf{ABE.Dec}(\mathsf{ct}, \cdot)$ *only needs to read partial bits of* sk_f *to decrypt.*

The leakage rate of this scheme is defined as $\frac{\ell}{|\mathsf{sk}_f|}$*. Furthermore, the scheme is abbreviated as* ℓ*-leakage resilient if the parameter* $\omega = 1$ *in the experiment.*

Policy Function Classes. This work considers three function classes: (1) ID comparison functions, (2) t-CNF^* formulas, and (3) general circuits. (1) and (3) are clear from the literature. We elaborate on (2). First we present the definition of the function class t-CNF.

Definition 2.6 (t-CNF [38]). *A* t-CNF *policy* $f : \{0,1\}^\ell \to \{0,1\}$ *is a set of classes* $f = \{(T_i, f_i)\}_i$*, where for all* $i, T_i \subseteq [\ell], |T_i| = t$ *and* $f_i : \{0,1\}^t \to \{0,1\}$*. For all* $x \in \{0,1\}^\ell$ *the value of* $f(x)$ *is computed as* $f(x) = \bigwedge_i f_i(x_{T_i})$*, where* x_T *is the length-*t *bit-string consisting of the bits of* x *in the indices* T*. A function class* $\mathcal{F}$ *is* t-CNF *if it consists only of* t-CNF *policies for some fixed* $\ell \in \mathbb{N}$ *and a constant* $t \leq \ell$*. If* $\mathcal{F}$ *is a* t-CNF *class, we say that* t *is the* CNF *locality of* $\mathcal{F}$*.*

In this paper, we use the "dual" form of t-CNF, called t-CNF^*. The use of the dual version is because the prior work [38] worked on the ciphertext-policy ABE for t-CNF, and this work presents the result in the key-policy setting.

Definition 2.7 (t-CNF*). *For any* $x \in \{0,1\}^\ell$ *(the domain of* t-CNF*), let* $U_x(\cdot)$ *denote the function for which* x *is hardwired into* $U_x(\cdot)$*, and* $U_x(\cdot)$ *takes* $f \in t$-CNF *as input and outputs* $U_x(f)$ *such that* $U_x(f) = f(x)$*.* $U_x(\cdot)$ *is uniquely determined by* x*. We denote the function class* $\{U_x(\cdot)\}$ *as* t-CNF^**.*

2.2 Entropy and Extractors

Definition 2.8 (Min-Entropy). *The min-entropy of a random variable X, denoted as $H_\infty(X)$ is defined as* $H_\infty(x) = -\log\left(\max_{x_0 \in X} \Pr[x = x_0]\right)$.

Definition 2.9 (Average-Conditional Min-Entropy [15]). *The average-conditional min-entropy of a random variable X conditioned on a correlated variable Z, denoted as $H_\infty(X|Z)$ is defined as*

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

This notion of conditional min-entropy measures the best guess for X by an adversary that may observe an average-case correlated variable Z.

Lemma 2.10 ([15]). *Let X,Y,Z be arbitrarily correlated random variables where the support of Y has at most 2^ℓ elements. Then $H_\infty(X|(Y,Z)) \geq H_\infty(X|Z) - \ell$. In particular, $H_\infty(X|Y) \geq H_\infty(X) - \ell$.*

We also give the definition of randomness extractors [34], which is somewhat stronger than the average-case strong extractor [15].

Definition 2.11 (Randomness Extractor). *An efficient function $\mathsf{Ext} : \mathcal{X} \times \mathcal{S} \to \mathcal{Y}$ is a (v, ε)-extractor if for all (correlated) random variable X, Z such that the support of X is $\mathcal{X}$ and $H_\infty(X|Z) \geq v$, we have $\Delta((Z, S, \mathsf{Ext}(X; S)), (Z, S, Y)) \leq \varepsilon$, where S (also called the seed) and Y are distributed uniformly and independently over their domains $\mathcal{S}, \mathcal{Y}$ respectively.*

Theorem 2.12 ([15]). *Let $\mathcal{H} = \{h_s : \mathcal{X} \to \mathcal{Y}\}_{s \in \mathcal{S}}$ be a universal family of hash functions meaning that for all $x = x' \in \mathcal{X}$ we have $\mathsf{Pr}_{s \leftarrow \mathcal{S}}[h_s(x) = h_s(x')] \leq \frac{1}{|\mathcal{Y}|}$. Then $\mathsf{Ext}(x, s) \stackrel{def}{=} h_s(x)$, is a (v, ε)-extractor for any parameter $v \geq \log|\mathcal{Y}| + 2\log(1/\varepsilon)$.*

3 Attribute-Based Weak Hash Proof Systems

In this section, we first present a generalization of the weak hash proof system called *attribute-based* weak hash proof system (AB-wHPS). This notion associates attributes and policy functions to the system following the spirit of attribute-based encryption. Next, we show how to construct AB-wHPS from ABE that achieves the property of *succinct keys*, which is the key to leakage resilience with the optimal rate. With a new fine-grained approach, we are able to achieve AB-wHPS with selective security for general circuits, adaptive security of identity comparison functions (i.e., identity-based wHPS), and adaptive security for t-CNF* functions[4], from lattices. This would imply lattice-based leakage resilient, adaptively secure PKE, IBE, ABE for t-CNF*, and selectively secure ABE for general circuits, all with the optimal rate, matching the best known non-leakage resilient selectively/adaptively secure constructions.

[4] We use a "dual" variant of the CNF functions as we discussed in the introduction. The formal definition is presented in Sect. 2.1.

3.1 Formal Definition of Attribute-Based wHPS

We first present the formal definition of an AB-wHPS.

Definition 3.1 (AB-wHPS). *An attribute-based weak hash proof system (*AB-wHPS*) for an attribute space* $\mathcal{X}_\lambda = \{0,1\}^*$ *and a class of functions* $\mathcal{F}_\lambda = \{f : \mathcal{X}_\lambda \to \{0,1\}\}$ *consists of five algorithms* AB-wHPS.$\{$Setup, KeyGen, Encap, Encap*, Decap$\}$*:*

- **Setup.** AB-wHPS.Setup(1^λ) *takes a security parameter* λ *as input, and generates a pair of master public key and master secret key* $(\mathsf{mpk}, \mathsf{msk})$*. The attribute space* $\mathcal{X}_\lambda$ *and the encapsulated key space* $\mathcal{K}$ *are determined by* mpk*.*
- **Key generation.** AB-wHPS.KeyGen (f, msk) *takes as input a function* $f \in \mathcal{F}_\lambda$ *and the master secret key* msk*, and generates a secret key* (f, sk_f)*. Without loss of generality, we think the secret key contains two parts, the function description* f*, and an extra* sk_f*. The secret key is succinct if* $|\mathsf{sk}_f| = o(|f|)$*. When the context is clear, we often omit the description of* f*.*
- **Valid encapsulation.** AB-wHPS.Encap$(\mathsf{mpk}, \boldsymbol{x})$ *takes as input the master public key* mpk *and an attribute* $\boldsymbol{x} \in \mathcal{X}_\lambda$*, and outputs a valid ciphertext* CT *and its corresponding encapsulated key* $k \in \mathcal{K}$*.*
- **Invalid encapsulation.** AB-wHPS.Encap$^*(\mathsf{mpk}, \boldsymbol{x})$ *takes as input the master public key* mpk *and* $\boldsymbol{x} \in \mathcal{X}_\lambda$*, and outputs an invalid ciphertext* CT**.*
- **Decapsulation.** AB-wHPS.Decap$(\mathsf{sk}_f, \mathsf{CT})$ *takes as input a secret key* sk_f *and a ciphertext* CT*, and deterministically outputs* $k \in \mathcal{K}$ *if* $f(\boldsymbol{x}) = 1$ *and* $\perp$ *if* $f(\boldsymbol{x}) = 0$*, where* $\boldsymbol{x}$ *is the corresponding attribute used to generate* CT*.*

Furthermore, an AB-wHPS needs to satisfy three properties: correctness, ciphertext indistinguishability, and universality.

Correctness. For $(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$}$ AB-wHPS.Setup(λ), any $\boldsymbol{x} \in \mathcal{X}_\lambda$ and any $f \in \mathcal{F}_\lambda$ such that $f(\boldsymbol{x}) = 1$, we have

$$\Pr\Big[k = k' \Big| \mathsf{sk}_f \xleftarrow{\$} \text{AB-wHPS.KeyGen}(f, \mathsf{msk}), \\ (\mathsf{CT}, k) \xleftarrow{\$} \text{AB-wHPS.Encap}(\mathsf{mpk}, \boldsymbol{x}), k' = \text{AB-wHPS.Decap}(\mathsf{sk}_f, c)\Big] = 1.$$

Ciphertext Indistinguishability. For any challenge attribute $\boldsymbol{x}^*$, valid/in-valid ciphertexts output by AB-wHPS. Encap$(\mathsf{mpk}, \boldsymbol{x}^*)$ and AB-wHPS.Encap$^*(\mathsf{mpk}, \boldsymbol{x}^*)$ are indistinguishable, even given one secret "1-key" sk_f such that $f(\boldsymbol{x}^*) = 1$ and perhaps many "0-keys" $\mathsf{sk}_{f'}$ such that $f'(\boldsymbol{x}^*) = 0$. More formally, this indistinguishability is always described by the experiment between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ in Table 1.

We define the advantage of $\mathcal{A}$ in the above game to be $\mathbf{Adv}^{\text{AB-wHPS}}_{\Pi,\mathcal{A},\mathcal{F}_\lambda}(\lambda) = |\Pr[\mathcal{A}\ wins] - 1/2|$. The indistinguishability means that $\mathbf{Adv}^{\text{AB-wHPS}}_{\Pi,\mathcal{A},\mathcal{F}_\lambda}(\lambda) \leq \mathsf{negl}(\lambda)$.

Remark 3.2 *In this definition, we require ciphertext indistinguishability to hold even given a single* sk_f *such that* $f(\boldsymbol{x}^*) = 1$. *This suffices to achieve leakage resilient* PKE, IBE, *and* $(\ell, 1)$*-leakage resilient* ABE *directly, and* (ℓ, ω)*-leakage resilient* ABE *for any bounded-polynomial* ω *via a bootstrapping procedure (ref. Sect. 6), where* $\ell \approx (1 - o(1))|\mathsf{sk}_f|$.

