Equipping Public-Key Cryptographic Primitives with Watermarking (or: A Hole Is to Watermark)

Ryo Nishimaki[1]

NTT Secure Platform Laboratories, Tokyo, Japan
ryo.nishimaki.zk@hco.ntt.co.jp

Abstract. Program watermarking enables users to embed an arbitrary string called a mark into a program while preserving the functionality of the program. Adversaries cannot remove the mark without destroying the functionality. Although there exist generic constructions of watermarking schemes for public-key cryptographic (PKC) primitives, those schemes are constructed from scratch and not efficient.
In this work, we present a general framework to equip a broad class of PKC primitives with an efficient watermarking scheme. The class consists of PKC primitives that have a *canonical all-but-one (ABO) reduction*. Canonical ABO reductions are standard techniques to prove selective security of PKC primitives, where adversaries must commit a target attribute at the beginning of the security game. Thus, we can obtain watermarking schemes for many existing efficient PKC schemes from standard cryptographic assumptions via our framework. Most well-known selectively secure PKC schemes have canonical ABO reductions. Notably, we can achieve watermarking for public-key encryption whose ciphertexts and secret-keys are constant-size, and that is chosen-ciphertext secure.
Our approach accommodates the canonical ABO reduction technique to the puncturable pseudorandom function (PRF) technique, which is used to achieve watermarkable PRFs. We find that canonical ABO reductions are compatible with such puncturable PRF-based watermarking schemes.

Keywords. watermarking, public-key cryptography, all-but-one reduction

1 Introduction

1.1 Background

Watermarking. Watermarking enables us to embed an arbitrary string called a "mark" into a digital object such as images, videos, programs. While an embedded mark is extractable, a watermarked object should be almost functionally equivalent to the original one. Watermarking ensures that no one can remove an embedded mark without destroying the original functionality. Watermarking has two main applications. One is identifying ownership of an object. We can verify who is the original creator of objects by extracting an embedded mark that includes a unique identifier. The other is tracing malicious users that illegally copy objects. Therefore, watermarking deters unauthorized distribution.

Barak, Goldreich, Impagliazzo, Rudich, Sahai, Vadhan, and Yang initiated the study of program watermarking and gave rigorous definitions of cryptographic watermarking for programs [8]. They proved that program watermarking with perfect functionality-preserving property does not exist if there exists indistinguishability obfuscation (IO) [8]. Hopper, Molnar, and Wagner gave more definitions of cryptographic watermarking for perceptual objects and studied the relationships among them [34].

Earlier works presented watermarking schemes for specific classes of cryptographic functionalities [42,56,44]. However, those schemes are secure in restricted models where we limit adversary's strategies due to the impossibility results by Barak et al. [8]. That is, earlier works [42,56,44] do not consider arbitrary removal strategies. Cohen, Holmgren, Nishimaki, Vaikuntanathan, and Wichs presented the first watermarking scheme for pseudorandom functions (PRFs) against arbitrary removal strategies by introducing a relaxed functionality-preserving property [21]. In addition, they observed two facts: even if we relax the functionality-preserving property, (1) we need to pick a target circuit from a distribution with high min-entropy to avoid trivial attacks in the security game. (2) learnable circuit families are not watermarkable [21]. These two facts are the reasons why most studies on cryptographic watermarking [21,12,38,46,39,27,55] focus on cryptographic primitives rather than arbitrary circuits.

We focus on achieving secure watermarking for *public-key cryptographic primitives* against arbitrary removal strategies in this study since public-key primitives are more versatile than secret-key ones.

Why watermarking public-key primitives?: An application. Cohen et al. [21] presented an application of watermarked PRFs to electronic locks for cars. A car contains a PRF F and can only be opened by running a typical challenge-response identification protocol. A car owner has a software key (e.g., a smartphone application) that includes a marked PRF. We can embed some identifying information to PRFs. No one can remove the owner's information without losing the ability to unlock the car. Therefore, we can identify the car owner even if the software key is copied and the car is stolen (license plates can be forged). However, an automobile manufacturer can know user keys in this scenario since they are hard-coded in cars.[1]

If we can independently generate a key pair (public and secret-keys) of a public-key primitive from the watermarking setup, then an automobile manufacturer installs the public key to a car and need not know the secret-key. Therefore, we can run a typical challenge-response protocol by watermarkable public-key encryption (PKE) or signature without revealing secret-keys to manufacturers.[2]

Watermarking from scratch or retrofit. Goyal, Kim, Manohar, Waters, and Wu [27] presented the first feasibility result of watermarkable public-key cryptographic primitives from standard assumptions. This is an excellent work on

[1] If a car owner can directly install a PRF key into a car, and a watermarking scheme is public marking type, then watermarkable PRFs work in this scenario. However, this situation is not preferable.

[2] If a watermarking scheme is secret marking type, then we run a secure two-party computation between a user and a manufacturer.

general constructions of watermarkable public-key cryptographic primitives. However, their constructions of cryptographic primitives are built from scratch. Many efficient public-key cryptographic schemes (without watermarking functionalities) have been already proposed. One natural question is whether we can equip *existing* public-key cryptographic schemes with watermarking functionalities. If it is possible, we can obtain many efficient watermarkable cryptographic primitives. Our main question in this study is as follows.

Is there any general framework to equip public-key cryptographic schemes with watermarking functionalities?

We affirmatively answer to this question in this paper.

1.2 Our Contribution

We present a general framework to equip a broad class of public-key primitives with watermarking functionalities. The features of our watermarking schemes are as follows. Our watermarking schemes:

- almost preserve the efficiency of the original public-key primitives.
- apply to various primitives such as signature, PKE, key encapsulation mechanism (KEM), identity-based encryption (IBE), attribute-based encryption (ABE), inner-product encryption (IPE), predicate encryption (PE).
- are secure under the same assumptions as ones used in the original public-key primitives (i.e., CDH, decisional linear (DLIN), DBDH, short integer solution (SIS), LWE assumptions, and more).
- are independent of the original public-key primitives. (We do not need watermarking parameters to setup public-key primitives.)
- use simulation algorithms in security reductions of the original primitives.

More details of our watermarking schemes are explained in Sec. 1.4. We will explain our technique in Sec. 1.3.

Our primary advantages are: (1) semi-general applicability, that is, we can use many existing public-key schemes almost as they are. We do not need to construct watermarkable public-key schemes from scratch. (2) achieving CCA security for PKE. (3) efficiency based on concrete cryptographic assumptions. (See the comparison in Table 1.) Those are obtained from our framework using simulation algorithms.

Using proof techniques as real algorithms. Our construction technique significantly deviates from those of previous works. The most notable feature of our result is that we present a general method to use simulation algorithms that appear in reduction-based proofs as real cryptographic algorithms. Although our study is not the first study that uses simulation algorithms to achieve new cryptographic functionalities [44,35,36],[3] we present the first systematic approach

[3]Katsumata et al. [35,36] use simulation algorithms of ABE schemes to achieve homomorphic signatures.

using simulation algorithms in real schemes. We abstract a commonly used proof technique and show that if a public-key cryptographic scheme is proven to be secure via the proof technique, we can use simulation algorithms in the reduction as watermarked cryptographic functionalities. See Sec. 1.3 for the detail. This approach enables us to equip existing schemes with watermarking functionalities.

Terminology. Before we give a technical overview, we more formally explain watermarking. A watermarking scheme consists of three algorithms called setup, marking, and extraction algorithms. A setup algorithm Setup generates a marking key wmk and extraction key wxk. A marking algorithm Mark takes as input wmk, a circuit C, and a message ω, and outputs a marked circuit $\widetilde{C}$. Here, $\widetilde{C}$ should output the same output by C for most inputs. An extraction algorithm $\mathsf{Extract}$ takes as input wxk and circuit C', and outputs a string ω or special message $\mathsf{unmarked}$. This type of watermarking is called message-embedding. If Mark does not take ω as input and $\mathsf{Extract}$ outputs marked or $\mathsf{unmarked}$, then we call messageless watermarking. The basic security notion is unremovability, which means no adversary can construct a circuit C^* such that the functionality of C^* is almost equivalent to that of $\widetilde{C}$, but $\mathsf{Extract}(\mathsf{wxk}, C^*)$ outputs $\omega^* \neq \omega$. If we can/not publish wmk and wxk, then we call public/secret marking and public/secret extraction, respectively.

1.3 Technical Overview

We present how to equip public-key primitives that have *canonical all-but-one reductions*[4] with watermarking functionalities. All-but-one (ABO) reductions are standard proof techniques to prove selective security of public-key primitives [9,49,30,37,1,3,25,10,26]. Although our technique is not fully general, that is, we cannot apply our technique to *all* selectively secure public-key primitives, many well-known schemes fall into the class of canonical ABO reductions, where our technique applies. Roughly speaking, our watermarked cryptographic functionalities are simulation algorithms in ABO reductions. This technique is of independent interest because we can use simulators in security reductions as real algorithms for achieving new functionalities.

Our watermarking schemes based on canonical ABO reductions are messageless. To achieve message-embedding watermarking, we need to extend (canonical) ABO reductions to (canonical) all-but-N (ABN) reductions. However, ABO reductions are simpler to explain and it is easy to upgrade ABO reductions to ABN reductions for pairing-based schemes.[5] Thus, we first explain ABO reductions.

[4] See Sec. 4.2 for the formal definition and the meaning of "canonical".

[5] There is no general conversion from ABO to ABN reductions, but upgrading is possible for many concrete schemes by using programmable hash. See Sec. 4.5 for more detail.

All-but-one reduction. An ABO reduction is a polynomial-time algorithm that solves a problem instance π of a hard problem $\mathit{\Pi}$ by using an adversary $\mathcal{A}$ that breaks *selective security* of a cryptographic primitive $\mathit{\Sigma}$. To explain ABO reductions and selective security, we introduce oracles in security games.

Adversaries have access to oracles that receives queries from adversaries and returns answers in some security games. Adversaries also declare a target to attack $\mathit{\Sigma}$ at some point in the security game of $\mathit{\Sigma}$. We prohibit adversaries from sending a special query (or queries) that satisfies some conditions related to the target to prevent trivial attacks. We call such a special query "query on the target". In selective security games, adversaries must declare the target at the very beginning of the game.[6]

When we prove that if $\mathit{\Pi}$ is hard, then $\mathit{\Sigma}$ is selectively secure, we construct the following reduction R. After an adversary declares a target at the beginning of a selective security game, R simulates a public parameter by using a problem instance of $\mathit{\Pi}$ and the target and sends the public parameter to the adversary. Then, R simulates answers to all queries from the adversary *except the queries on the target* by using the problem instance (and the target). Note that R completes the simulation *without (master) secret-keys* of $\mathit{\Sigma}$. This type of reduction is called *all-but-one* reductions due to the simulation manner. In other words, if there exists an ABO reduction, then there exists an oracle simulation algorithm that works for all queries except the target.

We give an example. In the selective security game of signature, an adversary $\mathcal{A}$ declares a target message m^* at the beginning of the game. Then a challenger sends a public verification-key VK to $\mathcal{A}$. After that, $\mathcal{A}$ can send polynomially many queries (i.e., messages) and receives signatures corresponding to the queried messages (except m^*). At some point, $\mathcal{A}$ sends a challenge (m^*, σ^*).

A typical example of ABO reductions is the security reduction of the Boneh-Boyen signature scheme [9]. The reduction (or called simulator) R is given a CDH instance $\pi = (G, G^x, G^y)$ where G is a generator of a group $\mathbb{G}$. When the adversary $\mathcal{A}$ declares a target m^*, R simulates VK by using π and m^* (embedding π and m^* into VK). Next, R simulates signatures σ_{m} for queried message m from $\mathcal{A}$ except m^*. Here, R *implicitly* embeds G^{xy} into the signing key by setting parameters carefully (note that R does *not* have G^{xy}). Thus, if we assume $\mathcal{A}$ breaks the signature scheme, then R can extract G^{xy} from the forged signature σ^* output by $\mathcal{A}$.

Although R embeds m^* in VK, the distribution of VK by R is perfectly the same as the original distribution. In addition, R can perfectly simulate signatures for messages *except for the target message* m^* due to the embedding of m^*. For notational convention, we separate this signature simulation algorithm part as $\mathsf{SimSign}_{\neq \mathsf{m}^*}$. That is, we can construct an algorithm $\mathsf{SimSign}_{\neq \mathsf{m}^*}$ from π and m^* that outputs σ_{m} for input m except m^*. This is not necessarily possible for all selectively secure schemes since R might use oracle answers for simulation. Thus, we say a reduction is "canonical" if $\mathsf{SimSign}_{\neq \mathsf{m}^*}$ does not rely on oracle answers and is described as a stateless randomized algorithm. This proof style

[6] In adaptive security games, adversaries can select the target at any time.

is sometimes called *puncturing proof technique* [48] since m^* is like a *hole* in the message space and the reduction has no way to generate σ_{m^*} for m^*. The graphical explanation is described in Fig. 1.

Although the case of encryption is slightly different from that of signatures, we can consider similar simulation strategies for encryption. In the PKE case, there is no "attribute", but we can use a part of a ciphertext (sometimes called tag) as an attribute (in particular, in the CCA setting).