Universality. We need one additional information theoretic property, requiring that for any adversary with public parameters, the decapsulation of an invalid ciphertext has information entropy. We define this property in as follow.

Definition 3.3. (Universal AB-wHPS). *We say that an* AB-wHPS *is* $(l, \bar{w})$*-universal, if for any attribute* $\boldsymbol{x} \in \mathcal{X}_\lambda$, $(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{AB\text{-}wHPS.Setup}(1^\lambda)$, *and* $\mathsf{CT}^* \xleftarrow{\$} \mathsf{AB\text{-}wHPS.Encap}^*(\mathsf{mpk}, \boldsymbol{x})$, *it holds*

$$H_\infty(\mathsf{AB\text{-}wHPS.Decap}(\mathsf{CT}^*, \mathsf{sk}_f)|\mathsf{mpk}, \mathsf{msk}, \mathsf{CT}^*, \boldsymbol{x}) \geq \bar{w},$$

where $\mathsf{sk}_f = \mathsf{AB\text{-}wHPS.KeyGen}(f, \mathsf{msk})$ *with* $f(\boldsymbol{x}) = 1$, *and* l *is the bit-length of the decapsulated value from* $\mathsf{AB\text{-}wHPS.Decap}(\mathsf{CT}^*, \mathsf{sk})$.

Table 1. X

Valid/Invalid Ciphertext Indistinguishability Experiment
Attribute Challenge: In the setting of selective case, $\mathcal{A}$ chooses an challenge attribute $\boldsymbol{x}^* \in \mathcal{X}_\lambda$ before the Setup stage and sends it to $\mathcal{C}$; In the setting of adaptive case, $\mathcal{A}$ chooses a challenge $\boldsymbol{x}^* \in \mathcal{X}_\lambda$ in any arbitrary stage before the challenge stage, and sends it to $\mathcal{C}$.
Setup: The challenger $\mathcal{C}$ gets a pair of $(\mathsf{mpk}, \mathsf{msk})$ by running $\mathsf{AB\text{-}wHPS.Setup}(1^\lambda)$, and sends mpk to $\mathcal{A}$.
Test Stage 1: $\mathcal{A}$ adaptively queries the challenger $\mathcal{C}$ with $f \in \mathcal{F}_\lambda$, and $\mathcal{C}$ responds with (f, sk_f).
Challenge Stage: $\mathcal{C}$ selects $b \xleftarrow{\$} \{0, 1\}$. If $b = 0$, $\mathcal{C}$ computes $(\mathsf{CT}, k) \xleftarrow{\$} \mathsf{AB\text{-}wHPS.Encap}(\mathsf{mpk}, \boldsymbol{x}^*)$. If $b = 1$, $\mathcal{C}$ computes $\mathsf{CT} \xleftarrow{\$} \mathsf{AB\text{-}wHPS.Encap}^*(\mathsf{mpk}, \boldsymbol{x}^*)$. Then $\mathcal{C}$ returns CT to $\mathcal{A}$.
Test Stage 2: $\mathcal{A}$ adaptively queries the challenger $\mathcal{C}$ with $f \in \mathcal{F}$. Then $\mathcal{C}$ responds with (f, sk_f).
Output: $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$. $\mathcal{A}$ wins the experiment, if $b = b'$ and at most one of $\mathcal{A}$'s key queries f satisfies $f(\boldsymbol{x}^*) = 1$.

3.2 Fine-Grained Security Notions and General Construction of AB-wHPSfrom ABE

In this section, we present how to construct AB-wHPS from ABE. To achieve adaptive security for several subclasses of policy functions, we present a more fine-grained approach as follows. We first define a notion called partially selective/adaptive security over partitioned attributes. Next we show for a *specific class* $\mathcal{G}$, if an ABE is $(\mathsf{X}, \mathsf{sel})$-secure for class $\mathcal{F} \wedge_{\|} \mathcal{G}$ for $\mathsf{X} \in \{\mathsf{sel}, \mathsf{ada}\}$, then we can construct an X-secure AB-wHPS for $\mathcal{F}$. Moreover, suppose the underlying ABE has succinct keys, so does the AB-wHPS. In the next section, we show instantiations of $(\mathsf{ada}, \mathsf{sel})$-secure ABE for various function classes. Below we elaborate on the notations and the new security definition.

Definition 3.4. *Let* $\mathcal{F}_1 = \{f_1 : \mathcal{X}_1 \to \{0,1\}\}$ *and* $\mathcal{F}_2 = \{f_2 : \mathcal{X}_2 \to \{0,1\}\}$ *be two function classes. We define the operator* $\wedge_{\|}$ *over two function classes as follow:* $\mathcal{F} := \mathcal{F}_1 \wedge_{\|} \mathcal{F}_2$ *is a function class that consists of function maps* $\mathcal{X}_1 \times \mathcal{X}_2 \to \{0,1\}$*, where each function* $f_{f_1,f_2} \in \mathcal{F}$ *is indexed by two functions* $f_1 \in \mathcal{F}_1$ *and* $f_2 \in \mathcal{F}_2$ *such that on input* $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathcal{X}_1 \times \mathcal{X}_2$*,* $f_{f_1,f_2}(\boldsymbol{x}) = f_1(\boldsymbol{x}_1) \wedge f_2(\boldsymbol{x}_2)$*.*

Using this composed function class in Definition 3.4, we can naturally consider any combination of selective/adaptive security for ABE as follows.

Definition 3.5. (Partial Selective/Adaptive Security). *For any* ABE *with the attribute space* $\mathcal{X}_1 \times \mathcal{X}_2$ *for the policy function class* $\mathcal{F} := \mathcal{F}_1 \wedge_{\|} \mathcal{F}_2$ *defined as in Definition 3.4, we define partial selective/adaptive security as follows:*

- ada-sel *security: For any challenge attribute* $\boldsymbol{x}^* = (\boldsymbol{x}_1^*, \boldsymbol{x}_2^*) \in \mathcal{X}_1 \times \mathcal{X}_2$*,* $\boldsymbol{x}_1^*$ *is chosen adaptively but* $\boldsymbol{x}_2^*$ *is chosen selectively in the corresponding indistinguishability experiment.*
- sel-ada *security: For any challenge attribute* $\boldsymbol{x}^* = (\boldsymbol{x}_1^*, \boldsymbol{x}_2^*) \in \mathcal{X}_1 \times \mathcal{X}_2$*,* $\boldsymbol{x}_1^*$ *is chosen selectively and* $\boldsymbol{x}_2^*$ *is chosen adaptively in the corresponding indistinguishability experiment.*

This notion also captures the standard selective (or adaptive) security as sel-sel *(or* ada-ada*) security, where both parts of the challenge attribute are chosen selectively (or adaptively).*

Remark 3.6. *In this work, we need a slightly weaker version of the partial selective/adaptive security from* ABE *– the adversary is only allowed to query one key* (f, g) *such that* $f(x_1^*) = 1$ *and* $g(x_2^*) = 0$*. The other keys are of the form* (f', g') *such that* $f'(x_1^*) = 0$*. Therefore, throughout this work we will use this slightly weaker version by default.*

Remark 3.7. *In the same way, we can define the partial selective/adaptive ciphertext indistinguishability for* AB-wHPS*.*

Remark 3.8. *This definition can be defined recursively. For example, the first part* $\mathcal{F}_1$ *can also consists of two parts, i.e.,* $\mathcal{F}_1 = \mathcal{F}_{1,1} \wedge_{\|} \mathcal{F}_{1,2}$*. In this case, we can consider* (X-Y)-Z *security for any combination of* $\mathsf{X}, \mathsf{Y}, \mathsf{Z} \in \{\mathsf{sel}, \mathsf{ada}\}$*.*

To construct our desired AB-wHPS for $\mathcal{F}$, we need an ABE for $\mathcal{F} \wedge_{\|} \mathcal{G}$ for this specific $\mathcal{G}$ as we describe below.

Definition 3.9. *Let $m = m(\lambda)$ and $n = n(\lambda)$ be two integer parameters, and we define a function class $\mathcal{G} = \{g : [n] \times [m] \to \{0,1\}\}$ as follows. Each function $g_{\boldsymbol{y}} \in \mathcal{G}$ is indexed by a vector $\boldsymbol{y} = (y_1, \ldots, y_n)^\top \in [m]^n$, and $g_{\boldsymbol{y}}(x_1, x_2) = 1$ if and only if $x_2 = y_{x_1}$.*

Remark 3.10. *The class $\mathcal{G}$ can be captured by boolean circuits with input length $\log n + \log m$, and depth within $O(\log(n+m))$, i.e., $\bigvee_{i \in [n]} (i \stackrel{?}{=} x_1) \wedge (y_i \stackrel{?}{=} x_2)$.*

Given this particular class $\mathcal{G}$ (with parameters m, n) defined in Definition 3.9 and a class $\mathcal{F}$, we show how to use ABE for $\mathcal{F} \wedge_{\|} \mathcal{G}$ to construct AB-wHPS for $\mathcal{F}$. For different classes $\mathcal{F}$'s, the AB-wHPS can be used to further derive leakage resilient PKE, IBE, and ABE.

Construction 3.11 (AB-wHPS from ABE). *Let $\Pi_{\mathsf{ABE}} = \mathsf{ABE}.\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$ be an ABE scheme with attribute-space $\bar{\mathcal{X}}_\lambda = \mathcal{X}_\lambda \times \mathcal{X}'_\lambda = \{0,1\}^* \times \{[n] \times [m]\}$, message-space $\mathcal{M} = \mathbb{Z}_m$ and ciphertext space $\mathcal{CT}$ for the policy-function class $\mathcal{F} \wedge_{\|} \mathcal{G}$ for the class $\mathcal{G}$ as in Definition 3.9 with parameters m, n. Then, an AB-wHPS $\Pi_{\mathsf{AB\text{-}wHPS}}$ with attribute space $\mathcal{X}_\lambda = \{0,1\}^*$ and the encapsulated-key-space $\mathcal{K} = \mathbb{Z}_m^n$ for the policy-function class $\mathcal{F} = \{f : \{0,1\}^* \to \{0,1\}\}$ can be constructed as follows:*

- $\mathsf{AB\text{-}wHPS.Setup}(1^\lambda)$*: Given the security parameter λ as input, the algorithm runs* $\mathsf{ABE.Setup}$ *to generate* $(\mathsf{mpk}^{\mathsf{ABE}}, \mathsf{msk}^{\mathsf{ABE}}) \stackrel{\$}{\leftarrow} \mathsf{ABE.Setup}(1^\lambda)$*, and outputs* $\mathsf{mpk} := \mathsf{mpk}^{\mathsf{ABE}}$ *and* $\mathsf{msk} := \mathsf{msk}^{\mathsf{ABE}}$*.*
- $\mathsf{AB\text{-}wHPS.KeyGen}(\mathsf{msk}, f)$*: Given a master secret-key* $\mathsf{msk} := \mathsf{msk}^{\mathsf{ABE}}$ *and a function $f \in \mathcal{F}$ as input, the algorithm first chooses a random vector $\boldsymbol{y} \stackrel{\$}{\leftarrow} [m]^n$, and sets $\hat{f} := \hat{f}_{f, g_{\boldsymbol{y}}} \in \mathcal{F} \wedge_{\|} \mathcal{G}$. Then the algorithm runs* $\mathsf{ABE.KeyGen}$ *to generate* $\mathsf{sk}_{\hat{f}}^{\mathsf{ABE}} \stackrel{\$}{\leftarrow} \mathsf{ABE.KeyGen}(\mathsf{msk}^{\mathsf{ABE}}, \hat{f})$*, and outputs* $\mathsf{sk}_f := (\hat{f}, \mathsf{sk}_{\hat{f}}^{\mathsf{ABE}})$ *as the secret key for f. Note that the description of $\hat{f}$ can be expressed as $(f, \boldsymbol{y})$*
- $\mathsf{AB\text{-}wHPS.Encap}(\mathsf{mpk}, \boldsymbol{x})$*: Given a master public-key* mpk *and an attribute $\boldsymbol{x} \in \{0,1\}^*$ as input, the algorithm first samples a random vector $\boldsymbol{k} = (k_1, \ldots, k_n)^\top \in \mathbb{Z}_m^n$, and then runs* $\mathsf{ABE.Enc}$ *mn times with attributes $\boldsymbol{x}_{i,j} = (\boldsymbol{x}, i, j) \in \{0,1\}^* \times [n] \times [m]$ to set*

$$\mathsf{CT} := \{\mathsf{ct}_{i,j} \stackrel{\$}{\leftarrow} \mathsf{ABE.Enc}(\mathsf{mpk}, \boldsymbol{x}_{i,j}, k_i)\}_{(i,j) \in [n] \times [m]} \in \mathcal{CT}^{n \times m}, \text{ i.e.,}$$

$$\mathsf{CT} := \begin{bmatrix} \mathsf{ABE.Enc}(\boldsymbol{x}_{1,1}, k_1) & \ldots & \mathsf{ABE.Enc}(\boldsymbol{x}_{1,j}, k_1) & \ldots & \mathsf{ABE.Enc}(\boldsymbol{x}_{1,m}, k_1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathsf{ABE.Enc}(\boldsymbol{x}_{n,1}, k_n) & \ldots & \mathsf{ABE.Enc}(\boldsymbol{x}_{n,j}, k_n) & \ldots & \mathsf{ABE.Enc}(\boldsymbol{x}_{n,m}, k_n) \end{bmatrix}.$$

Finally, the algorithm outputs $(\mathsf{CT}, \boldsymbol{k})$.

– $\mathsf{AB\text{-}wHPS.Encap}^*(\mathsf{mpk}, \boldsymbol{x})$*: Given a master public-key* mpk *and an attribute* $\boldsymbol{x} \in \{0,1\}^*$ *as input, the algorithm first samples a random vector* $\boldsymbol{k} = (k_1, \ldots, k_n)^\top \in \mathbb{Z}_m^n$*, and then runs* $\mathsf{ABE.Enc}$ *mn times with attributes* $\boldsymbol{x}_{i,j} = (\boldsymbol{x}, i, j)$ *to set*

$$\mathsf{CT}^* := \{\mathsf{ct}^*_{i,j} \xleftarrow{\$} \mathsf{ABE.Enc}(\mathsf{mpk}, \boldsymbol{x}_{i,j}, k_i + j)\}_{(i,j)\in[n]\times[m]} \in \mathcal{CT}^{n\times m}, \text{ i.e.,}$$

$$\mathsf{CT}^* := \begin{bmatrix} \mathsf{ABE.Enc}(\boldsymbol{x}_{1,1}, k_1{+}1) & \ldots & \mathsf{ABE.Enc}(\boldsymbol{x}_{1,j}, k_1{+}j) & \ldots & \mathsf{ABE.Enc}(\boldsymbol{x}_{1,m}, k_1{+}m) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathsf{ABE.Enc}(\boldsymbol{x}_{n,1}, k_n{+}1) & \ldots & \mathsf{ABE.Enc}(\boldsymbol{x}_{n,j}, k_n{+}j) & \ldots & \mathsf{ABE.Enc}(\boldsymbol{x}_{n,m}, k_n{+}m) \end{bmatrix},$$

where the addition $k_i + j$ *is performed over* $\mathbb{Z}_m$*. The algorithm outputs* CT^**.*

– $\mathsf{AB\text{-}wHPS.Decap}(\mathsf{sk}_f, \mathsf{CT})$*: Given a secret key* $\mathsf{sk}_f := (\boldsymbol{y}, \mathsf{sk}^{\mathsf{ABE}}_{\hat{f}})$ *and* $\mathsf{CT} := \{\mathsf{ct}_{i,j}\}_{(i,j)\in[n]\times[m]}$ *as input, the algorithm runs* $\mathsf{ABE.Dec}$ *to compute* $k_i = \mathsf{ABE.Dec}(\mathsf{sk}^{\mathsf{ABE}}_{\hat{f}}, \mathsf{ct}_{i,y_i})$ *for all* $i \in [n]$*, and then outputs* $\boldsymbol{k} = (k_1, \ldots, k_n)^\top$*, if* $\hat{f}(\boldsymbol{x}, i, y_i) = f(\boldsymbol{x}) \wedge g_{\boldsymbol{y}}(i, y_i) = 1$ *for all* $i \in [n]$*, and* $\perp$ *otherwise.*

Intuitively, our attribute design (the class $\mathcal{G}$) allows the secret key to open one ciphertext per row while keeps the others secret. For the valid encapsulation, all ciphertexts in a row encrypts the same element, while for the invalid encapsulation, they encrypt different elements. As the secret key can only open one per row, an adversary cannot distinguish a valid from an invalid encapsulation, even given the secret key.

Our $\mathsf{AB\text{-}wHPS}$ secret key would be of length $|\hat{f}_{f,g_y}| + s(\hat{f}_{f,g_y}) = |\boldsymbol{y}| + |f| + s(\hat{f}_{f,g_y}) = n \log m + |f| + s(\hat{f}_{f,g_y})$, where $s(\cdot)$ is the key-size function (of the extra part, excluding the function description) of the underlying ABE. If the underlying ABE has succinct keys, i.e., $s(f) = o(|f|)$, then our $\mathsf{AB\text{-}wHPS}$ secret would have size $n \log m + |f| + s(\hat{f}_{f,g_y}) = n \log m + |f| + o(n \log m + |f|)$. By setting sufficiently large n, m, we can achieve ABE with the optimal leakage rate, ref. Sect. 4.

Next we present the following theorem. Due to space limit, we defer the full proof to the full version, due to space limit.

Theorem 3.12. ($\mathsf{AB\text{-}wHPS}$ from ABE). *Suppose* Π_{ABE} *is a secure* ABE *scheme with attribute space* $\bar{\mathcal{X}}_\lambda = \mathcal{X}_\lambda \times \mathcal{X}'_\lambda = \{0,1\}^* \times \{[n] \times [m]\}$ *for the function class* $\mathcal{F} \wedge_{\|} \mathcal{G}$*, where* $\mathcal{G}$ *is the class as in Definition 3.9 with parameters* m, n*, then the construction* $\Pi_{\mathsf{AB\text{-}wHPS}}$ *described above is an* $(n \log m, n \log m)$*-universal* $\mathsf{AB\text{-}wHPS}$ *with the attribute space* $\mathcal{X}_\lambda$ *and the encapsulated-key-space* $\mathcal{K} = \mathbb{Z}_m^n$*, for the function class* $\mathcal{F}$*. Furthermore,*

– *if the* ABE *is* $\mathsf{X\text{-}sel}$ *secure for* $\mathsf{X} \in \{\mathsf{sel}, \mathsf{ada}\}$*, then the* $\mathsf{AB\text{-}wHPS}$ *is* X *secure;*
– *if the key-size (of the extra part, excluding the function description) of the* ABE *scheme for policy function* f *is* $s(f)$*, then the key size of the* $\mathsf{AB\text{-}wHPS}$ *for* f *is* $n \log m + |f| + s(\hat{f}_{f,g_y})$*, where* $s(\cdot)$ *is the key-size function (of the extra part, excluding the function description) of the underlying* ABE*.*

3.3 Instantiations of AB-wHPS from Lattices

Now we show how to instantiate the required underlying ABE. By combining the work [7] with [2] or [38], we get ABE for the following three classes.

Theorem 3.13. *Assuming* LWE*, then there exist:*

1. ada-sel*-secure* ABE *for* $\mathcal{I} \wedge_{\|} \mathcal{G}$*, where* $\mathcal{I}$ *is the comparison function (*IBE*).*
2. ada-sel*-secure* ABE *for* t-CNF* $\wedge_{\|} \mathcal{G}$*, where* t-CNF* *is the dual of the* t *conjunctive normal form formula. (Ref. Sect. 2.1.)*
3. sel-sel *secure* ABE *for* $\mathcal{F} \wedge_{\|} \mathcal{G}$*, where* $\mathcal{F}$ *is the general boolean circuits.*

In all three cases, the size of the secret keys (excluding the function description) depends only on the depth of the circuit but not the size.

We present the constructions in full version for completeness. As a direct corollary of this theorem, we obtain the following AB-wHPS from lattices.