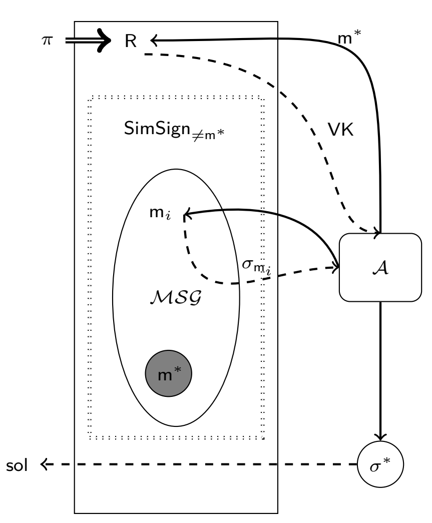

Fig. 1: Illustration of ABO reduction from the selective security of signature to Π. Solid lines denote outputs by the adversary $\mathcal{A}$ of signature. Dashed lines denote simulation by the reduction R. The grayed circle is the hole. Value sol denotes a solution to π.

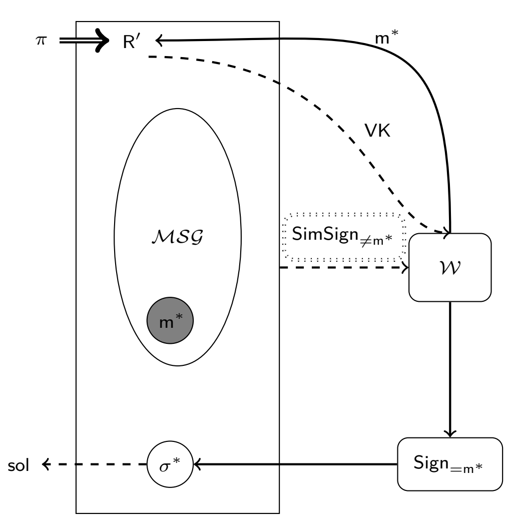

Fig. 2: Illustration of reduction from the security of watermarking to π. Solid lines denote outputs by the adverasry $\mathcal{W}$ of watermarking. Dashed lines denote simulation by reduction R'. The grayed circle is the hole. Value sol denotes a solution to π.

A hole is to watermark. We move to explain our unified framework to achieve watermarkable public-key primitives by using canonical ABO reductions. Roughly speaking, *a punctured hole in an ABO reduction works as a watermark because adversaries cannot fill the hole.* More concretely, we can consider the oracle simulation part $\mathsf{SimSign}_{\neq \mathsf{m}^*}$ of the canonical reduction R as a watermarked signature generation circuit in the signature case. In addition, no adversary can recover the ability to generate σ_{m^*} from $\mathsf{SimSign}_{\neq \mathsf{m}^*}$ because otherwise, the adversary can break the security of the signature scheme. (The message m^* is the target.)

The ABO oracle simulation algorithm $\mathsf{SimSign}_{\neq \mathsf{m}^*}$ preserves the functionality of the signature generation circuit except for an input m^*. To detect whether a circuit is watermarked or not, we check whether the circuit generates a correct output for the punctured input.[7] We can check whether a signature is valid for an message or not by using its verfication algorithm. If a circuit does not generate a valid output for the punctured input (i.e., the hole), then we consider it as watermarked. In almost all ABO reductions, we have efficient algorithms that check the validity of answers from oracles.

The unremovability holds as follows. We construct a reduction R' that solves a problem instance π by using a watermarking adversary $\mathcal{W}$. R' can give $\mathsf{SimSign}_{\neq \mathsf{m}^*}$ to $\mathcal{W}$ since R' has π and m^*.[8] Assume that $\mathcal{W}$ can remove the watermark. That is, we assume $\mathcal{W}$ is given $\mathsf{SimSign}_{\neq \mathsf{m}^*}$ and generates a circuit $\mathsf{Sign}_{= \mathsf{m}^*}$ that can generate a signature for the target m^* (i.e., filling *the hole*). Then, R' can break the security of signature. This is because $\mathsf{Sign}_{= \mathsf{m}^*}$ yields a forgery σ^* for the target m^*. We can extract the solution for π from σ^* as the ABO reduction for Boneh-Boyen signature scheme.

Put it differently, the canonical ABO reduction $\mathsf{R}(\pi)$ works as well even if we replace the adversary $\mathcal{A}$ of a cryptographic scheme Σ with the adversary $\mathcal{W}$ for watermarking, which removes the watermark. The modified reduction $\mathsf{R}'(\pi)$ can solve π because the power of removing the watermark by $\mathcal{W}$ leads to breaking the security of Σ. Therefore, the watermarking scheme is secure if the underlying problem is hard. The graphical explanation is described as in Fig. 2.

There are a few issues in the overview above. One issue is giving the description of $\mathsf{SimSign}_{\neq \mathsf{m}^*}$ to the adversary since it has only black-box access to the signature generation oracle in the security game. This issue is the reason why we use "canonical" ABO reductions. If ABO reductions satisfy the canonical property, then $\mathsf{SimSign}_{\neq \mathsf{m}^*}$ does not need oracle answers from the hard problem Π to simulate the signature generation oracle and can be described as a stateless randomized algorithm.

Another issue is how to prepare a problem instance and randomness for simulating VK in an ABO reduction. To create an ABO reduction in the real world, we need a problem instance π. However, what we have in the real world is not a problem instance but a secret signing-key. It is easy to find that we can perfectly simulate a problem instance and randomness for reductions by using a secret key in the real world for most ABO reductions. In addition, although $\mathsf{SimSign}_{\neq \mathsf{m}^*}$ includes randomness for simulating VK, this is not an issue thanks to the randomness of the problem instance π (i.e., secret-key in the real world). See Sections 4 to 6 for details.

Although we gave only intuitions in this section, we formalize properties of canonical ABO reductions in Sec. 4 and prove that we can achieve watermarking from canonical ABO reductions in Sections 5 and 6.

Extension to all-but-N reduction. The watermarking based on ABO reductions above is message-less watermarking. To embed an arbitrary N-bit string, we

[7]A useless circuit that outputs $\perp$ for all inputs is watermarked by this detection. To prevent this, we test the functionalities of circuits. See Sec. 6 for details.

[8]We do not explain how to determine m^* here since it is not essential in this overview.

need all-but-N reduction, which can simulate oracle answers except queries on N targets. Here, N is an a-priori bounded polynomial in the security parameter. We can easily extend known cryptographic primitives that have ABO reductions to ones that have all-but-N reductions by using the technique of programmable hash functions [33] for pairing-based cryptography. We also use the fully key-homomorphic technique [10] in the lattice setting or dynamic q-type assumptions [5] for the Boneh-Boyen IBE. See Sec. 4.4 for the detail.

First, we explain a reasonable but faulty idea to achieve message-embedding watermarking based on all-but-N reductions since it helps to understand our idea. We prepare N pairs of strings $\{t^*_{i,b}\}_{i\in[N],b\in\{0,1\}}$ as the public parameter of watermarking. To embed a message $\omega = (\omega_1, \ldots, \omega_N) \in \{0,1\}^N$, we consider an oracle simulation algorithm that can generate answers for queries except N points in $P := \{t^*_{1,\omega_1}, \ldots, t^*_{N,\omega_N}\}$. Concretely, in the case of signature, a signature oracle simulation algorithm $\mathsf{SimSign}_{\notin P}$ outputs a signature σ_{m} for a message m such that $\mathsf{m} \notin P$.[9] To extract an embedded message from a circuit C', we run the answer checking algorithm as in the message-less scheme for each $i \in [N]$ and $b \in \{0,1\}$. If C' outputs a valid $\sigma_{t^*_{i,1}}$ for input $t^*_{i,1}$ and does not output a valid $\sigma_{t^*_{i,0}}$ for input $t^*_{i,0}$, then we set the i-th bit of a message to 0 and vice versa.

This construction achieves the functionality of message-embedding watermarking. However, it is not secure because the adversary knows which points should not be punctured. That is, the points in $\overline{P} := \{t^*_{1,1-\omega_1}, \ldots, t^*_{N,1-\omega_N}\}$ (and P) are publicly available information. We call $\overline{P}$ the negation of punctured points P in this section. As already observed in some watermarkable PRFs [21,38,46], public punctured points could hurt watermarking security. In our case, adversary can easily destroy the functionality of cryptographic primitive at any point. More concretely, the adversary can easily modify a watermarked circuit where t^*_{i,ω_i} is punctured but $t^*_{i,1-\omega_i}$ is not punctured into a circuit that does not work for point $t^*_{i,1-\omega_i}$ too. Then, the extraction algorithm above outputs $\perp$ for the malformed circuit since the circuit outputs $\perp$ both for $t^*_{i,0}$ and $t^*_{i,1}$.

To solve the issue, we generate punctured points P and its negation $\overline{P}$ by using PRFs and hide them instead of using publicly known punctured points and its negation. This technique is commonly used in watermarkable PRFs [21,38,46]. We pseudo-randomly determine punctured points and its negation based on an embedded mark and the public parameter of the target master secret-key to be watermarked. Then, the adversary has no idea about the negation of punctured points $\overline{P}$ (and P). Therefore, it is hard for the adversary to intentionally modify a watermarked circuit into a circuit that does not work for points in $\overline{P}$. In fact, we must prepare many punctured points $p_i := (t^{(1)}_{i,\omega_i}, \ldots, t^{(T)}_{i,\omega_i})$ and its negation $\overline{p}_i := (t^{(1)}_{i,1-\omega_i}, \ldots, t^{(T)}_{i,1-\omega_i})$ for each bit position i and check all points to extract i-th bit of an embedded message, where T is a polynomial in the security parameter.

[9] All-but-N reductions should be able to generate N simulated challenge ciphertexts in the encryption case. This simulation is easy to achieve by using random self-reducibility of underlying hard problems for the discrete-logarithm-based case. In the LWE case, polynomially many (so, N) problem instances can be given.

If a circuit output $\bot$ for all points in p_i and *a correct value for at least one point in* $\overline{p}_i$, we extract ω_i as the i-th bit. To change the i-th bit of the embedded message without recovering the original functionality, adversaries must destroy the functionality of a circuit for all points in $\overline{p}_i$. Advesaries can indiscriminately destroy the functionality without knowing points $(p_i, \overline{p}_i)$. However, if the adversary makes a circuit that does not work for a $1/2$ plus a non-negligible fraction of inputs, then we can check that the circuit is not functionally similar to the original watermarked circuit. To make a circuit that is functionally similar to the watermarked circuit, but the extraction algorithm does not output ω_i from, all the adversary can do is recovering the functionality of the watermarked circuit at punctured points P (p_i). This event contradicts to all-but-N reductions as the case of the message-less scheme. Thus, we can achieve unremovability.

Although the message-embedding scheme above is secret marking and secret extraction, it is secure even if the adversary has the oracle access to the marking and extraction oracles. See Sec. 6 for the detail.

1.4 Comparison and Related Work

In this section, we review previous works on watermarking.[10] First, we compare our watermarking schemes with the schemes by Goyal et al. [27].

Efficient direct constructions and generic constructions. Goyal et al. [27] constructed a secret marking and secret extraction watermarking scheme for ABE (GKM+ABE) from mixed functional encryption (FE) and delegatable ABE, which can be instantiated only by the LWE assumption. They also constructed a public marking and public extraction watermarking scheme for PE (GKM+PE) from (bounded collusion-resistant) hierarchical FE, which can be instantiated by any PKE. Although the LWE assumption instantiates the schemes, the constructions are inefficient since they rely on heavy tools like mixed FE and hierarchical FE *even for watermarkable PKE.* In particular, in their watermarkable encryption schemes, not only the public key length but also the ciphertext length depend on the length of embedded massages (and the number of collusions in the GKM+PE case). The ciphertext size of GKM+ABE and GKM+PE is huge (See Table 1). They constructed a public marking and public extraction watermarking scheme for signature (GKM+SIG) from a prefix-constrained signature, which is instantiated with OWFs. GKM+SIG scheme is relatively efficient if it is instantiated with a signature scheme based on the symmetric external Diffie-Hellman (SXDH) assumption [18] since the transformation does not incur significant overhead.[11]

Our watermarking schemes can generally equip public-key primitives with watermarking functionalities if the primitives satisfy some conditions. The equipping procedure incurs only a little overhead. Although we need to modify public-key schemes so that they have $O(\ell\lambda)$-size master public parameters to achieve

[10] We do not consider constructions from strong assumptions such as IO in this study.

[11] We focus on constructions in the standard model in this paper. If we instantiate a signature scheme with Schnorr signature scheme [50], GKM+SIG would be more efficient.

Table 1: Efficiency Comparison of Message-Embedding Watermarking (Advanced) Public-Key Encryption and Signature. We ignore MPK part in MSK. In "Assumption" column, we put references for concrete instantiations. Parameters λ and ℓ are the security parameter and the length of marks, respectively. In general, $|\mathbb{G}| = c\lambda$ and $|\mathbb{G}_T| = c_T\lambda$ for some small constant c and c_T (depends on pairing groups). We do not put Ours2 in this table since it is message-less type.