Corollary 3.14. *Assuming* LWE*, there exists* AB-wHPS *that is*

1. *adaptively secure for the comparison functions;*
2. *adaptively secure for* t-CNF* *functions.*
3. *selectively secure for general circuits.*

Moreover, the secret key size (excluding the function description) of the AB-wHPS *only depends on the depth of the function, but not the size.*

4 Optimal-Rate Leakage-Resilient Encryption Schemes in the Relative Leakage Model

Prior work (e.g., Naor and Segev [33], Alwen et al. [5], and Hazay et al. [25]) showed how to construct leakage resilient PKE/IBE from wHPS/IB-wHPS in the relative model. The construction can be generalized to construct leakage resilient ABE from AB-wHPS in the same spirit. To further achieve the optimal leakage rate, we observe that all we need is an AB-wHPS with succinct keys (which do not depend on the function size). This is what we construct in Sect. 3.2, i.e., Construction 3.11, Theorem 3.12, AB-wHPS and the underlying ABE instantiations in Corollary 3.14.

Construction 4.1. *Let* $\Pi =$AB-wHPS.$\{$Setup, KeyGen, Encap, Encap*, Decap$\}$ *be a* $(\log|\mathcal{K}|, \log|\mathcal{K}|)$*-universal* AB-wHPS *with the encapsulated-key-space* $\mathcal{K}$ *and attribute space* $\mathcal{X} = \{0,1\}^*$ *for a class of policy functions* $\mathcal{F} = \{f : \{0,1\}^* \to \{0,1\}\}$*. Let* Ext $: \mathcal{K} \times \mathcal{S} \to \mathcal{M}$ *be a* $(\log|\mathcal{K}| - \ell, \varepsilon)$*-extractor, where three sets* $\mathcal{K}, \mathcal{S}, \mathcal{M}$ *are efficient ensembles,* $\ell = \ell(\lambda)$ *is some parameter and* $\varepsilon = \varepsilon(\lambda) =$ negl(λ) *is negligible. Furthermore, assume that* $\mathcal{M}$ *is an additive group. Then, a leakage-resilient* ABE *scheme* $\Pi_{\mathcal{F}} = \Pi_{\mathcal{F}}.\{$Setup, KeyGen, Enc, Dec$\}$ *with message space* $\mathcal{M}$ *and policy function class* $\mathcal{F}$ *can be constructed as follows:*

- $\Pi_{\mathcal{F}}.\mathsf{Setup}(1^\lambda)$: *The algorithm runs* $(\mathsf{mpk}^{\Pi}, \mathsf{msk}^{\Pi}) \xleftarrow{\$} \Pi.\mathsf{Setup}(1^\lambda)$, *and outputs* $\mathsf{mpk} := \mathsf{mpk}^{\Pi}$, *and* $\mathsf{msk} := \mathsf{msk}^{\Pi}$.
- $\Pi_{\mathcal{F}}.\mathsf{KeyGen}(\mathsf{msk}, f)$: *Given a master secret-key* msk *and a function* $f \in \mathcal{F}$ *as input, the algorithm runs* AB-wHPS.KeyGen *to generate and output* (f, sk_f^{Π}), *where* $\mathsf{sk}_f := \mathsf{sk}_f^{\Pi} \xleftarrow{\$} \mathsf{AB\text{-}wHPS.KeyGen}(\mathsf{msk}, f)$.
- $\Pi_{\mathcal{F}}.\mathsf{Enc}(\mathsf{mpk}, \boldsymbol{x}, \mu)$: *Given a master public-key* mpk, *an attribute* $\boldsymbol{x} \in \mathcal{X} = \{0,1\}^*$, *and a message* $\mu \in \mathcal{M}$ *as input, the algorithm runs* AB-wHPS.Encap *to generate* $(\mathsf{CT}', k) \leftarrow \mathsf{AB\text{-}wHPS.Encap}(\mathsf{mpk}, \boldsymbol{x})$, *and then samples* $s \xleftarrow{\$} \mathcal{S}$. *Furthermore, the algorithm computes and outputs*

$$\mathsf{ct} = (s, \mathsf{ct}_0, \mathsf{ct}_1) = (s, \mathsf{CT}', \mu + \mathsf{Ext}(k, s)).$$

- $\Pi_{\mathcal{F}}.\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$: *Given a ciphertext* $\mathsf{ct} = (s, \mathsf{ct}_0, \mathsf{ct}_1)$ *and a secret key* sk_f *as input, the algorithm runs* AB-wHPS.Decap *to generate* $k = \mathsf{AB\text{-}wHPS.Decap}(\mathsf{sk}_f, \mathsf{ct}_0)$, *and then output* $\mu = \mathsf{ct}_1 - \mathsf{Ext}(k, s)$.

Our construction achieves a leakage resilient ABE, and can be re-calibrated into a leakage resilient PKE/IBE. We summarize the results in the following theorem, and defer the full proof to the full version, due to space limit.

Theorem 4.2. *Assume* Π *is a selectively (or adaptively, resp.) secure* $(\log|\mathcal{K}|, \log|\mathcal{K}|)$*-universal* AB-wHPS *for the policy function class* $\mathcal{F}$, *and* $\mathsf{Ext} : \mathcal{K} \times \mathcal{S} \to \mathcal{M}$ *be a* $(\log|\mathcal{K}| - \ell, \mathsf{negl}(\lambda))$*-extractor. Then the above* ABE *scheme* $\Pi_{\mathcal{F}} = \Pi_{\mathcal{F}}.\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$ *for* $\mathcal{F}$ *is a selectively (or adaptively, resp.)* $\ell(\lambda)$*-leakage resilient attribute-based encryption scheme for the policy function class* $\mathcal{F}$ *in the relative-leakage model. Particularly,* $\Pi_{\mathcal{F}}$ *is aslo*

- *an* $\ell(\lambda)$*-leakage-resilient* PKE *in the relative-leakage model, if* $\mathcal{F}$ *contains only a single function that always outputs* 1.
- *an* $\ell(\lambda)$*-leakage-resilient* IBE *in the relative-leakage model, if* $\mathcal{F}$ *contains the following comparison functions, i.e., each function* $f_{\boldsymbol{y}} \in \mathcal{F}$ *is indexed by a vector* $\boldsymbol{y}$, *and* $f_{\boldsymbol{y}}(\boldsymbol{x}) = 1$ *if and only if* $\boldsymbol{y} = \boldsymbol{x}$.

Combining Theorem 3.12 and Theorem 4.2, we obtain the following results. Assume there exists a sel-sel (or ada-sel) secure ABE scheme with the message space $\mathbb{Z}_m$ for the function class $\mathcal{F} \wedge_{\|} \mathcal{G}$, where $\mathcal{G}$ is the class as in Definition 3.9 with parameters m, n, and the key-length (of the extra part, excluding the function description of f) of this underlying ABE scheme for policy function f is $s(f)$. Then the allowed leakage length of the above leakage resilient ABE (or IBE or PKE) scheme $\Pi_{\mathcal{F}}$ for the function class $\mathcal{F}$ is $\ell = (n \log m - 2\lambda)$ and the key-length of $\Pi_{\mathcal{F}}$ for f is $|\mathsf{sk}_f| = n \log m + |f| + s(\hat{f}_{f,g_y})$.

Furthermore, if the secret key size $s(\hat{f}_{f,g_y})$ is succinct, i.e., $s(\hat{f}_{f,g_y}) = o(|\hat{f}_{f,g_y}|) = o(n \log m + |f|)$, then we can set sufficiently large n, m such that $n \log m = \omega(|f|)$. Consequently, the leakage rate of this scheme $\Pi_{\mathcal{F}}$ is $\frac{n \log m - 2\lambda}{n \log m + |f| + s(\hat{f}_{f,g_y})} = \frac{1 - \frac{2\lambda}{n \log m}}{1 + \frac{s(\hat{f}_{f,g_y}) + |f|}{n \log m}} \approx 1 - o(1)$, achieving the desired optimal leakage rate.

Finally, by combining Corollary 3.14 and Theorem 4.2, we obtain the following Corollary.

Corollary 4.3. *Assuming* LWE*, for all polynomial* $S = \mathsf{poly}(\lambda)$*, there exist* $1 - o(1)$ *leakage resilient* ABE *schemes in the relative leakage model, which are*

1. *adaptively secure for the comparison functions;*
2. *adaptively secure for* t-CNF^* *functions of size up to* S*;*
3. *selectively secure for general circuits of size up to* S*.*

Remark 4.4. *We note that our ABE schemes are leakage resilient even if the policy function goes beyond the size bound* S*. The leakage rate would still be* $1 - o(1)$ *for a slightly restricted class that leaks* $n \log m - 2\lambda$ *on the part* $\boldsymbol{y}$*, the whole description of* f*, and the extra part of* sk_f^{Π} *(excluding the function description) of the underlying* AB-wHPS*. This is more restrictive than functions that leak* $n \log m - 2\lambda + |f|$ *from the whole secret key.*

5 Extension I: Optimal-Rate Leakage-Resilient Encryption Schemes in the BRM

In this section, we present how to use AB-wHPS to construct optimal-rate leakage resilient ABE in the BRM. We follow the structure of [5,25] by first amplifying the hash proof system and then combining it with a locally computable extractor [40]. In particular, we first amplify AB-wHPS through parallel repetition and random sampling in Sect. 5.1. Then, in Sect. 5.2, we generalize the notion of locally computable extractor by Vadhan [40] into one with larger alphabets, and show that a refined analysis of this tool can be used to derive $1 - o(1)$ leakage rate in the BRM, improving the prior analysis [5,35] that can only achieve a constant leakage rate. Finally in Sect. 5.3, we present the overall construction of our leakage resilient ABE in the BRM with the optimal leakage rate.