	$\lvert\mathsf{MPK}\rvert$	$\lvert\mathsf{MSK}\rvert$	$\lvert\mathsf{SK}\rvert$ or $\lvert\sigma\rvert$	$\lvert\mathsf{CT}\rvert$	Assumption
GKM+ABE	$\mathrm{poly}(\lambda,\ell)$	$\mathrm{poly}(\lambda)$	$\mathrm{poly}(\lambda)$	$\mathrm{poly}(\lambda,\ell)$[c]	LWE [28]
GKM+PE	$Q\cdot\mathrm{poly}(\lambda,\ell)$	$Q\cdot\mathrm{poly}(\lambda,\ell)$	$\mathrm{poly}(\lambda,\ell)$	$Q\cdot\mathrm{poly}(\lambda,\ell)$[d]	PKE
Ours1 PKE[a]	$(2\ell\lambda+5)\lvert\mathbb{G}\rvert$	$(2\ell\lambda+2)\lvert\mathbb{Z}_p\rvert$	N/A	$6\lvert\mathbb{G}\rvert$	DLIN [37]
Ours1 KEM[b]	$(\ell\lambda+4)\lvert\mathbb{G}\rvert+\lvert\mathsf{hk}\rvert$	$(\ell\lambda+3)\lvert\mathbb{Z}_p\rvert$	N/A	$2\lvert\mathbb{G}\rvert+\lvert r\rvert$	DBDH [13]
Ours1 KEM[b]	$4\lvert\mathbb{G}\rvert+\lvert\mathsf{hk}\rvert$	$3\lvert\mathbb{Z}_p\rvert$	N/A	$2\lvert\mathbb{G}\rvert+\lvert r\rvert$	q-type [5]
Ours1 IBE	$(\ell\lambda+4)\lvert\mathbb{G}\rvert$	$(\ell\lambda+3)\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert+\lvert\mathbb{G}_T\rvert$	DBDH [9]
Ours1 IBE	$4\lvert\mathbb{G}\rvert$	$3\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert+\lvert\mathbb{G}_T\rvert$	q-type [5]
Ours1 IBE	$\ell\mathrm{poly}(\lambda)$	$\mathrm{poly}(\lambda)$	$\mathrm{poly}(\lambda)$	$\mathrm{poly}(\lambda)$[e]	LWE [10]
GKM+SIG	$(\ell+3)\lvert\mathbb{G}\rvert$	$\lvert\mathbb{Z}_p\rvert$	$(\ell+7)\lvert\mathbb{G}\rvert$	N/A	CDH [52]
GKM+SIG	$8\lvert\mathbb{G}\rvert+\lvert\mathbb{G}_T\rvert$	$8\lvert\mathbb{Z}_p\rvert$	$16\lvert\mathbb{G}\rvert+\lvert\mathbb{G}_T\rvert$	N/A	SXDH [18]
Ours3 SIG	$(\ell\lambda+4)\lvert\mathbb{G}\rvert$	$(\ell\lambda+3)\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{G}\rvert$	N/A	CDH [9]
Ours3 SIG	$4\lvert\mathbb{G}\rvert$	$3\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{G}\rvert$	N/A	q-type [5]
Ours3 SIG	$\ell\mathrm{poly}(\lambda)$	$\mathrm{poly}(\lambda)$	$\mathrm{poly}(\lambda)$	N/A	LWE [10]

a Tag-based encryption.
b Value hk and r are a hash key and randomness of a chameleon hash function.
c At least $\ell^7\lambda^7$.
d At least $\ell^2\lambda^2$ if instantiated with FE by Ananth and Vaikuntanathan [4].
e At most $O(\lambda^3\log^2\lambda)$.

message-embedding watermarking where ℓ is the mark length and λ is the security parameter, the size of signatures/secret-keys/ciphertexts does not change. The signatures/secret-keys/ciphertexts consist of only a few group elements if we use group-based schemes. In addition, if we use a q-type assumption, we can use the original Boneh-Boyen scheme as it is (even the master public key is constant-size). Thus, our watermarkable public-key primitives are as efficient as known efficient public-key primitives such as Boneh-Boyen IBE scheme [9]. Therefore, in the case of encryption, our schemes are more efficient than those of Goyal et al. in the asymptotic sense. See Table 1 for the efficiency comparison.

Functionalities of watermarking. In GKM+PE, GKM+SIG, and our schemes, the watermarking setup algorithms are completely separated from the key generation algorithm of public-key primitives. However, in GKM+ABE, we need the public parameter of the watermarking scheme to generate keys of public-key primitives.

Although our message-embedding scheme is secret marking and secret extraction, it is secure even if adversaries have access to marking and extraction oracles, which answer a marked circuit and an embedded mark for queried circuits, respectively. GKM+ABE is also secret marking and secret extraction and secure under the marking and extraction oracles, but the number of extraction queries is a-priori bounded. On the other hand, GKM+PE and GKM+SIG are public marking and public extraction.

Our schemes for signature/TBE/KEM/IBE and all GKM+ schmes are message-embedding watermarking, but our schemes for ABE/PE are message-less watermarking.

Watermarking user secret-keys v.s. master secret-keys. In GKM+ABE and GKM+PE, we can watermark user secret-keys such as secret-keys for identities (resp. policies) in IBE (resp. ABE). On the other hand, in our schemes, we can watermark master secret-keys of tag-based encryption (TBE), KEM, IBE, ABE, and PE. TBE is a variant of PKE. For signature/KEM/PKE cases, there is no difference since master secret-keys are user secret-keys in these cases.

Security level. There are several security measures. (1) Ours for TBE/KEM achieves CCA-security, but GKM+ABE and GKM+PE for PKE do not. (2) GKM+PE and GKM+SIG are adaptively secure, but GKM+ABE and ours are selectively secure in terms of public-key primitives. In terms of embedded messages, GKM+ schemes are adaptively secure, but ours are selectively secure. See Sec. 3 for selective security of watermarking. (3) All schemes are secure even if the authority of watermarking setup is corrupted. (4) Regarding the parameter on how much adversaries should preserve functionalities to succeed attacks, GKM+ schemes are better than ours. (GKM+ is $1/\mathrm{poly}(\lambda)$ while ours is $1/2 + 1/\mathrm{poly}(\lambda)$.) (5) We can consider three types of collusion-resistance in this study.

Collusion-resistance w.r.t. cryptographic primitives: In security games of cryptographic primitives, adversaries are often allowed to send queries to master secret-key based oracles that gives additional information such as signatures in the signature case and secret-keys for identities in the IBE case. We say collusion-resistant w.r.t. cryptographic primitives if cryptographic schemes are secure even in such a setting. Both GKM+SIG and our watermarking schemes for signatures are collusion-resistant w.r.t. cryptographic primitives. GKM+ABE and our watermarking schemes for encryption (IBE, ABE, and PE) are collusion-resistant w.r.t. cryptographic primitives. On the other hand, GKM+PE is *bounded* collusion-resistant w.r.t. cryptographic primitives, where the number of queries is a-priori bounded.

Collusion-resistance w.r.t. watermarkable cryptographic primitives: We say that a watermarking scheme is collusion-resistant w.r.t. watermarkable cryptographic primitives if it is unremovable even if adversaries have access to the master secret-key based oracle explained above in security games of watermarking for public-key primitives. Both GKM+SIG and our schemes for signature are collusion-resistant w.r.t. watermarkable cryptographic primitives. Our watermarking schemes for encryption (IBE, ABE, and PE) are collusion-resistant w.r.t. watermarkable cryptographic primitives, but GKM+ABE and GKM+PE schemes are not.

Collusion-resistance w.r.t. watermarking: We say that a watermarking scheme is collusion-resistant w.r.t. watermarking (collusion-resistant watermarking) if it is unremovable even if adversaries are given many watermarked

keys for the same original key. GKM+ABE, GKM+PE, and GKM+SIG are collusion-resistant watermarking, but ours are not.
We emphasize that even if watermarking schemes do not satisfy collusion-resistance w.r.t. watermarking, they have an application to *ownership identification.* This is because each user can use *different keys* in some settings, as we can see in the application to electronic car-lock in Sec. 1.1. Moreover, collusion-resistant watermarkable encryption is essentially the same as traitor tracing (the definition by Goyal [27] for PKE implies traitor tracing).[12] In some scenarios (ownership identification), traitor tracing (and collusion-resistant watermarking) is over-engineered. Thus, watermarking without collusion-resistance w.r.t. watermarking is meaningful enough. Moreover, if we would like to use collusion-resistant watermarkable PKE, we already have traitor tracing schemes [14,29]. If we want to trace users in public-key primitives, we can directly consider traceable primitives rather than collusion-resistant watermarkable public-key primitives.
The construction technique by Goyal et al. relies on that of traitor tracing [19,45] to achieve collusion-resistance w.r.t. watermarking.

Summary of comparison. We summarize watermarkable public-key primitives by Goyal et al. [27] and ours in Tables 1 and 2. PE and ABE include PKE/IBE/IPE as special cases. Notably, ours achieves CCA security for PKE. In addition, our message-embedding scheme (Ours1 in Table 2) is much more efficient than GKM+ABE and GKM+PE as we see in Table 1. In particular, the size of secret-keys and ciphertexts in our scheme does not depend on ℓ. If we use q-type assumption, then even the size of master public key does not depend on ℓ.

The disadvantages of Ours1 and Ours3 are (1) not collusion-resistant (2) secret marking/extraction (3) selective security (4) watermarking for master secret-keys (this is not a disadvantage for PKE and signature) (5) not supporting functionalities beyond IBE. We do not have a useful application of watermarking for master secret-keys in IBE/ABE/PE cases. On the other hand, all GKM+ constructions achieve collusion-resistance, watermarking for user secret keys, and support functionalities beyond IBE. GKM+PE and GKM+SIG achieve adaptive security. Although Ours2 is public marking/extraction and supports functionalities beyond IBE, it is message-less type and watermarking for master secret-keys. Therefore, GKM+ constructions and ours are incomparable.

More on related work. Cohen et al. gave the first positive result on program watermarking by introducing the statistical functionality-preserving property [21]. They presented public extraction message-embedding watermarkable PRFs based on IO. Subsequently, Kim and Wu [38,39] (KW17 and KW19) and Quach, Wichs, and Zirdelis [46] (QWZ18) presented secret extraction message-embedding watermarkable PRFs based on the LWE assumption. The KW19 and QWZ18

[12] Collusion-resistant watermarkable signatures may have an application to group signatures. However, the application is non-trivial since we should be able to trace users from signatures (not from signing keys) in the group signature setting.

Table 2: Comparison of Watermarking (Advanced) Public-Key Encryption. WM, CR, prim., auth., $\mathcal{MO}$, and $\mathcal{XO}$ stands for watermarking (or watermarkable), collusion-resistance, primitive, authority, marking oracle, and extraction oracle, respectively.

	GKM+ABE	Ours1	Ours2	GKM+PE	GKM+SIG	Ours3
Primitive	ABE	PKE[a]/IBE	ABE/IPE/PE	PE	SIG	SIG
Assumption	LWE	DBDH/DLIN/LWE		PKE	OWF	CDH/SIS
Message-embedding	✓	✓	×	✓	✓	✓
Public mark	×	×	✓	✓	✓	×
Against $\mathcal{MO}$ attack	✓	✓	✓	✓	✓	✓
Public extraction	×	×	✓	✓	✓	×
Against $\mathcal{XO}$ attack	bounded	✓	✓	✓	✓	✓
Separated setup	×	✓	✓	✓	✓	✓
Marking MSK	×	✓	✓	×	N/A	N/A
Marking SK	✓	×	×	✓	✓	✓
CCA-secure PKE	×	✓[a]	✓[a]	×	N/A	N/A
CR w.r.t. prim.	✓	✓	✓	bounded	✓	✓
CR w.r.t. WM prim.	×	✓	✓	×	✓	✓
CR w.r.t. WM	✓	×	N/A	bounded	✓	×
Selective/Adaptive	selective	selective	selective	adaptive	adaptive	selective
Sec. against auth.	✓	✓	✓	✓	✓	✓

a TBE and KEM.

schemes are secure against extraction oracle attacks. In addition, QWZ18 scheme is public marking. Regarding message-embedding watermarkable PRFs, KW17, KW19, and QWZ18 schemes are relatively efficient since they are based on the LWE assumption.

Baldimtsi, Kiayias, and Samari presented watermarking schemes for public-key primitives in a relaxed model, where a trusted watermarking authority generates not only watermarked keys but also unmarked keys and algorithms are stateful [7]. We do not compare their scheme because this is a weaker model.

Goyal et al. presented not only constructions but also rigorous definitions of watermarkable public-key primitives and a relaxed functionality-preserving property for watermarkable public-key primitives [27].[13]

Organization. In Sec. 2, we provide basic notions. Sec. 3 introduces the syntax and security definitions of watermarking. Sec. 4 defines canonical ABO reductions and gives examples of them. In Sec. 5, we present our message-less watermarking scheme. In Sec. 6, we present our message-embedding watermarking scheme and prove its security. Due to space limitations, we omitted many contents.

2 Preliminaries

We define some notations and introduce cryptographic notions in this section.

[13] Cohen et al. [20] considered watermarkable public-key primitives before Goyal et al., but even if a scheme satisfies their definitions, there exists simple attacks as observed by Goyal et al. [27].

Notations and basic concepts. If $\mathcal{X}^{(b)} = \{X_\lambda^{(b)}\}_{\lambda\in\mathbb{N}}$ for $b \in \{0,1\}$ are two ensembles of random variables indexed by $\lambda \in \mathbb{N}$, we say that $\mathcal{X}^{(0)}$ and $\mathcal{X}^{(1)}$ are computationally indistinguishable if for any PPT distinguisher $\mathcal{D}$, there exists a negligible function $\mathsf{negl}(\lambda)$, such that

$$\Delta \coloneqq |\Pr[\mathcal{D}(X_\lambda^{(0)}) = 1] - \Pr[\mathcal{D}(X_\lambda^{(1)}) = 1]| \leq \mathsf{negl}(\lambda).$$

We write $\mathcal{X}^{(0)} \overset{\mathsf{c}}{\approx} \mathcal{X}^{(1)}$ to denote that the advantage Δ is negligible.

The statistical distance between $\mathcal{X}^{(0)}$ and $\mathcal{X}^{(1)}$ over a countable set S is defined as $\Delta_{\mathsf{s}}(\mathcal{X}^{(0)}, \mathcal{X}^{(1)}) \coloneqq \frac{1}{2}\sum_{\alpha\in S} |\Pr[X_\lambda^{(0)} = \alpha] - \Pr[X_\lambda^{(1)} = \alpha]|$. We say that $\mathcal{X}^{(0)}$ and $\mathcal{X}^{(1)}$ are statistically/perfectly indistinguishable (denoted by $\mathcal{X}^{(0)} \overset{\mathsf{s}}{\approx} \mathcal{X}^{(1)}/\mathcal{X}^{(0)} \overset{\mathsf{p}}{\approx} \mathcal{X}^{(1)}$) if $\Delta_{\mathsf{s}}(\mathcal{X}^{(0)}, \mathcal{X}^{(1)}) \leq \mathsf{negl}(\lambda)$ and $\Delta_{\mathsf{s}}(\mathcal{X}^{(0)}, \mathcal{X}^{(1)}) = 0$, respectively. We also say that $\mathcal{X}^{(0)}$ is ϵ-close to $\mathcal{X}^{(1)}$ if $\Delta_{\mathsf{s}}(\mathcal{X}^{(0)}, \mathcal{X}^{(1)}) = \epsilon$.

Definition 2.1 (Circuit similarity). *Let $\mathcal{C}$ be a circuit class whose input space is $\{0,1\}^\ell$. For two circuits $C, C' \in \mathcal{C}$ and a non-decreasing function $\epsilon : \mathbb{N} \to \mathbb{N}$, we say that C is ϵ-close to C' if it holds that*

$$\Pr[C(x) \neq C'(x) \mid x \leftarrow \{0,1\}^\ell] \leq \epsilon. \textit{ (denoted by } C \cong_\epsilon C'\textit{)}$$

Similarly, we say that C is ϵ-far to C' if it holds that

$$\Pr[C(x) \neq C'(x) \mid x \leftarrow \{0,1\}^\ell] > \epsilon. \textit{ (denoted by } C \ncong_\epsilon C'\textit{)}$$

3 Definitions of Watermarking for Cryptographic Primitives

In this section, we introduce the definitions of watermarking for cryptographic primitives. Although our definitions basically follow those of Goyal et al. [27], there are several differences.

We focus on cryptographic primitives that have a master parameter generation algorithm PGen and a master secret-key based algorithm MSKAlg in this study. For example, in IBE/ABE/IPE, PGen is a setup algorithm Setup and MSKAlg is a key generation algorithm for identity/attribute/policy KeyGen. In TBE/KEM/signature, PGen is a key generation algorithm Gen and MSKAlg is a decryption/signing algorithm $\mathsf{Dec}/\mathsf{Sign}$. Hereafter, we do not explicitly treat KEM, but it is easy to adapt all definitions to the KEM setting. We formalize the notion of master secret-key based cryptographic schemes as follows.

Definition 3.1 (Master secret-key based cryptographic scheme). *A master secret-key based cryptographic scheme Σ with spaces $(\mathcal{T}, \mathcal{Q}, \mathcal{P}, \mathcal{R}_{\mathsf{mka}})$ has at least two algorithms* PGen *and* MSKAlg.

Master parameter generation: $\mathsf{PGen}(1^\lambda)$ *takes as input the security parameter and outputs a master public parameter* $\mathsf{PP} \in \mathcal{PP}$ *and a master secret key* $\mathsf{MSK} \in \mathcal{MSK}$. *We often omit spaces $\mathcal{PP}$ and $\mathcal{MSK}$ from Σ.*

Master secret-key based algorithm: $\mathsf{MSKAlg}(\mathsf{MSK}, X)$ *takes* MSK *and an input* $X \in \mathcal{Q}$ *and outputs* $Y \in \mathcal{P}$. *The randomness space of* MSKAlg *is* $\mathcal{R}_{\mathsf{mka}}$.

We assume that MSK *includes* PP. $\Sigma = (\mathsf{PGen}, \mathsf{MSKAlg}, \ldots)$ *has additional algorithm other than* PGen *and* MSKAlg. *The space* $\mathcal{T}$ *is used in the security game defined later.*[14]

Remark 3.1. In Def. 3.1, an output by MSKAlg is typically a secret key for an identity/policy X, signature for a message X. In the TBE case, X consists of a tag and ciphertext, and Y is a plaintext. We can consider encryption, decryption, and verification algorithms as additional algorithms. Def. 3.1 captures most popular cryptographic schemes such as PKE, TBE, IBE, ABE, IPE, PE, FE, signature, constrained signature.

Table 3: Concrete spaces and algorithms of master secret-key based cryptographic scheme.

	tag-based PKE	IBE	SIG
$\mathcal{T}$	tag space $\mathcal{TAG}$	identity space $\mathcal{ID}$	message space $\mathcal{MSG}$
$\mathcal{Q}$	tag and ciphertext space $\mathcal{TAG} \times \mathcal{CT}$	$\mathcal{ID}$	$\mathcal{MSG}$
$\mathcal{P}$	plaintext space $\mathcal{PT} \cup \{\perp\}$	secret key space $\mathcal{SK}$	signature space $\mathcal{SIG}$
$\mathsf{MSKAlg}(\mathsf{MSK}, \cdot)$	$\mathsf{Dec}(\mathsf{sk}, \cdot)$	$\mathsf{KeyGen}(\mathsf{MSK}, \cdot)$	$\mathsf{Sign}(\mathsf{sk}, \cdot)$

Definition 3.2 (Validity check algorithm for master secret-key based cryptographic scheme). *A master secret-key based cryptographic scheme* Σ *with spaces* $(\mathcal{T}, \mathcal{Q}, \mathcal{P}, \mathcal{R}_{\mathsf{mka}})$ *can have an optional algorithm* $\mathsf{Valid\text{-}Out}$ *that takes as inputs* PP, $X \in \mathcal{Q}$, *and* $Y \in \mathcal{P}$ *and outputs* $\top/\perp$. *For all* $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{PGen}(1^\lambda)$ *and all* $X \in \mathcal{Q}$, $\mathsf{Valid\text{-}Out}(\mathsf{PP}, X, Y)$ *outputs* $\top$ *if and only if* $Y \leftarrow \mathsf{MSKAlg}(\mathsf{MSK}, X)$.

Remark 3.2. Although we do not explicitly consider validity check algorithms in signature and advanced encryption schemes, we can implement validity check algorithms in most schemes (and all schemes in this paper). See examples in Sections 4.3 and 4.5. Note that Y is not necessarily unique since MSKAlg might be a randomized algorithm.

Definition 3.3 (Watermarkable Public-Key Scheme). *A watermarking scheme with mark space* $\mathcal{M}_{\mathsf{w}}$ *for master secret-key based cryptographic scheme* Σ *with spaces* $(\mathcal{T}, \mathcal{Q}, \mathcal{P}, \mathcal{R}_{\mathsf{mka}})$ *is a tuple of algorithms* $(\mathsf{WMSetup}, \mathsf{Mark}, \mathsf{Extract})$ *with the following properties:*

[14] Jumping ahead, $\mathcal{T}$ is a space where adversaries select targets at the beginning of security games.

Setup: $\mathsf{WMSetup}(1^\lambda)$ *takes as input the security parameter and outputs a watermarking public parameter* wpp*, a marking key* wmk*, and an extraction key* wxk.

Mark: $\mathsf{Mark}(\mathsf{wpp}, \mathsf{wmk}, \mathsf{MSK}, \omega)$ *takes as input* wpp*,* wmk*, the master secret key* $\mathsf{MSK} \in \mathcal{MSK}$ *of* Σ*, and a mark* $\omega \in \mathcal{M}_{\mathsf{w}}$ *and outputs a deterministic circuit* $\widetilde{C} : \mathcal{Q} \times \mathcal{R}_{\mathsf{mka}} \to \mathcal{P}$*. Note that* $\widetilde{C}$ *explicitly takes the randomness of* MSKAlg.

Extract: $\mathsf{Extract}(\mathsf{wpp}, \mathsf{wxk}, \mathsf{PP}, C')$ *takes as input* wpp*,* wxk*, the public parameter* $\mathsf{PP} \in \mathcal{PP}$ *of* Σ*, and a circuit* $C' : \mathcal{Q} \times \mathcal{R}_{\mathsf{mka}} \to \mathcal{P}$ *and outputs a mark* $\omega' \in \mathcal{M}_{\mathsf{w}}$ *or a special symbol* $\mathsf{unmarked}$.

Remark 3.3. We can separately treat watermarking schemes and cryptographic primitives in our definition while in the definition of Goyal et al. [27], key generation algorithms of cryptographic primitives need public parameters of watermarking. The separated definition is preferable and the same definition as that of Cohen et al. [21].

Hereafter, we set $\mathsf{wsk} := \mathsf{wmk} = \mathsf{wxk}$ since we consider only two cases. One is the public marking and extraction case ($\mathsf{wmk} = \mathsf{wxk} = \bot$) and the other is the secret marking and extraction case ($\mathsf{wsk} = \mathsf{wmk} = \mathsf{wxk}$) in this paper.

Hereafter, we focus on advanced encryption (IBE, IPE, ABE, PE) rather than TBE and signature for readability. Due to space limitations, we omit the definitions for TBE and signature.

Definition 3.4 (Correctness (Advanced encryption)). *Let* $\mathsf{WM}_\Sigma = (\mathsf{WMSetup}, \mathsf{Mark}, \mathsf{Extract})$ *be a watermarking scheme for advanced encryption scheme* $\Sigma = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *with spaces* $(\mathcal{T}, \mathcal{Q}, \mathcal{P}, \mathcal{R}_{\mathsf{mka}})$*. In this case,* $\mathcal{T} = \mathcal{ATT}$*,* $\mathcal{Q} = \mathcal{POL}$*,* $\mathcal{P} = \mathcal{SK}$*, where* $\mathcal{ATT}$ *and* $\mathcal{POL}$ *is an attribute and policy space, respectively. We say that* WM_Σ *is correct if it satisfies the following.*

Extraction correctness: *For all* $(\mathsf{wpp}, \mathsf{wsk}) \leftarrow \mathsf{WMSetup}(1^\lambda)$*, all marks* $\omega \in \mathcal{M}_{\mathsf{w}}$*,*

$$\Pr\left[\mathsf{Extract}(\mathsf{wpp}, \mathsf{wsk}, \mathsf{PP}, \widetilde{C}) \neq \omega \;\middle|\; \begin{array}{r}(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \widetilde{C} \leftarrow \mathsf{Mark}(\mathsf{wpp}, \mathsf{wsk}, \mathsf{MSK}, \omega)\end{array}\right] \leq \mathsf{negl}(\lambda).$$

Meaningfulness: *There are two variants of meaningfulness.*

Strong meaningfulness. *For all fixed circuits* $C : \mathcal{POL} \times \mathcal{R}_{\mathsf{mka}} \to \mathcal{SK}$*,*

$$\Pr\left[\begin{array}{l}\mathsf{Extract}(\mathsf{wpp}, \mathsf{wsk}, \mathsf{PP}, C) \\ = \mathsf{unmarked}\end{array} \;\middle|\; \begin{array}{l}(\mathsf{wpp}, \mathsf{wsk}) \leftarrow \mathsf{WMSetup}(1^\lambda) \\ (\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)\end{array}\right] > 1 - \mathsf{negl}(\lambda).$$

Weak meaningfulness. *For all* $(\mathsf{wpp}, \mathsf{wsk}) \leftarrow \mathsf{WMSetup}(1^\lambda)$*,*

$$\Pr\left[\begin{array}{l}\mathsf{Extract}(\mathsf{wpp}, \mathsf{wsk}, \mathsf{PP}, \mathsf{KeyGen}(\mathsf{MSK}, \cdot)) \\ = \mathsf{unmarked}\end{array} \;\middle|\; (\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)\right] > 1 - \mathsf{negl}(\lambda).$$

Functionality-preserving: *For all* $(\mathsf{wpp}, \mathsf{wsk}) \leftarrow \mathsf{WMSetup}(1^\lambda)$*, for all* $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$*, all marks* $\omega \in \mathcal{M}_{\mathsf{w}}$*, there exists* $\mathcal{PS} \subset \mathcal{ATT}$ *such that* $N :=$

$|\mathcal{PS}| \leq \mathrm{poly}(\lambda)$, for all $\rho_{\mathsf{mka}} \in \mathcal{R}_{\mathsf{mka}}$, all attributes $x \in \mathcal{ATT} \setminus \mathcal{PS}$ and all policy $\mathsf{P} \in \mathcal{POL}$ such that $\mathsf{P}(x) = \top$, we have that

$$\Pr[\widetilde{C}(\mathsf{P}, \rho_{\mathsf{mka}}) \overset{\mathsf{p}}{\approx} \mathsf{KeyGen}(\mathsf{MSK}, \mathsf{P}) \mid \widetilde{C} \leftarrow \mathsf{Mark}(\mathsf{wpp}, \mathsf{wsk}, \mathsf{MSK}, \omega)] > 1 - \mathsf{negl}(\lambda).$$

Here, $\mathcal{PS}$ stands for a "punctured set" since $\widetilde{C}$ does not work for policy P such that $x \in \mathcal{PS}$ and $\mathsf{P}(x) = \bot$.

Condition $\mathsf{P}(x) = \bot$ means attribute x is not qualified to policy P.

In the IBE case, $\mathcal{T} = \mathcal{Q} = \mathcal{ID}$ (identity space), $\mathsf{P} = \mathsf{id}_i$, $x = \mathsf{id}$, and $\mathsf{P}(x) = \bot$ means $\mathsf{id}_i \neq \mathsf{id}$.

Remark 3.4. Although our definition has a few differences from the standard functionality preserving in the cryptographic watermarking context [21,38] on the surface, ours is basically the same as the standard one. We select the definition above to emphasize that there exists a punctured set $\mathcal{PS}$, and the set is explicitly used in the security definition.