5.1 Amplification of AB-wHPS

Definition 5.1. *Let* n' *be a positive integer, and* $\mathcal{H} = \{h : [n'] \to \{0,1\}\}$ *be a function class where each function* $h_y \in \mathcal{H}$ *is indexed by a value* $y \in [n']$*, and* $h_y(x) = 1$ *if and only if* $x = y$*.*

Construction 52. (Construction of Amplified AB-wHPS.). *Let* $\Pi = \mathsf{AB\text{-}wHPS}.\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Encap}^*, \mathsf{Decap}\}$ *be an* AB-wHPS *with the encapsulated-key-space* $\mathcal{K}$ *and attribute space* $\mathcal{X} = \{0,1\}^* \times [n']$ *for a class of functions* $\mathcal{F} \wedge_{\|} \mathcal{H}$*, and let* $t \leq n'$ *be a positive integer. Then a new* AB-wHPS $\Pi_{\|}^{n',t}$ *with attribute space* $\{0,1\}^*$ *and the encapsulated-key-space* $\mathcal{K}^t$ *for the function class* $\mathcal{F}$ *can be constructed.*

- $\Pi_{\|}^{n',t}.\mathsf{Setup}(1^\lambda)$*: The algorithm runs* $(\mathsf{mpk}^{\Pi}, \mathsf{msk}^{\Pi}) \xleftarrow{\$} \Pi.\mathsf{Setup}(1^\lambda)$*, and outputs* $\mathsf{mpk} := \mathsf{mpk}^{\Pi}$*, and* $\mathsf{msk} := \mathsf{msk}^{\Pi}$*.*

– $\Pi_{\|}^{n',t}.\mathsf{KeyGen}(\mathsf{msk}, f)$: *Given a function* $f \in \mathcal{F}$, *the algorithm first sets* $\hat{f}^i = \hat{f}^i_{f,h_i} \in \mathcal{F} \wedge_{\|} \mathcal{H}$ *for every* $i \in [n']$, *and runs* AB-wHPS.KeyGen n' *times to generate* $\mathsf{sk}_{\hat{f}^i} \xleftarrow{\$} \Pi.\mathsf{KeyGen}(\mathsf{msk}^{\Pi}, \hat{f}^i)$ *for* $i \in [n']$. *The algorithm outputs*

$$\mathsf{sk}_f := \left(\mathsf{sk}_{\hat{f}^1},\ \mathsf{sk}_{\hat{f}^2}, \ldots,\ \mathsf{sk}_{\hat{f}^{n'}}\right).$$

– $\Pi_{\|}^{n',t}.\mathsf{Encap}(\mathsf{mpk}, \boldsymbol{x})$: *Given* mpk *and an attribute* $\boldsymbol{x} \in \{0,1\}^*$ *as input, the algorithm chooses a random subset* $\boldsymbol{r} := \{r_1, \ldots, r_t\} \subseteq [n']$ *and computes*

$$(\mathsf{CT}_i, k_i) \xleftarrow{\$} \Pi.\mathsf{Encap}(\mathsf{mpk}, (\boldsymbol{x}, r_i)) \text{ for all } i \in [t].$$

The algorithm finally outputs $\mathsf{CT} := (\boldsymbol{r}, \mathsf{CT}_1, \ldots, \mathsf{CT}_t)$ *and* $\boldsymbol{k} = (k_1, \ldots, k_t)^\top$.

– $\Pi_{\|}^{n',t}.\mathsf{Encap}^*(\mathsf{mpk}, \boldsymbol{x})$: *Given* mpk *and an attribute* $\boldsymbol{x} \in \{0,1\}^*$ *as input, the algorithm chooses a random subset* $\boldsymbol{r} := \{r_1, \ldots, r_t\} \subseteq [n']$ *and computes*

$$\mathsf{CT}_i \xleftarrow{\$} \Pi.\mathsf{Encap}^*(\mathsf{mpk}, (\boldsymbol{x}, r_i)) \text{ for all } i \in [t].$$

Finally, the algorithm outputs $\mathsf{CT} := (\boldsymbol{r}, \mathsf{CT}_1, \ldots, \mathsf{CT}_t)$.

– $\Pi_{\|}^{n',t}.\mathsf{Decap}(\mathsf{sk}_f, \mathsf{CT})$: *Given a ciphertext* $\mathsf{CT} := (\boldsymbol{r}, \mathsf{CT}_1, \ldots, \mathsf{CT}_t)$ *and a secret key* $\mathsf{sk}_f := \left(\mathsf{sk}_{\hat{f}^1},\ \mathsf{sk}_{\hat{f}^2}, \ldots,\ \mathsf{sk}_{\hat{f}^{n'}}\right)$, *the algorithm runs* $\Pi.\mathsf{Decap}$ *to generate* $k_i = \Pi.\mathsf{Decap}(\mathsf{sk}_{\hat{f}^{r_i}}, \mathsf{CT}_i)$ *for* $i \in [t]$, *and outputs* $\boldsymbol{k} = (k_1, \ldots, k_t)^\top$ *if* $\hat{f}^{r_i}(\boldsymbol{x}, r_i) = 1$ *for all* $i \in [t]$. *Otherwise, the algorithm outputs* $\perp$.

Next, we present the following amplification theorem, which is essential an extension of the work [5]. Due to space limit, we defer the full proof to the full version of this paper.

Theorem 5.3. *Assume* Π *is an* (l, w)*-universal* AB-wHPS *with the encapsulated-key-space* $\mathcal{K}$ *for* $\mathcal{F} \wedge_{\|} \mathcal{H}$. *Then the above amplified construction of* $\Pi_{\|}^{n',t}$ *is an* $(t \cdot l, t \cdot w)$*-universal* AB-wHPS *with the encapsulated-key-set* $\mathcal{K}^t$ *for* $\mathcal{F}$. *Furthermore,*

– *if the underlying* Π *is selectively (or adaptively) secure, then the* $\Pi_{\|}^{n',t}$ *is also selectively (or adaptively) secure;*
– *if the secret-key-size of* Π *scheme for the policy function* f *is* $(|f| + s(f))$,[5] *then the secret-key size of the* $\Pi_{\|}^{n',t}$ *for* f *is* $n' \times (|f| + \log n' + s(\hat{f}_{f,h}))$.

Combining Theorem 3.12 and Theorem 5.3, we obtain the following corollary.

Corollary 5.4. *Assume there exists an* ABE *scheme with the message space* $\mathbb{Z}_m$ *for the function class* $\mathcal{F} \wedge_{\|} \mathcal{H} \wedge_{\|} \mathcal{G}$, *where* $\mathcal{G}$ *with parameters* m, n *and* $\mathcal{H}$ *with parameter* n' *are as Definitions 3.9 and 5.1, then there exists an amplified* AB-wHPS *with the encapsulated-key-space* $\mathbb{Z}_m^t$ *for the function class* $\mathcal{F}$.

[5] Recall that the function $s(f)$ denotes the size of the extra part of the secret key, excluding the description of the function.

5.2 Locally Computable Extractor

Definition 5.5 (Locally Computable Extractor, Definition 6 in [40]). *An extractor* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^v$ *is said to be* t*-locally computable if for every* $r \in \{0,1\}^d$, $\mathsf{Ext}(\boldsymbol{x}, r)$ *depends only on* t*-bits of* $\boldsymbol{x} \in \{0,1\}^n$.

For our application (constructing leakage-resilient encryption in the BRM), we need a generalized variant of the above notion. Let $\boldsymbol{x} \in \{0,1\}^{nk}$ be a vector. We can view it as a concatenation of n vectors $\boldsymbol{x}_i \in \{0,1\}^k$ for $i \in [n]$, i.e., $\boldsymbol{x} = (\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_n^\top)^\top$. In this case, each $\boldsymbol{x}_i \in \{0,1\}^k$ can be viewed as a symbol of some larger alphabet, i.e., $\Gamma = \{0,1\}^k$, and we will need a locally computable extractor for Γ as follow.

Definition 5.6 (Locally Computable Extractor for Larger Alphabets). *Let* $\Gamma = \{0,1\}^k$ *be some alphabet. An extractor* $\mathsf{Ext} : \Gamma^n \times \{0,1\}^d \to \{0,1\}^v$ *is* t*-locally computable with respect to* Γ *if for every* $\boldsymbol{r} \in \{0,1\}^d$, $\mathsf{Ext}(\boldsymbol{x}, \boldsymbol{r})$ *depends only on* t *symbols of* $\boldsymbol{x} = (\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_n^\top)^\top \in \Gamma^n$.

Generally, a locally computable extractor can be obtained in two steps [40]: (1) the extractor uses part of the seed to select t bits (or symbols) of $\boldsymbol{x}$, and (2) the remaining seed is used to apply a standard extractor on the selected bits/symbols in the previous step. Vadhan [40] showed that as long as the selection in step (1) achieves an average sampler, then the combined steps would achieve a locally computable extractor. We summarize the result of Vadhan [40] below. We first recall the notion of an average sampler.

Definition 5.7 (Average Sampler, Definition 8 in [40]). *A function* $\mathsf{Samp} : \{0,1\}^r \to [n]^t$ *is a* (μ, θ, γ) *average sampler if for every function* $f : [n] \to [0,1]$ *with average value* $\frac{1}{n}\sum_i f(i) \geq \mu$,

$$\Pr_{(i_1,\ldots,i_t) \xleftarrow{\$} \mathsf{Samp}(U_r)} \left[\frac{1}{t} \sum_{j=1}^{t} f(i_j) < \mu - \theta \right] \leq \gamma.$$

Next, we present a theorem by Vadhan in [40] that describes detailed requirements for a locally computable extractor.

Theorem 5.8 (Theorem 10 in [40]). *Suppose that* $\mathsf{Samp} : \{0,1\}^r \to [n]^t$ *is a* (μ, θ, γ) *average sampler with distinct samples for* $\mu = (\delta - 2\tau)/\log(1/\tau)$ *and* $\theta = \tau/\log(1/\tau)$, *and* $\mathsf{Ext} : \{0,1\}^t \times \{0,1\}^d \to \{0,1\}^v$ *is a strong* $((\delta - 3\tau)t, \varepsilon)$ *extractor. Define* $\mathsf{Ext}' : \{0,1\}^n \times \{0,1\}^{r+d} \to \{0,1\}^v$ *by*

$$\mathsf{Ext}'(\boldsymbol{x}, (\boldsymbol{y}_1, \boldsymbol{y}_2)) = \mathsf{Ext}(\boldsymbol{x}_{\mathsf{Samp}(\boldsymbol{y}_1)}, \boldsymbol{y}_2).$$

Then Ext' *is a* t*-local strong* $(\delta n, \varepsilon + \gamma + 2^{-\Omega(\tau n)})$ *extractor.*

As we mentioned above, our application needs a locally computable extractor for larger alphabets, which may not be implied directly from Theorem 5.8. To tackle this issue, we define the following sampling procedure **Sampler 1** that outputs t distinct symbols of samples, and then prove that **Sampler 1** is in fact

a good average sampler as needed in Theorem 5.8. This would imply a locally computable extractor for larger alphabets as required in our application.