In addition, this functionality-preserving is stronger than that by Goyal et al. [27] since the output distribution of marked circuits is perfectly the same as that of the original circuit on almost all inputs.

Definition 3.5 (Selective-Mark ϵ-Unremovability for Advanced Encryption). *For every PPT $\mathcal{A}$, we have*

$$\Pr[\mathsf{Exp}^{\mathsf{urmv\text{-}enc}}_{\mathcal{A},\mathsf{WM}_{\Sigma}}(\lambda, \epsilon) = 1] \leq \mathsf{negl}(\lambda),$$

where ϵ is a parameter of the scheme called the approximation factor *and $\mathsf{Exp}^{\mathsf{urmv\text{-}enc}}_{\mathcal{A},\mathsf{WM}_{\Sigma}}(\lambda, \epsilon)$ is the game defined as follows.*

1. *The adversary $\mathcal{A}$ declares a target mark $\omega^* \in \mathcal{M}_{\mathsf{w}}$.*
2. *The challenger generates $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{wpp}, \mathsf{wsk}) \leftarrow \mathsf{WMSetup}(1^\lambda)$, and $\widetilde{C} \leftarrow \mathsf{Mark}(\mathsf{wpp}, \mathsf{wsk}, \mathsf{MSK}, \omega^*)$, and gives $(\mathsf{PP}, \mathsf{wpp}, \widetilde{C})$ to $\mathcal{A}$. At this point, a set $\mathcal{PS} \subset \mathcal{T}$ such that $|\mathcal{PS}| = \mathrm{poly}(\lambda)$ is uniquely determined by $(\mathsf{wpp}, \mathsf{wsk}, \mathsf{PP}, \omega^*)$.*
3. *$\mathcal{A}$ has oracle access to the key generation oracle $\mathcal{KO}$. If $\mathcal{KO}$ is queried with a policy $\mathsf{P} \in \mathcal{POL}$ such that $\mathsf{P}(t_i^*) = \bot$ for all $t_i^* \in \mathcal{PS}$, then $\mathcal{KO}$ answers with $\mathsf{KeyGen}(\mathsf{MSK}, \mathsf{P})$. Otherwise, it answers $\bot$. Condition $\mathsf{P}(x) = \bot$ means attribute x is not qualified to policy P.*
4. *$\mathcal{A}$ has oracle access to the marking oracle $\mathcal{MO}$. If $\mathcal{MO}$ is queried with a master secret key $\mathsf{MSK}' \in \mathcal{MSK}$ and a mark $\omega' \in \mathcal{M}_{\mathsf{w}}$, then does the following. If the corresponding master public parameter PP' is equal to PP, then outputs $\bot$. Otherwise, answers with $\mathsf{Mark}(\mathsf{wpp}, \mathsf{wsk}, \mathsf{MSK}', \omega')$.*
5. *$\mathcal{A}$ has oracle access to the extraction oracle $\mathcal{XO}$. If $\mathcal{XO}$ is queried with a PP' and circuit C', then $\mathcal{XO}$ answers with $\mathsf{Extract}(\mathsf{wpp}, \mathsf{wsk}, \mathsf{PP}', C')$.*
6. *Finally, $\mathcal{A}$ outputs a circuit C^*. If $\mathcal{A}$ is admissible (defined below) and $\mathsf{Extract}(\mathsf{wpp}, \mathsf{wsk}, \mathsf{PP}, C^*) \neq \omega^*$ then the experiment outputs 1, otherwise 0.*

We say that $\mathcal{A}$ is ϵ-admissible if C^* output by $\mathcal{A}$ in the experiment above satisfies

$$\Pr\left[\mathsf{Valid\text{-}Out}(\mathsf{PP},\mathsf{P},C^*(\mathsf{P},\rho_{\mathsf{mka}}))=\top \;\middle|\; \begin{array}{l}\mathsf{P}\leftarrow\mathcal{POL}\\ \rho_{\mathsf{mka}}\leftarrow\mathcal{R}_{\mathsf{mka}}\end{array}\right]\geq\epsilon.$$

See Def. 3.2 for Valid-Out*.*

The admissibility requires the adversary to output C^* that agrees on an ϵ fraction of inputs with C. This formalizes that C^* should be similar to the original circuit C.

Remark 3.5. Our definition is the same as that of Goyal et al. [27] except for that

1. $\mathcal{A}$ must declare the target mark ω at the beginning of the game.
2. $\mathcal{A}$ does not receives answers for inputs in $\mathcal{PS}$ from the key generation oracle.
3. we do not consider collusion-resistance w.r.t. watermarking. That is, $\mathcal{A}$ is given only one target circuit $\widetilde{C}$.
4. we consider the oracles $\mathcal{KO}$ in the unremovability game while Goyal et al. do not.
5. we consider watermarking for *master secret-keys*. Thus, the admissible condition for advanced encryption (i.e., beyond PKE or TBE) is in terms of Valid-Out.

Unforgeability. We can consider another security notion for watermarking, called unforgeability [21,12,38], in the secret marking setting. Unforgeability says that adversaries cannot generate a marked circuit with sufficiently different functionality from that of given marked circuits without a marking key.

We do not formally define unforgeability in this work as Goyal et al. did not. However, we can achieve unforgeability by embedding not only a mark but also a signature for the embedded mark and master public key as Goyal et al. observed [27].[15]

On security against malicious authority. Our watermarkable public-key primitives are trivially secure against authorities of watermarking schemes if the underlying public-key primitives are secure since parameter generation algorithms PGen are independent of watermarking setup algorithms WMSetup. Thus, we omit the definition of security against malicious authority.

4 All-But-One Reductions

In this section, we formalize a class of security reductions, called canonical all-but-one (ABO) reductions. Canonical ABO reductions are often used to prove the hardness of breaking many cryptographic primitives. A typical example is the security reduction of Boneh-Boyen IBE based on the decisional bilinear Diffie-Hellman assumption [9].

[15] ePrint archive report 2019/628, Section 3.4 and C.4 (version 20190908).

4.1 Assumptions and Security Games

We need to define cryptographic assumptions and security games before we formalize canonical ABO reductions. The types of reductions depend on whether security games and underlying cryptographic assumptions are computational or decisional. Therefore, we consider two types of assumptions and games. However, we focus on the decisional case in the main body for readability. See the full version for the computational case.

Definition 4.1 (Decisional assumption). *A decisional assumption* DA *for problem* Π *is formalized by a game between the challenger* $\mathcal{E}$ *and the adversary* $\mathcal{A}$*. The problem* Π *consists of an efficient problem sampling algorithm* $\mathsf{PSample}_b$ *for* $b \in \{0,1\}$*. The game* $\mathsf{Expt}^{\mathsf{DA}}_{\Pi,\mathcal{E}\leftrightarrow\mathcal{A}}(\lambda, b)$ *is formalized as follows.*

- *On input security parameter* λ, $\mathcal{E}$ *samples a problem instance* $\pi_b \leftarrow \mathsf{PSample}_b(1^\lambda)$.
- $\mathcal{E}$ *sends* π_b *to* $\mathcal{A}$ *and may interact with* $\mathcal{A}(1^\lambda, \pi_b)$.
- *At some point,* $\mathcal{A}$ *outputs a guess* coin^* *and the game outputs* coin^*.

We say a decisional assumption holds (or problem Π *is hard) if it holds*

$$\mathsf{Adv}^{\mathsf{DA}}_{\Pi,\mathcal{E}\leftrightarrow\mathcal{A}}(\lambda) := |\Pr[\mathsf{Expt}^{\mathsf{DA}}_{\Pi,\mathcal{E}\leftrightarrow\mathcal{A}}(\lambda,0)=1] - \Pr[\mathsf{Expt}^{\mathsf{DA}}_{\Pi,\mathcal{E}\leftrightarrow\mathcal{A}}(\lambda,1)=1]| \le \mathsf{negl}(\lambda).$$

This definition captures the well-known DDH, DBDH, k-Lin, matrix-DDH, quadratic residuosity, LWE, decisional q-type assumptions (and more). Note that the assumption above also captures interactive oracle assumptions since $\mathcal{A}$ may interact with the challenger that plays the role of oracles.

Definition 4.2 (Selective Security Game (Decisional Case)). *We define selective security games (decisional case) between a challenger* $\mathcal{C}$ *and an adversary* $\mathcal{A}$ *for a master secret-key based scheme* Σ *with spaces* $(\mathcal{T}, \mathcal{Q}, \mathcal{P}, \mathcal{R}_{\mathsf{mka}})$ *associated with challenge space* $\mathcal{H}$*, challenge answer space* $\mathcal{I}$*, and admissible condition* AdmI*. (See Table 4 for concrete examples.) The admissible condition* AdmI *outputs* $\top$ *or* $\perp$ *depending on whether a query is allowed or not.*

We define the experiment $\mathsf{Exp}^{\mathsf{d\text{-}goal\text{-}atk}}_{\mathcal{A},\Sigma}(\lambda, \mathsf{coin})$ *between an adversary* $\mathcal{A}$ *and a challenger as follows.*

1. $\mathcal{A}$ *submits a target* $t^* \in \mathcal{T}$ *to the challenger.*
2. *The challenger runs* $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{PGen}(1^\lambda)$*, and gives* PP *to* $\mathcal{A}$.
3. $\mathcal{A}$ *sends a query* $\mathsf{query} \in \mathcal{Q}$ *to the challenger. If* $\mathsf{AdmI}(t^*, \mathsf{query}) = \top$*, the challenger sends an answer* $\mathsf{answer} \leftarrow \mathsf{MSKAlg}(\mathsf{MSK}, \mathsf{query})$ *to* $\mathcal{A}$*. On the other hand, if* $\mathsf{AdmI}(t^*, \mathsf{query}) = \perp$*, the challenger outputs* $\perp$*. (*$\mathcal{A}$ *can send polynomially many queries.)*
4. *At some point,* $\mathcal{A}$ *sends a challenge* $\mathsf{challenge} \in \mathcal{H}$ *to the challenger. The challenger generates a challenge answer* $\mathsf{c\text{-}ans}^* \in \mathcal{I}$ *by using* $(t^*, \mathsf{PP}, \mathsf{challenge}, \mathsf{coin})$ *(denoted by* $\mathcal{C}_{\mathsf{a}}(t^*, \mathsf{PP}, \mathsf{challenge}, \mathsf{coin})$*) and sends* $\mathsf{c\text{-}ans}^*$ *to* $\mathcal{A}$.
5. *Again,* $\mathcal{A}$ *is allowed to query (polynomially many)* $\mathsf{query} \in \mathcal{Q}$ *such that* $\mathsf{AdmI}(t^*, \mathsf{query}) = \top$.
6. $\mathcal{A}$ *outputs a guess* coin^* *for* coin*. The experiment outputs* coin^*.

We say that Σ is secure if for all $\mathcal{A}$, it holds that

$$\mathsf{Adv}^{\mathsf{d\text{-}goal\text{-}atk}}_{\mathcal{A},\Sigma}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{d\text{-}goal\text{-}atk}}_{\mathcal{A},\Sigma}(\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathsf{d\text{-}goal\text{-}atk}}_{\mathcal{A},\Sigma}(\lambda, 1) = 1]| \leq \mathsf{negl}(\lambda).$$

We say an adversary is successful if the advantage is non-negligible. We can consider the multi-challenge case, where the targets are $\vec{t}^* \in \mathcal{T}^N$ instead of the single t^*.

A concrete example of $\mathsf{Adml}(t^*, \mathsf{query})$ is $\mathsf{Adml}(t^*, \mathsf{query}) = \top$ if and only if $t^* \neq t$ where $\mathsf{query} = t$ in the signature/TBE/IBE cases (t is a message/tag/identity).

Although we can consider a stronger variant, called adaptive security games, we consider only selective security games since ABO reductions are basically applicable in the selective setting.

4.2 Abstraction of All-But-One Reductions for Decisional Case

Now, we are ready to define ABO reductions for the decisional case. We put red underlines on the parts related to "canonical" parts.

First, we present a simplified definition that does not capture the TBE/KEM case for readability.

Definition 4.3 (Canonical All-But-One Reduction for Decisional Case (Simplified)). *Let Σ be a master secret-key based scheme with $(\mathcal{T}, \mathcal{Q}, \mathcal{P}, \mathcal{R}_{\mathsf{mka}})$ associated with challenge space $\mathcal{H}$, challenge answer space $\mathcal{I}$, and admissible condition* Adml. *(See Table 4 for concrete examples.) A security reduction algorithm* R *from Σ to a hard problem Π is a canonical all-but-one reduction (or Σ has a canonical all-but-one reduction to Π) if it satisfies the following properties.*

Oracle access: *$\mathcal{A}$ has oracle access to $\mathcal{O}_{\mathsf{MSK}} : \mathcal{Q} \to \mathcal{P}$ in the security game $\mathsf{Exp}^{\mathsf{d\text{-}goal\text{-}atk}}_{\mathcal{A},\Sigma}$. This oracle receives a query $\mathsf{query} \in \mathcal{Q}$ and does the following. If $\mathsf{Adml}(t^*, \mathsf{query}) = \top$, where t^* is defined below, it sends an answer $\mathsf{answer} \leftarrow \mathsf{MSKAlg}(\mathsf{MSK}, \mathsf{query})$ to $\mathcal{A}$. On the other hand, if $\mathsf{Adml}(t^*, \mathsf{query}) = \bot$, it outputs $\bot$.*

Selective reduction: R *simulates the security game $\mathsf{Exp}^{\mathsf{d\text{-}goal\text{-}atk}}_{\mathcal{A},\Sigma}$ of Σ between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ to win the game $\mathsf{Expt}^{\mathsf{DA}}_{\Pi,\mathcal{E}\leftrightarrow\mathsf{R}}$. That is,* R *plays the role of the challenger $\mathcal{C}$ in $\mathsf{Exp}^{\mathsf{d\text{-}goal\text{-}atk}}_{\mathcal{A},\Sigma}$ and that of the adversary in $\mathsf{Expt}^{\mathsf{DA}}_{\Pi,\mathcal{E}\leftrightarrow\mathsf{R}}$.*

1. *$\mathcal{A}$ declares an arbitrary string $t^* \in \mathcal{T}$ at the very begnning of the game and send t^* to* R. *(We can allow* R *to determine t^* in some security games.)*
2. R *is given a problem instance π of the hard problem Π.*
3. R *simulates public parameters* PP *of Σ by using π and t^* and sends* PP *to $\mathcal{A}$.*
4. R *simulates an oracle $\mathcal{O}_{\mathsf{MSK}}$ of the security game of Σ when $\mathcal{A}$ sends oracle queries. That is, when $\mathcal{A}$ sends a query $\mathsf{query} \in \mathcal{Q}$,* R *simulates the value $\mathcal{O}_{\mathsf{MSK}}(\mathsf{query})$ and returns a simulated value $\mathsf{answer} \in \mathcal{P}$ to $\mathcal{A}$. If $\mathsf{Adml}(t^*, \mathsf{query}) = \bot$, then* R *outputs $\bot$.*
   At the oracle simulation phase, R never interacts *with $\mathcal{E}$.*

5. *At some point,* $\mathcal{A}$ *sends a challenge query* $\mathsf{challenge} \in \mathcal{H}$ *to* R.
6. R *chooses* $\mathsf{coin} \leftarrow \{0,1\}$ *and simulates a challenge answer* $\mathsf{c\text{-}ans}^* \in \mathcal{I}$ *of* $\mathcal{C}_{\mathsf{a}}(\mathsf{PP}, t^*, \mathsf{challenge}, b)$ *by using* $(\pi, \mathsf{PP}, t^*, \mathsf{challenge}, \mathsf{coin})$. *It sends* $\mathsf{c\text{-}ans}^*$ *to* $\mathcal{A}$. R *is allowed to interact with* $\mathcal{E}$ *at this phase.*
7. *We can allow* $\mathcal{A}$ *to send queries to* $\mathcal{O}_{\mathsf{MSK}}$ *again. At some point,* $\mathcal{A}$ *outputs* coin^*.
8. *Finally,* R *outputs a bit* $\mathsf{sol} := 0$ *if* $\mathsf{coin} = \mathsf{coin}^*$. *Otherwise* ($\mathsf{coin} \neq \mathsf{coin}^*$), *outputs* $\mathsf{sol} := 1$.

R *consists of three algorithms* $(\mathsf{PSim}, \mathsf{OSim}, \mathsf{CSim})$ *introduced below.*

All-but-one oracle simulation: R *can perfectly simulate the public parameter of* Σ *and the oracle* $\mathcal{O}_{\mathsf{MSK}}$. *That is, there exist parameter and oracle simulation algorithms* PSim *and* OSim *such that for all* $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{PGen}(1^\lambda)$, $b \in \{0,1\}$, $\pi \leftarrow \mathsf{PSample}_b(1^\lambda)$, $t^* \in \mathcal{T}$, *and* $\mathsf{query} \in \mathcal{Q}$ *where* $\mathsf{AdmI}(t^*, \mathsf{query}) = \top$, *it holds that*

$$\mathsf{PSim}(\pi, t^*; \rho) \overset{\mathsf{p}}{\approx} \mathsf{PP},$$

$$\mathsf{OSim}(\pi, \rho, t^*, \mathsf{query}) \overset{\mathsf{p}}{\approx} \mathcal{O}_{\mathsf{MSK}}(\mathsf{query}),$$

where ρ *is the randomness of* PSim. *Note that a query* query *such that* $\mathsf{AdmI}(t^*, \mathsf{query}) = \bot$ *is not allowed in the selective security game of* Σ. *In particular,* OSim
- *is described as a stateless randomized algorithm.*
- *does not have any oracle access.*

Challenge simulation *Let* ρ *be the randomness used by* PSim. R *does all the steps from (1) to (5) in the selective reduction above and can simulate the challenge answer for the challenge query from* $\mathcal{A}$. *That is, there exists a challenge simulation algorithm* CSim *such that in the selective game above, if* $\pi_0 \leftarrow \mathsf{PSample}_0(1^\lambda)$, *then* R *perfectly simulates* $\mathsf{Exp}^{\mathsf{d\text{-}goal\text{-}atk}}_{\mathcal{A},\Sigma}(\lambda, \mathsf{coin})$ *and it holds that*

$$\mathsf{CSim}(\pi_0, \rho, t^*, \mathsf{challenge}, \mathsf{coin}) \overset{\mathsf{p}}{\approx} \mathcal{C}_{\mathsf{a}}(\mathsf{PP}, t^*, \mathsf{challenge}, \mathsf{coin}).$$

In addition, if $\pi_1 \leftarrow \mathsf{PSample}_1(1^\lambda)$, *then the output of* $\mathsf{CSim}(\pi_1, \rho, t^*, \mathsf{challenge}, \mathsf{coin})$ *is a valid challenge answer, but independent of* coin *and* $\Pr[\mathsf{coin} = \mathsf{coin}^*] = \frac{1}{2}$. *This property immediately implies*

$$\mathsf{Adv}^{\mathsf{DA}}_{\Pi,\mathcal{E}\leftrightarrow\mathsf{R}}(\lambda) \geq \frac{1}{2}\mathsf{Adv}^{\mathsf{d\text{-}goal\text{-}atk}}_{\mathcal{A},\Sigma}(\lambda).$$

Due to space limitations, we omit the proof.

Answer checkability: *There exists an efficient validity check algorithm* Valid *for* $\mathcal{Q}$ *such that for all* $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{PGen}(1^\lambda)$, $\mathsf{query} \leftarrow \mathcal{Q}$, $\mathsf{answer} \leftarrow \mathcal{O}_{\mathsf{MSK}}(\mathsf{query})$,

$$\Pr[\mathsf{Valid}(\mathsf{PP}, \mathsf{query}, \mathsf{answer}) = \top] = 1 - \mathsf{negl}(\lambda).$$

On the other hand, for all $b \in \{0,1\}$, $\pi \leftarrow \mathsf{PSample}_b(1^\lambda)$, $t^* \in \mathcal{T}$, $\mathsf{PP} \leftarrow \mathsf{PSim}(\pi, t^*; \rho)$, query *such that* $\mathsf{AdmI}(t^*, \mathsf{query}) = \bot$,

$$\Pr[\mathsf{Valid}(\mathsf{PP}, \mathsf{query}, \mathsf{OSim}(\pi, \rho, t^*, \mathsf{query})) = \top] \leq \mathsf{negl}(\lambda).$$

Attack substitution: R *can solve a problem* π *if we have a valid answer* $\mathsf{answer}^* \in \mathcal{P}$ *for* $\mathsf{query}^* \in \mathcal{Q}$ *such that* $\mathsf{AdmI}(t^*, \mathsf{query}^*) = \bot$ *(i.e., inadmissible query) instead of a successful adversary* $\mathcal{A}$ *in the selective reduction. That is, there exists an efficient algorithm* Solve *such that for all* $b \in \{0,1\}$*,* $\pi \leftarrow \mathsf{PSample}_b(1^\lambda)$*,* $t^* \in \mathcal{T}$*,* $\mathsf{query}^* \in \mathcal{Q}$*,* $\mathsf{answer}^* \in \mathcal{P}$ *such that* $\mathsf{Valid}(\mathsf{PP}, \mathsf{query}^*, \mathsf{answer}^*) = \top$ *and* $\mathsf{AdmI}(t^*, \mathsf{query}^*) = \bot$*, we have that* $\mathsf{Solve}(\pi, \rho, t^*, \mathsf{query}^*, \mathsf{answer}^*)$ *outputs* sol *for* π *and*

$$\mathsf{Adv}^{\mathsf{DA}}_{\Pi,\mathcal{E}\leftrightarrow\mathsf{R}}(\lambda) > \mathsf{negl}(\lambda),$$

where ρ *is the randomnesses to sample* PP *in the selective reduction.*

Problem instance simulation: *We can perfectly simulate a problem instance and randomness used to generate* PP *in* PSim *if we have a master secret key of* Σ*. That is, there exists an efficient algorithm* MSKtoP *such that for all* $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{PGen}(1^\lambda)$*,* $\pi \leftarrow \mathsf{PSample}_0(1^\lambda)$*, all* $\rho \leftarrow \mathcal{R}_{\mathsf{PSim}}$*, and all* $t^* \in \mathcal{T}$*,*

$$(\pi', \rho', \mathsf{PP}) \overset{\mathsf{p}}{\approx} (\pi, \rho, \mathsf{PP}'),$$

where $(\pi', \rho') \leftarrow \mathsf{MSKtoP}(1^\lambda, \mathsf{MSK}, t^*)$*,* $\mathsf{PP}' = \mathsf{PSim}(\pi, t^*; \rho)$*,* ρ' *is a randomness to simulate* PP *via* PSim*, and* $\mathcal{R}_{\mathsf{PSim}}$ *is the randomness space of* PSim*. We can relax this condition to statistical indistinguishability for uniformly random* t^* *(instead of all* $t^* \in \mathcal{T}$*).*

On canonical property. As we can see in concrete examples (not only) in Sections 4.3 and 4.5 (but also in many works), well-known selectively secure schemes have canonical ABO reductions. If a scheme has a reduction that must interact with the challenger in an assumption to simulate $\mathcal{O}_{\mathsf{MSK}}$, then the reduction is not canonical. Interestingly, even if a reduction is allowed to interact with the challenger, the reduction could be canonical *as long as the reduction does not need the interaction for simulating* $\mathcal{O}_{\mathsf{MSK}}$. More specifically, a canonical reduction is allowed to interact with the challenger in the assumption to *simulate a challenge answer*. See the full version for such an example.

Due to space limtations, we omit the general definition of canonical ABO reductions that also captures the TBE case.

Table 4 shows concrete example of spaces and oracles for various cryptographic primitives.

On validity check algorithm. The validity check algorithm in Def. 4.3 verifies that a value in $\mathcal{P}$ is a correct value for input $\mathsf{query} \in \mathcal{Q}$. Let $\rho_{\mathsf{mka}} \leftarrow \mathcal{R}_{\mathsf{mka}}$ and $\mathsf{answer} = C(\mathsf{query}, \rho_{\mathsf{mka}})$. Then, Valid is described as follows.

$$\mathsf{Valid}(\mathsf{PP}, \mathsf{query}, \rho_{\mathsf{q}}, \mathsf{answer}) := \mathsf{Valid\text{-}Out}(\mathsf{PP}, \mathsf{str}, C(\mathsf{str}, \rho_{\mathsf{mka}})) \quad \text{SIG/IBE/ABE}$$

4.3 Concrete Examples

First, we list the references of well-known schemes that fall into the class of canonical ABO reductions [13,9,49,37,30,11,17,6,2,53,3,40,25,10,26]. Note that this is not the exhaustive list.

Table 4: Concrete sets, oracle, and admissible condition of ABO reductions for encryption.

ABO reduction	tag-based PKE	IBE	KP-ABE
$\mathcal{T}$	tag space $\mathcal{TAG}$	identity space $\mathcal{ID}$	attribute space $\mathcal{ATT}$
$\mathcal{Q}$	tag space $\mathcal{TAG}$	identity space $\mathcal{ID}$	policy space $\mathcal{POL}$
$\mathcal{P}$	plaintext space $\mathcal{PT} \cup \{\perp\}$	secret key space $\mathcal{SK}$	secret key space $\mathcal{SK}$
$\mathcal{H}$	plaintext space $\mathcal{PT}^2$	plaintext space $\mathcal{PT}^2$	plaintext space $\mathcal{PT}^2$
$\mathcal{I}$	$\mathcal{TAG} \times \mathcal{CT}$	$\mathcal{CT}$	$\mathcal{CT}$
$\mathcal{O}_{\mathsf{MSK}}$	dec oracle $\mathsf{Dec}(\mathsf{dk}, \cdot)$	key oracle $\mathsf{KeyGen}(\mathsf{MSK}, \cdot)$	key oracle $\mathsf{KeyGen}(\mathsf{MSK}, \cdot)$
$\mathsf{Adml}(\cdot,\cdot) = \top$	$t^* \neq t$	$t^* \neq \mathsf{id}$	$\mathsf{P}(t^*) = \perp$

Next, we present concrete examples by picking up well-known selectively secure schemes. We often omit parameters if it is clear from the context.

Example 4.1 (Boneh-Boyen IBE). The Boneh-Boyen IBE scheme BB consists of the following algorithms.

$\mathsf{Setup}(1^\lambda)$:
- Generate $\mathsf{params} := (p, \mathbb{G}, \mathbb{G}_T, e, G) \leftarrow \mathcal{G}_{\mathsf{bmp}}(1^\lambda)$.
- Choose $x, y \leftarrow \mathbb{Z}_p$ and $h \leftarrow \mathbb{Z}_p$ and set $G_1 := G^x, G_2 := G^y, H := G^h$.
- Output $\mathsf{MPK} := (\mathsf{params}, G, G_1, G_2, H)$ and $\mathsf{MSK} := (\mathsf{MPK}, x, y, h)$.

$\mathsf{KeyGen}(\mathsf{MSK}, \mathsf{id})$:
- For $\mathsf{id} \in \mathbb{Z}_p$, choose $r \leftarrow \mathbb{Z}_p$ and output $\mathsf{SK}_{\mathsf{id}} := (G_2^x (G_1^{\mathsf{id}} \cdot H)^r, G^r)$.