Notations for the Sampling. Before describing the algorithm, we set up some notations as follows. Let $\Gamma = \{0,1\}^k$ and $\boldsymbol{x} = (\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_n^\top)^\top \in \Gamma^n$ be a vector of n symbols, where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})^\top \in \Gamma = \{0,1\}^k$ for $i \in [n]$. Let S denote a subset of $[n] \times [k]$, i.e. S contains tuples $(i,j) \in [n] \times [k]$ as its elements. In this case, we define $\boldsymbol{x}_S = \{x_{ij}\}_{(i,j)\in S}$. Then, we define **Sampler 1** as below.

Sampler 1: Sample a random subset R of $[n]$ that contains t *distinct* elements, i.e., $R = \{r_1, \ldots, r_t\}$, and output $S := \{(r_i, j)\}_{i\in[t], j\in[k]}$. Then we derive the following lemma.

Lemma 5.9. *For any $\lambda \in \mathbb{Z}$, $\mu, \theta \in (0,1]$ and $\gamma = 2\lambda \exp(-t\theta^2/4) + \left(\frac{t(t-1)}{2n}\right)^\lambda$, **Sampler 1** is a (μ, θ, γ) averaging sampler.*

Proof. According to the natural bijection between $[nk]$ and $[n]\times[k]$, to prove that **Sampler 1** is a good average sampler as Definition 5.7, it suffices to show that for any $f : [n] \times [k] \to [0,1]$ such that $\frac{1}{nk}\sum_{i\in[n], j\in[k]} f(i,j) \geq \mu$, the following inequality holds:

$$\Pr_{S \xleftarrow{\$} \textbf{Sampler 1}} \left[\frac{1}{|S|} \sum_{(i,j)\in S} f(i,j) < \mu - \theta \right] \leq \gamma. \tag{1}$$

It might be hard to prove inequality (1) directly, since all blocks output by **Sampler 1** are distinct. To handle this issue, we then define the following **Sampler 2** through using "sample with replacement" and rejection sampling. It is not hard to show that these two procedures are statistically close. Furthermore, by using use a Chernoff bound argument, we show that **Sampler 2** is a good average sampler as required in Theorem 5.8. Thus, we conclude that **Sampler 1** with any strong extractor yields a locally computable extractor for larger alphabets.

Sampler 2:

1. Sample $R = \{r_1, \ldots, r_t\}$ from $[n]^t$ uniformly at random.
   - If all elements are distinct, then output $S := \{(r_i, j)\}_{i\in[t], j\in[k]}$ and terminate.
2. Otherwise, i.e., there is a repeated element, discard the whole sample and redo Step 1.
   Note: the algorithm will only redo Step 1 up to λ times. If the algorithm does not produce an output by then, then output $\perp$.

Next we analyze **Sampler 1** and **Sampler 2** by the following two claims. Due to space limit, we defer the full proof to the full version of this paper.

Claim 5.10. *For a set X consisting of $n = n(\lambda)$ different blocks and the parameters $t = t(\lambda)$ such that $t(t-1) < n$, the output distributions of Sample 1 and Sample 2 are statistically close.*

Claim 5.11. *For any μ, t, θ, n, Sampler 2 is a (μ, θ, γ) average sampler conditioned on non-$\perp$ output, where $\gamma = 2\lambda \exp(-t\theta^2/4)$.*

The proof of the lemma follows by the above Claims 5.10 and 5.11. □

Furthermore, by applying the **Sample 1** to Theorem 5.8 with the following parameters setting, we derive the following theorem.

Parameter Setting. Taking λ as the security parameter, we set all the parameters in the following way: $k = \mathsf{poly}(\lambda), n = \mathsf{poly}(\lambda), t = \lambda \log^3(nk), \delta = \frac{1}{\log(nk)}$, $\tau = \frac{1}{6\log(nk)}, \mu = \frac{2}{3\log(nk)\log(6\log(nk))}$, $\theta = \frac{1}{6\log(nk)\log(6\log(nk))}$, $\gamma = 2\lambda \exp(-t\theta^2/4) + \left(\frac{t(t-1)}{2n}\right)^{\lambda}$, $\varepsilon = \mathsf{negl}(\lambda)$.

Theorem 5.12. *Let $\Gamma = \{0,1\}^k$, $\mathrm{Samp} : \{0,1\}^r \to [n]^t$ be the **Sampler 1** (as a (μ, θ, γ) average sampler), and let $\mathsf{Ext} : \Gamma^t \times \{0,1\}^d \to \{0,1\}^v$ be a strong $((\delta - 3\tau)tk, \varepsilon)$ extractor. Define $\mathsf{Ext}' : \Gamma^n \times \{0,1\}^{r+d} \to \{0,1\}^v$ as*

$$\mathsf{Ext}'(\boldsymbol{x}, (\boldsymbol{y}_1, \boldsymbol{y}_2)) = \mathsf{Ext}(\boldsymbol{x}_{\mathrm{Samp}(\boldsymbol{y}_1)}, \boldsymbol{y}_2).$$

Then Ext' is a t-block-local strong $(\delta nk, \varepsilon + \gamma + 2^{-\Omega(\tau n)})$ extractor, where $\varepsilon + \gamma + 2^{-\Omega(\tau n)} = \mathsf{negl}(\lambda)$ according to the setting of parameters.

5.3 Leakage-Resilient Encryption in the Bounded-Retrieval Model

In this section, we construct leakage-resilient encryption schemes in the BRM, through combining an random extractor with an amplified AB-wHPS presented in Sect. 5.1. Below, we give the specific construction of leakage resilient ABE scheme in the BRM from an amplified AB-wHPS.

Construction 513. (Construction in the) BRM). *Let $\Pi = \mathsf{AB\text{-}wHPS}.\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Encap}^*, \mathsf{Decap}\}$ be an amplified AB-wHPS with integer parameters n', t, the encapsulated-key-space $\mathcal{K}^t$ and attribute space $\mathcal{X} = \{0,1\}^*$ for a class of policy functions $\mathcal{F} = \{f : \{0,1\}^* \to \{0,1\}\}$. Let $\mathsf{Ext} : \mathcal{K}^t \times \mathcal{S} \to \mathcal{M}$ be a strong extractor, where three sets $\mathcal{K}, \mathcal{S}, \mathcal{M}$ are efficient ensembles, k denotes the size of $\mathcal{K}$. Furthermore, assume that $\mathcal{M}$ is an additive group. Then, an ABE scheme $\Pi_{\mathcal{F}} = \Pi_{\mathcal{F}}.\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$ with message space $\mathcal{M}$ and policy function class $\mathcal{F}$ can be constructed as follows:*

- *$\Pi_{\mathcal{F}}.\mathsf{Setup}(1^\lambda)$: The algorithm runs $(\mathsf{mpk}^{\Pi}, \mathsf{msk}^{\Pi}) \xleftarrow{\$} \Pi.\mathsf{Setup}(1^\lambda)$, and outputs $\mathsf{mpk} := \mathsf{mpk}^{\Pi}$, and $\mathsf{msk} := \mathsf{msk}^{\Pi}$.*
- *$\Pi_{\mathcal{F}}.\mathsf{KeyGen}(\mathsf{msk}, f)$: $\Pi_{\mathcal{F}}.\mathsf{KeyGen}(\mathsf{msk}, f)$: Given a master secret-key msk and a function $f \in \mathcal{F}$ as input, the algorithm runs $\mathsf{sk}_f^{\Pi} \xleftarrow{\$} \mathsf{AB\text{-}wHPS.KeyGen}(\mathsf{msk}, f)$ and output $\mathsf{sk}_f := \mathsf{sk}_f^{\Pi}$.*

– $\Pi_{\mathcal{F}}.\mathsf{Enc}(\mathsf{mpk}, \boldsymbol{x}, \mu)$*: Given a master public-key* mpk*, an attribute* $\boldsymbol{x} \in \{0,1\}^*$ *and a message* $\mu \in \mathcal{M}$ *as input, the algorithm runs* AB-wHPS.Encap *to generate* $(\mathsf{CT}', \boldsymbol{k}) \leftarrow \mathsf{AB\text{-}wHPS.Encap}(\mathsf{mpk}, \boldsymbol{x})$ *with* $\boldsymbol{k} \in \mathcal{K}^t$*, and then samples* $s \xleftarrow{\$} \mathcal{S}$*. Furthermore, the algorithm computes and outputs*

$$\mathsf{ct} = (s, \mathsf{ct}_0, \mathsf{ct}_1) = (s, \mathsf{CT}', \mu + \mathsf{Ext}(\boldsymbol{k}, s)).$$

– $\Pi_{\mathcal{F}}.\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$*: Given a ciphertext* $\mathsf{ct} = (s, \mathsf{ct}_0, \mathsf{ct}_1)$ *and a secret key* sk_f *as input, the algorithm runs* AB-wHPS.Decap *to generate* $\boldsymbol{k} = \mathsf{AB\text{-}wHPS.Decap}(\mathsf{sk}_f, \mathsf{ct}_0)$ *with* $\boldsymbol{k} \in \mathcal{K}^t$*, and then output* $\mu = \mathsf{ct}_1 - \mathsf{Ext}(\boldsymbol{k}, s)$.

Parameter Setting. For security parameter λ, we set the system parameters as follows: $k = \mathsf{poly}(\lambda), n' = \mathsf{poly}(\lambda), t = \lambda \log^3(n'k)$, $\delta = \frac{1}{\log(n'k)}, \tau = \frac{1}{6\log(n'k)}, \varepsilon = \mathsf{negl}(\lambda)$. Moreover, for the proof of leakage-resilience in the BRM, we let $\mathsf{Ext} : \mathcal{K}^t \times \mathcal{S} \to \mathcal{M}$ be a $((\delta - 3\tau)tk, \varepsilon)$-extractor.

Next, we prove that the construction is a leakage resilient ABE in the BRM. Our proof uses a technique of locally computable extractors [40], i.e., Theorem 5.12, in a black-box way. Due to the space limit, we defer the detailed proof to the full version of this paper.

Theorem 5.14. *Assume* Π *is a selectively (or adaptively, resp.) secure amplified* AB-wHPS *with integer parameters* $n', t = \lambda \log^3(n'k)$ *for the policy function class* $\mathcal{F}$*, and* $\mathsf{Ext} : \mathcal{K}^t \times \mathcal{S} \to \mathcal{M}$ *be a strong extractor. Then the above* ABE *scheme* $\Pi_{\mathcal{F}} = \Pi_{\mathcal{F}}.\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$ *for* $\mathcal{F}$ *is a selectively (or adaptively, resp.)* ℓ*-leakage-resilient attribute-based encryption scheme with message space* $\mathcal{M}$ *in the* BRM *where* $\ell = kn' - \frac{kn'}{\log(kn')}$.

Particularly, $\Pi_{\mathcal{F}}$ *is also*

– *an* ℓ*-leakage-resilient public-key encryption scheme in the* BRM *with* $\ell = kn' - \frac{kn'}{\log(kn')}$*, if* $\mathcal{F}$ *contains only a single function that always outputs* 1.
– *a selectively (or adaptively, resp.)* ℓ*-leakage-resilient identity-based encryption scheme in the* BRM *with* $\ell = kn' - \frac{kn'}{\log(kn')}$*, if* $\mathcal{F}$ *contains the following comparison functions, i.e., each function* $f_{\boldsymbol{y}} \in \mathcal{F}$ *is indexed by a vector* $\boldsymbol{y}$*, and* $f_{\boldsymbol{y}}(\boldsymbol{x}) = 1$ *if and only if* $\boldsymbol{y} = \boldsymbol{x}$.

Moreover,

1. *Public-key (resp. master public-key) size of* $\Pi_{\mathcal{F}}$ *is the same as that of* Π*, which is not dependent on leakage parameter* ℓ.
2. *The locality-parameter is* $t = \lambda \log^3(n'k)$*. Thus, the size of secret-key accessed during decryption depends on* t*, but not* ℓ.
3. *The ciphertext-size/encryption-time/decryption-time of* $\Pi_{\mathcal{F}}$ *depends on* t*, but not* ℓ.

Combining Corollary 5.4 and Theorem 5.14, we obtain the following results. Assume there exists an ABE scheme with the message space $\mathbb{Z}_m$ for the function

class $\mathcal{F} \wedge_{\|} \mathcal{H} \wedge_{\|} \mathcal{G}$, where $\mathcal{G}$ with parameters m, n and $\mathcal{H}$ with parameter n' are as defined in Definitions 3.9 and 5.1, and the key-length (of the extra part, excluding the function description of f) of this underlying ABE scheme for policy function f is $s(f)$. Then the largest allowed leakage length of the above ABE (or IBE or PKE) scheme $\Pi_{\mathcal{F}}$ for the function class $\mathcal{F}$ is $\ell = (kn' - \frac{kn'}{\log(kn')})$ with $k = n \log m$ and the key-length of $\Pi_{\mathcal{F}}$ for f is $|\mathsf{sk}_f| = n'(n \log m + \log n' + |f| + s(\hat{f}_{f,h,g_y}))$.

Furthermore, if the secret key size $s(\hat{f}_{f,h,g_y}))$is succinct, i.e., $s(\hat{f}_{f,h,g_y}) = o(|\hat{f}_{f,h,g_y}|) = o(n \log m + \log n' + |f|)$, then we can set sufficiently large n, m, n' such that $(\log n' + |f|) = o(n \log m)$. Consequently, the leakage rate of this scheme $\Pi_{\mathcal{F}}$ is $\frac{kn' - \frac{kn'}{\log(kn')}}{n'(n \log m + \log n' + |f| + s(\hat{f}_{f,h,g_y}))} = \frac{1 - \frac{1}{\log(nn' \log m)}}{1 + \frac{\log n' + |f| + s(\hat{f}_{f,h,g_y})}{n \log m}} \approx 1 - o(1)$, achieving the desired optimal leakage rate.

Finally, by combining Corollary 3.14 and Theorem 5.14, we obtain the following Corollary.

Corollary 5.15. *Assuming* LWE*, for all polynomial* $S = \mathsf{poly}(\lambda)$*, there exist* $1 - o(1)$ *leakage resilient* ABE *schemes in the* BRM*, which are*

1. *adaptively secure for the comparison functions;*
2. *adaptively secure for* t-CNF* *functions of size up to* S*;*
3. *selectively secure for general circuits of size up to* S*.*

For unbounded polynomial S, our schemes are still leakage resilient with the optimal rate for a smaller function class. See Remark 4.4 for the discussion.

6 Extension II: Leakage on Multiple Keys

Our prior ABE constructions from AB-wHPS only achieve leakage resilience in the one-key setting where the adversary can only leak on one of the all possible decrypting keys with respect to the challenge attribute. In this section, we show how to achieve leakage resilience in the *multiple-key* setting where the attacker can obtain leakage on ω possible decrypting keys for any bounded polynomial ω. Our construction leverages the normal AB-wHPS (where the ciphertext indistinguishability holds when the adversary gets one decrypting key) and a threshold secret sharing scheme, following the bootstrapping idea of the work [21].

Construction 61. (Extended Leakage Resilient ABE). *Let* $\Pi = \Pi.\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Encap}^*, \mathsf{Decap}\}$ *be a* $(\log|\mathcal{K}|, \log|\mathcal{K}|)$*-universal* AB-wHPS *with the encapsulated-key-space* $\mathcal{K}$ *and attribute space* $\mathcal{X} = \{0,1\}^*$ *for a class of policy functions* $\mathcal{F} = \{f : \{0,1\}^* \to \{0,1\}\}$*. Let* $\mathsf{Ext} : \mathcal{K} \times \mathcal{S} \to \mathcal{M}$ *be a* $(\log|\mathcal{K}| - \ell, \varepsilon)$*-extractor, where* $\mathcal{K}, \mathcal{S}, \mathcal{M}$ *are efficient ensembles,* $\ell = \ell(\lambda)$ *is some parameter and* $\varepsilon = \varepsilon(\lambda) = \mathsf{negl}(\lambda)$ *is negligible. In addition, let* $(\mathsf{Share}, \mathsf{Rec})$ *be a* $(\hat{t}+1)$*-out-of-*n *threshold secret sharing scheme with respect to secret domain* $\mathcal{M}$*, an additive group.*

Then, a leakage-resilient ABE *scheme* $\Pi_{\mathcal{F}} = \Pi_{\mathcal{F}}.\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$ *with message space* $\mathcal{M}$ *for policy function class* $\mathcal{F}$ *can be constructed as follows:*

- $\Pi_{\mathcal{F}}.\mathsf{Setup}(1^\lambda, n)$: *The algorithm runs* $(\mathsf{mpk}_i^{\Pi}, \mathsf{msk}_i^{\Pi}) \xleftarrow{\$} \Pi.\mathsf{Setup}(1^\lambda)$ *for every* $i \in [n]$, *and outputs* $\mathsf{mpk} := \{\mathsf{mpk}_i^{\Pi}\}_{i\in[n]}$ *and* $\mathsf{msk} := \{\mathsf{msk}_i^{\Pi}\}_{i\in[n]}$.
- $\Pi_{\mathcal{F}}.\mathsf{KeyGen}(\mathsf{msk}, f)$: *Given a master secret-key* $\mathsf{msk} := \{\mathsf{msk}_i^{\Pi}\}_{i\in[n]}$ *and a function* $f \in \mathcal{F}$ *as input, the algorithm first chooses a random subset of cardinality* $\hat{t}+1$, *i.e.,* $\Gamma = \{r_1, \ldots, r_{\hat{t}+1}\} \subseteq [n]$, *and then runs* $\mathsf{sk}_f^{(r_i)} \xleftarrow{\$} \Pi.\mathsf{KeyGen}(\mathsf{msk}_{r_i}^{\Pi}, f)$ *for* $i \in [\hat{t}+1]$. *Finally, the algorithm outputs*

$$\mathsf{sk}_f := (\Gamma, \mathsf{sk}_f^{(r_1)}, \ldots, \mathsf{sk}_f^{(r_{\hat{t}+1})}).$$

- $\Pi_{\mathcal{F}}.\mathsf{Enc}(\mathsf{mpk}, \boldsymbol{x}, \mu)$: *Given a master public-key* $\mathsf{mpk} := \{\mathsf{mpk}_i^{\Pi}\}_{i\in[n]}$, *an attribute* $\boldsymbol{x} \in \mathcal{X} = \{0,1\}^*$ *and a message* $\mu \in \mathcal{M}$ *as input, the algorithm first runs* $(\mu_1, \ldots, \mu_n) \xleftarrow{\$} \mathsf{Share}(\mu)$. *Furthermore, the algorithm runs* $\Pi.\mathsf{Encap}$ *to generate* $(\mathsf{CT}_i, k_i) \xleftarrow{\$} \Pi.\mathsf{Encap}(\mathsf{mpkmpk}_i, \boldsymbol{x})$ *for every* $i \in [n]$. *Next, the algorithm samples* $s_1, \ldots, s_n \xleftarrow{\$} \mathcal{S}$, *and outputs*

$$\begin{aligned}\mathsf{ct} &= (s_1, \ldots, s_n, \mathsf{ct}_1, \ldots, \mathsf{ct}_n, \mathsf{ct}_{n+1}, \ldots, \mathsf{ct}_{2n}) \\ &= (s_1, \ldots, s_n, \mathsf{CT}_1, \ldots, \mathsf{CT}_n, \mu_1 + \mathsf{Ext}(k_1, s_1), \ldots, \mu_n + \mathsf{Ext}(k_n, s_n)).\end{aligned}$$

- $\Pi_{\mathcal{F}}.\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$: *Given a ciphertext* $\mathsf{ct} = (\{s_i\}_{i\in[n]}, \{\mathsf{ct}_i\}_{i\in[2n]})$ *and a secret key* $\mathsf{sk}_f = (\Gamma, \{\mathsf{sk}_f^{(r_i)}\}_{i\in[\hat{t}+1]})$ *as input, the algorithm first runs* $\Pi.\mathsf{Decap}$ *to generate* $k_{r_i} = \Pi.\mathsf{Decap}(\mathsf{sk}_f^{(r_i)}, \mathsf{ct}_{r_i})$ *and* $\mu_{r_i} = \mathsf{ct}_{n+r_i} - \mathsf{Ext}(k_{r_i}, s_{r_i})$ *for every* $i \in [\hat{t}+1]$. *Then, the algorithm outputs* $\mu = \mathsf{Rec}(\mu_{r_1}, \ldots, \mu_{r_{\hat{t}+1}})$.

Parameter Setting. For security parameter λ, given any $\omega = \mathsf{poly}(\lambda)$, we set $\hat{t} = \Theta(\omega^2\lambda)$ and $n = \Theta(\omega^2\hat{t})$. For details, we refer readers to the full version of this paper.