$\mathsf{Enc}(\mathsf{MPK}, m)$:
- For $M \in \mathbb{G}_T$, choose $s \leftarrow \mathbb{Z}_p$ and output $\mathsf{CT} := (e(G_1, G_2)^s \cdot M, G^s, (G_1^{\mathsf{id}} \cdot H)^s)$.

$\mathsf{Dec}(\mathsf{SK}_{\mathsf{id}}, \mathsf{CT})$:
- Parse $\mathsf{sk}_{\mathsf{id}} = (D_1, D_2)$ and $\mathsf{CT} = (C_0, C_1, C_2)$, output $C_0 \cdot e(C_2, D_2) \cdot e(C_1, D_1)^{-1}$.

The reduction algorithm R of BB IBE scheme consists of three algorithms $(\mathsf{PSim}, \mathsf{OSim}, \mathsf{CSim})$. Below, we let $\pi := (G, G^x, G^y, G^z, T)$, $t^* := \mathsf{id}^*$, $\mathsf{query} := \mathsf{id}_i$, $\overline{\mathsf{query}} := \perp$, $\rho_{\mathsf{q}} := \perp$, $\mathsf{challenge} := (M_0, M_1)$, be a DBDH instance, the target identity, a query to the key generation oracle, a sub-query, the randomness to sample $\overline{\mathsf{query}} \in \overline{\mathcal{Q}}_{\mathsf{aux}}$, the challenge messages, respectively.

$\mathsf{PSim}(\pi, t^*)$**:** This algorithm is given a DBDH instance π and a target identity $t^* = \mathsf{id}^*$ and simulate MPK. It chooses $\beta \leftarrow \mathbb{Z}_p$, sets $G_1 := G^x$, $G_2 := G^y$, and $H := G_1^{-\mathsf{id}^*} \cdot G^\beta$, and outputs $\mathsf{MPK} := (G, G_1, G_2, H)$. The randomness ρ of this algorithm is $\rho := \beta$

$\mathsf{OSim}(\pi, \rho, t^*, \mathsf{query})$**:** This algorithms simulate secret keys for identity $\mathsf{query} = \mathsf{id}_i \in \mathbb{Z}_p$ such that $\mathsf{id}_i \neq \mathsf{id}^* = t^*$. It parses $\rho = \beta$, chooses $r \leftarrow \mathbb{Z}_p$ and outputs $\mathsf{SK}_{\mathsf{id}_i} = (D_1, D_2)$ where

$$D_1 := G_2^{\frac{-\beta}{\mathsf{id}_i - \mathsf{id}^*}} (G_1^{\mathsf{id}_i} H)^r, D_2 := G_2^{\frac{-1}{\mathsf{id}_i - \mathsf{id}^*}} G^r.$$

The randomness ρ_{o} of this algorithm is $\rho_{\mathsf{o}} = r$.

$\mathsf{CSim}(\pi, \rho, t^*, \mathsf{challenge}, \mathsf{coin})$**:** This algorithms simulate a challenge ciphertext for $\mathsf{challenge} = (M_0, M_1)$ under identity $t^* = \mathsf{id}^*$. It parses $\rho = \beta$ and outputs

$$\mathsf{CT}^* \coloneqq (M_{\mathsf{coin}} \cdot T, G^z, (G^z)^\beta).$$

The auxiliary ABO reduction algorithms of BB IBE scheme consists of three algorithms $(\mathsf{Valid}, \mathsf{Solve}, \mathsf{MSKtoP})$.

$\mathsf{Valid}(\mathsf{MPK}, \mathsf{query}, \rho_{\mathsf{q}}, \mathsf{answer})$**:** This algorithm parses $\mathsf{MPK} = (G, G_1, G_2, H)$, $\mathsf{query} = (\mathsf{id}, \bot)$, $\rho_{\mathsf{q}} = \bot$, and $\mathsf{answer} = (D_1, D_2)$ (this is secret key $\mathsf{SK}_{\mathsf{id}}$ for identity id) and checks

$$e(G, D_1) = e(G_1, G_2) \cdot e(G_1^{\mathsf{id}} H, D_2). \tag{1}$$

If it holds, then output $\top$. Otherwise, outputs $\bot$.

$\mathsf{Solve}(\pi, \rho, t^*, \mathsf{query}^*, \rho_{\mathsf{q}}, \mathsf{answer}^*)$**:** First, this algorithm parses $\mathsf{id}^* = t^*$, $\mathsf{query}^* = (\mathsf{id}^*, \bot)$, $\rho = \beta$, and $\rho_{\mathsf{q}} = \bot$. It chooses M_0, M_1 and $\mathsf{coin} \leftarrow \{0,1\}$ and computes

$$\mathsf{CT}^* \coloneqq (M_{\mathsf{coin}} \cdot T, G^z, (G^z)^\beta).$$

(this is the same as the output of $\mathsf{CSim}(\pi, \rho, t^*, \mathsf{challenge}, \mathsf{coin})$). Then, it parses $\mathsf{answer}^* = (G_2^x (G_1^{\mathsf{id}^*} H)^r, G^r)$ and decrypts CT^* by using $(G_2^x (G_1^{\mathsf{id}^*} H)^r, G^r)$. If it obtains M_{coin}, then outputs 0, otherwise 1.

$\mathsf{MSKtoP}(1^\lambda, \mathsf{MSK}, t^*)$**:** First, this algorithms parses $\mathsf{MSK} = (\mathsf{MPK}, x, y, h)$, chooses $z \leftarrow \mathbb{Z}_p$, and computes $\beta \coloneqq x \cdot \mathsf{id}^* + h$. Then, it outputs $\pi \coloneqq (G, G^x, G^y, G^z, e(G,G)^{xyz})$ and $\rho' \coloneqq \beta = x \cdot \mathsf{id}^* + h$.

Theorem 4.1. *Boneh-Boyen IBE scheme has a canocanil ABO reduction to the DBDH problem.*

Due to space limitations, we omit the proof.

4.4 All-But-N Reductions

We can extend canonical ABO reductions to canonical all-but-N (ABN) reductions. Here, N is an a-priori bounded/unbounded polynomial of the security parameter. Roughly speaking, a canonical ABN reduction punctures N points $\vec{t^*} = (t_1^*, \ldots, t_N^*) \in \mathcal{T}^N$ in a master secret-key based algorithm MSKAlg instead of a single point t^*.

We omit the definition due to space limitations. Basically, we simply replace a signle point t^* with N points $\vec{t^*} = (t_1^*, \ldots, t_N^*)$ and require $\mathsf{Adml}(t_i^*, \mathsf{query}) = \top$ for all $i \in [N]$ for admissible queries. See the full version for details.

4.5 Concrete Examples of canonical ABN Reductions

It is easy to extend ABO reductions to ABN reductions for pairing-based schemes by using (weak) programmable hash functions [33,32]. Due to space limitations, we omit details. We can obtain the modified Boneh-Boyen IBE scheme, which has a canonical all-but-N reduction, by using programmable hash $\mathsf{H}_{\mathsf{w}}(X) \coloneqq \prod_{i=0}^{n} H_i^{X^i}$ where the hash key is $(H_0, H_1, \ldots, H_N)$ instead of the Boneh-Boyen hash function $\mathsf{H}_{\mathsf{BB}}(X) \coloneqq G_1^X H$ where the hash key is (G_1, H).

The rough idea is as follows. The ABN reduction is given a DBDH instance $\pi = (G, G^x, G^y, G^z, T)$ and target identities $\vec{t}^* = \vec{\mathsf{id}}^* = (\mathsf{id}_1^*, \ldots, \mathsf{id}_N^*)$, and simulates MPK. It chooses $\mathsf{id}_0^* \leftarrow \mathbb{Z}_p$ and $(\beta_0, \ldots, \beta_N) \leftarrow \mathbb{Z}_p^{N+1}$, and computes $(\alpha_0, \ldots, \alpha_N)$ such that $\sum_{i=0}^{N} \alpha_i \cdot t^i = \prod_{i=0}^{N}(t - \mathsf{id}_i^*) \in \mathbb{Z}_p[t]$. Then, it sets $G_1 \coloneqq G^x$, $G_2 \coloneqq G^y$, and $H_i \coloneqq G_1^{\alpha_i} \cdot G^{\beta_i}$, and outputs $\mathsf{MPK} \coloneqq (G, G^x, G_2, H_0, \ldots, H_N)$. By this parameter setting, we can implement canonical ABN reductions in a simlar way to the ABO reduction of Boneh-Boyen IBE. See the full version for detail.

5 Message-Less Watermarking via Canonical ABO-reductions

In this section, we present a message-less watermarking scheme from all-but-one reductions. We focus on using canonical ABO reductions for the decisional case. It is easy to adapt that for the computational case, so we omit it.

First, we present our watermarking scheme $\mathsf{WM}_\Sigma = (\mathsf{WMSetup}, \mathsf{Mark}, \mathsf{Extract})$ for Σ. Let MSK be a master secret-key generated by the setup algorithm of Σ. WM_Σ is a public mark and public extraction scheme. Thus, we do not need watermarking secret-key wsk.

$\mathsf{WMSetup}(1^\lambda)$:
- Choose $t^* \leftarrow \mathcal{T}$ and output $\mathsf{wpp} \coloneqq t^*$.

$\mathsf{Mark}(\mathsf{wpp}, \mathsf{MSK})$:
- Read MSK and generate $(\pi', \rho') \leftarrow \mathsf{MSKtoP}(1^\lambda, \mathsf{MSK}, t^*)$.
- Generate a circuit $\widetilde{f_\Sigma}[\pi', \rho', t^*]$ described in Fig. 3.

$\mathsf{Extract}(\mathsf{wpp}, \mathsf{PP}, C')$:
- Choose $\mathsf{query} \leftarrow \mathcal{Q}$ such that $\mathsf{AdmI}(t^*, \mathsf{query}) = \top$.
- Sample $\rho_{\mathsf{o}} \leftarrow \mathcal{R}_{\mathsf{mka}}$ and compute $\mathsf{answer} \leftarrow C'(\mathsf{query}, \rho_{\mathsf{o}})$.
- Check $\mathsf{Valid}(\mathsf{PP}, \mathsf{query}, \mathsf{answer}) \overset{?}{=} \top$. If the equation holds, then output $\mathsf{unmarked}$. Otherwise, marked.

Marked master secret-key $\widetilde{f_\Sigma}[\pi', \rho', t^*]$

Hardwired: π', ρ', t^*.
Input: An input $\mathsf{query} \in \mathcal{Q}$ to MSKAlg and randomness $\rho_{\mathsf{o}} \in \mathcal{R}_{\mathsf{mka}}$.
Procedure: Compute and output $\mathsf{answer} \leftarrow \mathsf{OSim}(\pi', \rho', t^*, \mathsf{query}; \rho_{\mathsf{o}})$.

Fig. 3: The description of $\widetilde{f_\Sigma}$

Remark 5.1. Even a useless circuit that outputs $\perp$ for all inputs is marked in the watermarking scheme above since $\mathsf{Valid}(\mathsf{PP}, \mathsf{query}, \rho_{\mathsf{q}}, \perp) = \perp$ for any PP, query, and ρ_{q}. To prevent this trivial watermarking, we need to check whether a circuit is similar to a master secret-key based algorithm whose corresponding master public parameter is PP. Although we omit this checking procedure for simplicity here (our final goal is achieving message-embedding schemes), we present test algorithms for this check in Sec. 6.

Theorem 5.1. *Let Σ be a master secret-key based scheme with $(\mathcal{T}, \mathcal{Q}, \mathcal{P}, \mathcal{R}_{\mathsf{mka}})$ associated with sub-query space $\mathcal{Q}_{\mathsf{t}}$, aux-query space $\mathcal{Q}_{\mathsf{aux}}$, challenge space $\mathcal{H}$, challenge answer space $\mathcal{I}$, and admissible condition Adml. If Σ has a canonical all-but-one reduction to a hard problem Π, then there exists a message-less watermarking scheme WM_{Σ} for master secret-keys of Σ and WM satisfies Def. 3.5 with parameter $\epsilon = 1/\mathrm{poly}(\lambda)$ under the assumption that Π is hard.*

The intuition of security is that adversaries cannot recover the functionality of $\mathsf{MSKAlg}(\mathsf{MSK}, \cdot)$ for input t^* from the oracle simulation algorithm OSim since OSim is punctured at t^* (explained in Sec. 1.3). Due to space limitations, we omit the proof.

6 Message-Embedding Watermarking via Canonical ABN-reductions

In this section, we present a message-embedding watermarking scheme from canonical all-but-N reductions.

6.1 How to Test Circuit Similarity

Before we describe our message-embedding watermarking scheme, we present how to test a circuit is similar to the original circuit to be watermarked.

Test circuits by master public parameters. We define test algorithms Test described in Fig. 4 to verify that a circuit C' is close to a master secret-key based algorithm whose master secret key is MSK that corresponds to a master public parameter PP. We have two versions of Test since there are a few differences between one for signature/IBE/ABE/IPE/PE and one for TBE. However, we omit that of TBE due to space limitations. We set parameters $0 < \epsilon_1 < \epsilon_2 < 1/2$ where $\epsilon_2 - \epsilon_1 > 1/\mathrm{poly}(\lambda)$.