Our construction achieves a leakage resilient ABE in the multiple key setting. We summarize the results in the following theorem, and defer the full proof to the full version, due to space limit.

Theorem 6.2. *Assume* Π *is a selectively (or adaptively, resp.) secure* $(\log|\mathcal{K}|, \log|\mathcal{K}|)$*-universal* AB-wHPS *for the policy function class* $\mathcal{F}$, *and* $\mathsf{Ext} : \mathcal{K} \times \mathcal{S} \to \mathcal{M}$ *be a* $(\log|\mathcal{K}| - \ell, \mathsf{negl}(\lambda))$*-extractor. Then the above* ABE *scheme* $\Pi_{\mathcal{F}} = \Pi_{\mathcal{F}}.\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$ *for* $\mathcal{F}$ *is a selectively (or adaptively, resp.)* $(\ell(\lambda), \omega(\lambda))$*-leakage resilient attribute-based encryption scheme for* $\mathcal{F}$ *in the relative-leakage model, for any fixed bounded polynomial* $\omega(\lambda) = \mathsf{poly}(\lambda)$.

The corresponding leakage rate is $\frac{\ell(\lambda)}{(\hat{t}+1)(|\mathsf{sk}_f|+\log n)}$. *Furthermore, when the underlying secret keys* $(\mathsf{sk}_f^{(r_1)}, \ldots, \mathsf{sk}_f^{(r_{\hat{t}+1})})$ *form a block source under each leakage function, the corresponding leakage rate is* $\frac{\ell(\lambda)}{(|\mathsf{sk}_f|+\log n)}$.

Combining Theorem 3.12 and Theorem 6.2, we obtain the following results. Assume there exists an sel-ada/sel-sel (or ada-ada/ada-sel) secure ABE scheme with the message space $\mathbb{Z}_{\bar{m}}$ for the function class $\mathcal{F} \wedge_{\|} \mathcal{G}$, where $\mathcal{G}$ is the class

as in Definition 3.9 with parameters $\bar{m}, \bar{n}$, and the key-length (of the extra part, excluding the function description of f) of this underlying ABE scheme for policy function f is $s(f)$. Then the allowed leakage length of the above leakage resilient ABE scheme $\Pi_{\mathcal{F}}$ with parameters $n, \hat{t}, \omega$ as in the above paragraph setting for the function class $\mathcal{F}$ is $\ell = (\bar{n} \log \bar{m} - 2\lambda)$ and the key-length of $\Pi_{\mathcal{F}}$ for f is $|\mathsf{sk}_f| = (\hat{t}+1)(\log n + \bar{n} \log \bar{m} + |f| + s(\hat{f}_{f,g_y}))$.

Furthermore, if the secret key size $s(\hat{f}_{f,g_y})$ is succinct, i.e., $s(\hat{f}_{f,g_y}) = o(\bar{n} \log \bar{m} + |f|)$, then we can set sufficiently large $n, \bar{m}, \bar{n}$ such that $(\log n + |f|) = o(\bar{n} \log \bar{m})$. Consequently, when the underlying secret keys form a block source under each leakage function, the corresponding leakage rate of this scheme $\Pi_{\mathcal{F}}$ is $\frac{\bar{n} \log \bar{m} - 2\lambda}{\log n + \bar{n} \log \bar{m} + |f| + s(\hat{f}_{f,g_y})} = \frac{1 - \frac{2\lambda}{\bar{n} \log \bar{m}}}{1 + \frac{\log n + |f| + s(\hat{f}_{f,g_y})}{\bar{n} \log \bar{m}}} \approx 1 - o(1)$, achieving the desired optimal leakage rate.

Finally, by combining Corollary 3.14 and Theorem 6.2, we obtain the following Corollary.

Corollary 6.3. *Assuming* LWE*, for any* $S = \mathsf{poly}(\lambda)$ *and* $\omega = \mathsf{poly}(\lambda)$*, there exist* (ℓ, ω)*-leakage resilient* ABE*'s in the relative leakage model, which are*

1. *adaptively secure for* t-CNF* *functions of size up to* S*;*
2. *selectively secure for general circuits of size up to* S*.*

Moreover, when the underlying secret keys form a block source under the each leakage function, the corresponding leakage rate is $1 - o(1)$*.*

Furthermore, we can also achieve similar results in the BRM. By combining Corollary 3.14, Theorem 5.3 and Theorem 6.2, we obtain the following corollary.

Corollary 6.4. *Assuming* LWE*, for any polynomial* $S = \mathsf{poly}(\lambda)$ *and* $\omega = \mathsf{poly}(\lambda)$*, there exist* (ℓ, ω)*-leakage resilient* ABE *schemes in the* BRM*, which are*

1. *adaptively secure for* t-CNF* *functions of size up to* S*;*
2. *selectively secure for general circuits of size up to* S*.*

Moreover, when the underlying secret keys form a block source under the each leakage function, the corresponding leakage rate is $1 - o(1)$*.*

Acknowledgements. We would like to thank the reviewers of PKC 2022 for their insightful advices. Qiqi Lai is supported by the National Natural Science Foundation of China (62172266, 61802241, U2001205), the National Cryptography Development Foundation during the 13th Five-year Plan Period (MMJJ20180217), and the Fundamental Research Funds for the Central Universities (GK202103093).

Feng-Hao Liu and Zhedong Wang are supported by the NSF Career Award CNS-1942400.

References

1. Agrawal, D., Archambeault, B., Rao, J.R., Rohatgi, P.: The EM side—channel(s). In: Kaliski, B.S., Koç, K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 29–45. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36400-5_4
2. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert [19], pp. 553–572
3. Agrawal, S., Freeman, D.M., Vaikuntanathan, V.: Functional encryption for inner product predicates from learning with errors. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 21–40. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_2
4. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00457-5_28
5. Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-key encryption in the bounded-retrieval model. In: Gilbert [19], pp. 113–134
6. Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Halevi [24], pp. 36–54
7. Boneh, D., et al.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_30
8. Brakerski, Z., Goldwasser, S.: Circular and leakage resilient public-key encryption under subgroup indistinguishability. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 1–20. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_1
9. Brakerski, Z., Kalai, Y.T., Katz, J., Vaikuntanathan, V.: Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In: FOCS 2010 [18], pp. 501–510
10. Brakerski, Z., Lombardi, A., Segev, G., Vaikuntanathan, V.: Anonymous IBE, leakage resilience and circular security from new assumptions. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 535–564. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_20
11. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 595–624. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_20
12. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_4
13. Dodis, Y., Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Public-key encryption schemes with auxiliary inputs. In: Micciancio [32], pp. 361–381
14. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Cryptography against continuous memory attacks. In: FOCS 2010 [18], pp. 511–520
15. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008)

16. Dziembowski, S.: On forward-secure storage. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 251–270. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_15
17. Faust, S., Mukherjee, P., Nielsen, J.B., Venturi, D.: Continuous non-malleable codes. In: Lindell [30], pp. 465–488
18. 51st FOCS. IEEE Computer Society Press, October 2010
19. Gilbert, H. (ed.): EUROCRYPT 2010. LNCS, vol. 6110. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5
20. Gong, J., Chen, J., Dong, X., Cao, Z., Tang, S.: Extended nested dual system groups, revisited. In: Cheng, C.-M., Chung, K.-M., Persiano, G., Yang, B.-Y. (eds.) PKC 2016. LNCS, vol. 9614, pp. 133–163. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49384-7_6
21. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Functional encryption with bounded collusions via multi-party computation. In: Safavi-Naini and Canetti [36], pp. 162–179
22. Gorbunov, S., Vinayagamurthy, D.: Riding on asymmetry: efficient ABE for branching programs. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 550–574. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_23
23. Haldermany, J.A.: Lest we remember: cold boot attacks on encryption keys. Commun. ACM **52**(5), 91–98 (2008)
24. Halevi, S. (ed.): CRYPTO 2009. LNCS, vol. 5677. Springer, Heidelberg (2009)
25. Hazay, C., López-Alt, A., Wee, H., Wichs, D.: Leakage-resilient cryptography from minimal assumptions. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 160–176. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_10
26. Kiayias, A., Liu, F.-H., Tselekounis, Y.: Practical non-malleable codes from l-more extractable hash functions. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 16, pp. 1317–1328. ACM Press, Oct. (2016)
27. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68697-5_9
28. Lewko, A., Rouselakis, Y., Waters, B.: Achieving leakage resilience through dual system encryption. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 70–88. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_6
29. Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio [32], pp. 455–479
30. Lindell, Y. (ed.): TCC 2014. LNCS, vol. 8349. Springer, Heidelberg (2014)
31. Liu, F.-H., Lysyanskaya, A.: Tamper and leakage resilience in the split-state model. In: Safavi-Naini and Canetti [36], pp. 517–532
32. Micciancio, D. (ed.): TCC 2010. LNCS, vol. 5978. Springer, Heidelberg (2010)
33. Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi [24], pp. 18–35
34. Nisan, N., Zuckerman, D.: Randomness is Linear in Space. Academic Press Inc. (1996)
35. Nishimaki, R., Yamakawa, T.: Leakage-resilient identity-based encryption in bounded retrieval model with nearly optimal leakage-ratio. In: Lin, D., Sako, K. (eds.) PKC 2019. LNCS, vol. 11442, pp. 466–495. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17253-4_16
36. Safavi-Naini, R., Canetti, R. (eds.): CRYPTO 2012. LNCS, vol. 7417. Springer, Heidelberg (2012)

37. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
38. Tsabary, R.: Fully secure attribute-based encryption for t-CNF from LWE, pp. 62–85
39. Vadhan, S.P.: Pseudorandomness. Found. Trends Theor. Comput. Sci. **7**(1–3), 1–336 (2012)
40. Vadhan, S.P.: On constructing locally computable extractors and cryptosystems in the bounded storage model. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 61–77. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_4
41. Waters, B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: Halevi [24], pp. 619–636
42. Wee, H.: Dual system encryption via predicate encodings. In: Lindell [30], pp. 616–637
43. Zhang, L., Zhang, J., Mu, Y.: Novel leakage-resilient attribute-based encryption from hash proof system. Comput. J. **60**(4), 541–554 (2016)
44. Zhang, M., Zhang, Y., Su, Y., Huang, Q., Mu, Y.: Attribute-based hash proof system under learning-with-errors assumption in obfuscator-free and leakage-resilient environments. IEEE Syst. J. **11**(2), 1018–1026 (2017)