Theorem 6.1. *Assume that $0 < \epsilon_1 < \epsilon_2 < 1/2$ where $\epsilon_2 - \epsilon_1 > 1/\mathrm{poly}(\lambda)$. For all $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{PGen}(1^{\lambda})$,*

- *For all $C'(\cdot,\cdot) \cong_{\epsilon_1} \mathsf{MSKAlg}(\mathsf{MSK}, \cdot; \cdot)$, $\Pr[\mathsf{Test}(\mathsf{PP}, C') = \top] \geq 1 - \mathsf{negl}(\lambda)$.*
- *For all $C'(\cdot,\cdot) \not\cong_{\epsilon_2} \mathsf{MSKAlg}(\mathsf{MSK}, \cdot; \cdot)$, $\Pr[\mathsf{Test}(\mathsf{PP}, C') = \top] \leq \mathsf{negl}(\lambda)$.*

We omit the proof due to space limitations.

By the theorem, we can verify whether $C'(\cdot,\cdot) \cong_{\epsilon_1} \mathsf{MSKAlg}(\mathsf{MSK}, \cdot; \cdot)$ or not if $\epsilon_1 = 1/2 - 1/\mathrm{poly}(\lambda)$. That is, if the adversary $\mathcal{A}$ in ϵ-unremovability game is ϵ-admissible where $\epsilon = 1/2 + 1/\mathrm{poly}(\lambda)$, then the circuit C^* output by $\mathcal{A}$ passes the test.

```
Inputs: A public parameter $\mathsf{PP}$ and a circuit $C'$.
Parameters: $\delta \coloneqq (\epsilon_2 - \epsilon_1)/2$, $S \coloneqq \lambda/\delta^2$, $\epsilon \coloneqq (\epsilon_1 + \epsilon_2)/2$.

Set $\mathsf{cnt} \coloneqq 0$. For $i = 1, \ldots, S$, do

  1. Choose $z_i \leftarrow \mathcal{Q}$ and $\rho_i \leftarrow \mathcal{R}_{\mathsf{mka}}$.
  2. If $\mathsf{Valid\text{-}Out}(\mathsf{PP}, z_i, C'(z_i, \rho_i)) = \bot$, then sets $\mathsf{cnt} \coloneqq \mathsf{cnt} + 1$.

If $\mathsf{cnt} \leq \epsilon S$, then output $\top$. Otherwise $\bot$.
```

Fig. 4: Test algorithm Test for IBE or signature

6.2 Message-Embedding Scheme

We present our message-embedding watermarking scheme $\mathsf{msWM}_{\Sigma} = (\mathsf{WMSetup}, \mathsf{Mark}, \mathsf{Extract})$ for Σ. We consider none of ABE, IPE, and PE for the message-embedding scheme since we do not have (canonical) ABN reductions of them. Thus, $\mathcal{T} = \mathcal{Q}_{\mathsf{t}}$ in the rest of this section. Note that we implicitly assume that the master secret key MSK of Σ includes the corresponding public parameter PP. We use a PRF $(\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$ such that $\mathsf{PRF.Eval}(\mathsf{K}, \cdot) : \{0,1\}^{|\mathsf{PP}|} \times [\ell] \times \{0,1\} \rightarrow \mathcal{T}^T$. We show only for the decisional case, but it is easy to adapt to the computational case.

$\mathsf{WMSetup}(1^\lambda)$**:**
- Let $T \coloneqq \lambda$.
- Generate $\mathsf{K} \leftarrow \mathsf{PRF.Gen}(1^\lambda)$ and set $\mathsf{wpp} \coloneqq \bot$ and $\mathsf{wsk} \coloneqq \mathsf{K}$. We omit wpp hereafter since it is $\bot$.

$\mathsf{Mark}(\mathsf{wsk}, \mathsf{MSK}, \omega)$**:**
- Compute $\boldsymbol{t}_i = (t_i^{(1)}, \ldots, t_i^{(T)}) \leftarrow \mathsf{PRF.Eval}(\mathsf{K}, (\mathsf{PP}, i, \omega_i))$ for $i \in [\ell]$ and set $\boldsymbol{t}_\omega \coloneqq \{\boldsymbol{t}_i\}_{i \in [\ell]}$.
- Read MSK and generate $(\pi', \rho') \leftarrow \mathsf{MSKtoP}(1^\lambda, \mathsf{MSK}, \boldsymbol{t}_\omega)$.
- Generate a circuit $\widetilde{f}_{\Sigma}[\pi', \rho', \boldsymbol{t}_\omega]$ described in Fig. 5.

$\mathsf{Extract}(\mathsf{wsk}, \mathsf{PP}, C')$**:**
- Compute $b_{\mathsf{PP}} \leftarrow \mathsf{Test}(\mathsf{PP}, C')$. If $b_{\mathsf{PP}} = \bot$, then output `Invalid-Key` and halt. Otherwise, do the following steps.
- Compute $\widetilde{\boldsymbol{t}}_{i,b} = (\widetilde{t}_{i,b}^{(1)}, \ldots, \widetilde{t}_{i,b}^{(T)}) \leftarrow \mathsf{PRF.Eval}(\mathsf{K}, (\mathsf{PP}, i, b))$ for $i \in [\ell]$ and $b \in \{0,1\}$.
- For $i \in [\ell], b \in \{0,1\}$, set $\mathsf{query}_{i,b}^{(j)} \coloneqq \widetilde{t}_{i,b}^{(j)}$, compute $\mathsf{answer}_{i,b}^{(j)} \leftarrow C'(\mathsf{query}_{i,b}^{(j)}, \rho_{\mathsf{o},j})$. Let $\widehat{N}_{i,b}$ be the number of indices $j \in [T]$ such that $\mathsf{Valid}(\mathsf{PP}, \mathsf{query}_{i,b}^{(j)}, \mathsf{answer}_{i,b}^{(j)}) = \bot$.
  - If there exists an index $i \in [\ell]$ where $\widehat{N}_{i,0}, \widehat{N}_{i,1} < T$ or $\widehat{N}_{i,0} = \widehat{N}_{i,1} = T$, then output $\bot$.
  - Otherwise, for each $i \in [\ell]$, let $\omega'_i \in \{0,1\}$ be the unique bit where $\widehat{N}_{i,\omega'_i} = T \wedge \widehat{N}_{i,1-\omega'_i} < T$ and output $\omega' \coloneqq \omega'_1 \ldots \omega'_\ell$.

Marked master secret-key $\widetilde{f_\Sigma}[\pi', \rho', \boldsymbol{t}_\omega]$

Hardwired: π', ρ', $\boldsymbol{t}_\omega$.
Input: An input $\mathsf{query} \in \mathcal{Q}$ to MSKAlg and randomness $\rho_\mathsf{o} \in \mathcal{R}_\mathsf{mka}$.
Procedure: Compute and output $\mathsf{answer} \leftarrow \mathsf{OSim}(\pi', \rho', \boldsymbol{t}_\omega, \mathsf{query}; \rho_\mathsf{o})$.

Fig. 5: The description of $\widetilde{f_\Sigma}$

Theorem 6.2. *Let Σ be a master secret-key based scheme with $(\mathcal{T}, \mathcal{Q}, \mathcal{P}, \mathcal{R}_\mathsf{mka})$ associated with challenge space $\mathcal{H}$, challenge answer space $\mathcal{I}$, and admissible condition* AdmI. *If Σ has a canonical all-but-N reduction to a hard problem Π and* PRF *is a PRF where $N = \ell\lambda$, then there exists a message-embedding watermarking scheme* msWM_Σ *for master secret keys of Σ and* msWM_Σ *satisfies Def. 3.5 with parameter $\epsilon = 1/2 + 1/\mathrm{poly}(\lambda)$ under the assumption that Π is hard.*

Due to space limitations, we omit the proof.

Acknowledgments

The author would like to thank Fuyuki Kitagawa for valuable discussion and insightful comments on watermarking. The author also thanks Shuichi Katsumata and Shota Yamada for answering questions about lattices and programmable hash functions, and TCC 2020 reviewers for very constructive comments on the presentation.

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. *EUROCRYPT 2010*.
2. S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, and H. Wee. Functional encryption for threshold functions (or fuzzy ibe) from lattices. *PKC 2012*.
3. S. Agrawal, D. M. Freeman, and V. Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. *ASIACRYPT 2011*.
4. P. Ananth and V. Vaikuntanathan. Optimal bounded-collusion secure functional encryption. *TCC 2019, Part I*.
5. N. Attrapadung, G. Hanaoka, and S. Yamada. New security proof for the Boneh-Boyen IBE: Tight reduction in unbounded multi-challenge security. *ICICS 14*.
6. Nuttapong Attrapadung and Benoît Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. *PKC 2010*.
7. F. Baldimtsi, A. Kiayias, and K. Samari. Watermarking public-key cryptographic functionalities and implementations. *ISC 2017*.
8. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. *J.ACM*, 59(2):6, 2012.

9. D. Boneh and X. Boyen. Efficient selective identity-based encryption without random oracles. *Journal of Cryptology*, 24(4):659–693, 2011.
10. D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. *EUROCRYPT 2014*.
11. D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption schemes. *ASIACRYPT 2008*.
12. D. Boneh, K. Lewi, and D. J. Wu. Constraining pseudorandom functions privately. *PKC 2017*.
13. X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. *ACM CCS 2005*.
14. D. Boneh, A. Sahai, and B. Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. *EUROCRYPT 2006*.
15. Z. Brakerski and V. Vaikuntanathan. Constrained key-homomorphic PRFs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. *TCC 2015*.
16. Z. Brakerski and V. Vaikuntanathan. Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. *CRYPTO 2016*.
17. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of Cryptology*, 25(4):601–639, 2012.
18. J. Chen, H. Lim, S. Ling, H. Wang, and H. Wee. Shorter identity-based encryption via asymmetric pairings. *Des. Codes Cryptogr.*, 73(3):911–947.
19. B. Chor, A. Fiat, and M. Naor. Tracing traitors. *CRYPTO'94*.
20. A. Cohen, J. Holmgren, R. Nishimaki, V. Vaikuntanathan, and D. Wichs. Watermarking cryptographic capabilities. Cryptology ePrint Archive, Report 2015/1096.
21. A. Cohen, J. Holmgren, R. Nishimaki, V. Vaikuntanathan, and D. Wichs. Watermarking cryptographic capabilities. *SIAM J. Computing*, 47(6):2157–2202, 2018.
22. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.*, 33(1):167–226, 2003.
23. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. *40th ACM STOC*, 2008.
24. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.
25. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. *Journal of the ACM*, 62(6):45:1–45:33, 2015.
26. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate encryption for circuits from LWE. *CRYPTO 2015*.
27. R. Goyal, S. Kim, N. Manohar, B. Waters, and D. J. Wu. Watermarking public-key cryptographic primitives. *CRYPTO 2019*.
28. R. Goyal, V. Koppula, and B. Waters. Collusion resistant traitor tracing from learning with errors. *50th ACM STOC*, 2018.
29. R. Goyal, V. Koppula, and B. Waters. New approaches to traitor tracing with embedded identities. *TCC 2019, Part II*.
30. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. *ACM CCS 2006*
31. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. *IACR Cryptology ePrint Archive*, 2006:309, 2006. Version 20061007:061901.
32. D. Hofheinz, T. Jager, and E. Kiltz. Short signatures from weaker assumptions. *ASIACRYPT 2011*.

33. D. Hofheinz and E. Kiltz. Programmable hash functions and their applications. *Journal of Cryptology*, 25(3):484–527, 2012.
34. N. Hopper, D. Molnar, and D. Wagner. From weak to strong watermarking. *TCC 2007*.
35. S. Katsumata, R. Nishimaki, S. Yamada, and T. Yamakawa. Designated verifier/prover and preprocessing NIZKs from Diffie-Hellman assumptions. *EUROCRYPT 2019*.
36. S. Katsumata, R. Nishimaki, S. Yamada, and T. Yamakawa. Exploring constructions of compact NIZKs from various assumptions. *CRYPTO 2019*.
37. E. Kiltz. Chosen-ciphertext security from tag-based encryption. *TCC 2006*.
38. S. Kim and D. J. Wu. Watermarking cryptographic functionalities from standard lattice assumptions. *CRYPTO 2017*.
39. S. Kim and D. J. Wu. Watermarking PRFs from lattices: Stronger security via extractable PRFs. *CRYPTO 2019*.
40. K. Kurosawa and L. Phong. Leakage resilient IBE and IPE under the DLIN assumption. *ACNS 2013*.
41. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. *EUROCRYPT 2012*.
42. D. Naccache, A. Shamir, and J. P. Stern. How to copyright a function? *PKC'99*.
43. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. *22nd ACM STOC*, 1990.
44. R. Nishimaki. How to watermark cryptographic functions by bilinear maps. *IEICE Transactions*, 102-A(1):99–113, 2019.
45. R. Nishimaki, D. Wichs, and M. Zhandry. Anonymous traitor tracing: How to embed arbitrary information in a key. *EUROCRYPT 2016*.
46. W. Quach, D. Wichs, and G. Zirdelis. Watermarking PRFs under standard assumptions: Public marking and security with extraction queries. *TCC 2018*.
47. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34:1–34:40, 2009.
48. A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. *46th ACM STOC*, 2014.
49. A. Sahai and B. R. Waters. Fuzzy identity-based encryption. *EUROCRYPT 2005*.
50. C. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174.
51. R. Tsabary. Fully secure attribute-based encryption for t-CNF from LWE. *CRYPTO 2019*.
52. B. Waters. Efficient identity-based encryption without random oracles. *EUROCRYPT 2005*.
53. B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. *PKC 2011*.
54. S. Yamada. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. *CRYPTO 2017*.
55. R. Yang, M. H. Au, J. Lai, Q. Xu, and Z. Yu. Collusion resistant watermarking schemes for cryptographic functionalities. *ASIACRYPT 2019*.
56. M. Yoshida and T. Fujiwara. Toward digital watermarking for cryptographic data. *IEICE Transactions*, 94-A(1):270–272, 2011.