The 35th Symposium on Information Theory
and its Applications (SITA2012)
Beppu, Oita, Japan, Dec. 11–14, 2012

A secure instantiation of the random oracle by a computable function

Kohtaro Tadaki* Norihisa Doi*

Abstract— In this paper we investigate the problem of secure instantiation of the random oracle, based on the concepts and methods of algorithmic randomness. We show that, for any secure signature scheme in the random oracle model, there exists a specific computable function which can instantiate the random oracle while keeping the security originally proved in the random oracle model. Our results use the general form of definitions of security notions for signature schemes, and depend neither on specific schemes nor on specific security notions.

Keywords— cryptography, random oracle model, provable security, cryptographic hash function, algorithmic randomness

1 Introduction

In modern cryptography, *the random oracle model* is widely used as an *imaginary* framework in which the security of a cryptographic scheme is discussed. In the random oracle model, the cryptographic hash function used in a cryptographic scheme is formulated as a random variable uniformly distributed over all possibility of the function, called *the random oracle*, and the legitimate users and the adversary against the scheme are modeled so as to get the values of the hash function not by evaluating it in their own but by querying the random oracle [1]. Since the random oracle is an imaginary object, even if the security of a cryptographic scheme is proved in the random oracle model, the random oracle has to be instantiated using a concrete cryptographic hash function such as the SHA hash functions if we want to use the scheme in the real world. In fact, the instantiations of the random oracle by concrete cryptographic hash functions are widely used in modern cryptography to produce efficient cryptographic schemes. Once the random oracle is instantiated, however, the original security proof in the random oracle model is spoiled and goes back to square one. Actually, it is not clear how much the instantiation can maintain the security originally proved in the random oracle model, nor is it clear whether the random oracle can be instantiated somehow while keeping the original security.

In the present paper we investigate this problem, based on concepts and methods of *algorithmic randomness*, also known as *algorithmic information theory*. In algorithmic randomness, the notion of a *random real* plays a central role. It is an individual infinite binary sequence which is classified as "random", and not a random variable such as the random oracle. Algorithmic randomness enables us to classify an individual infinite binary sequence into random or not. It originated in the groundbreaking works of Solomonoff, Kolmogorov, and Chaitin in the mid-1960s. They independently introduced the notion of *program-size complexity*, also known as *Kolmogorov complexity*, in order to quantify the randomness of an individual object. In the 21st century, algorithmic randomness is making remarkable progress through close interaction with recursion theory [5, 3].

To summarize our contributions, we first review the security proof in the random oracle model (see e.g. Katz and Lindell [4, Chapter 13] for the detail). In the random oracle model, a cryptographic scheme Π relies on an oracle h which is a certain type of function mapping finite strings to finite strings, depending on a security parameter n. Let Hash_n denote the set of all such functions h on a security parameter n. Then the random oracle is the sequence $\{H_n\}$ of random variables such that each H_n is uniformly distributed over functions in Hash_n. Now, in order to introduce a security notion, such as CCA-security for encryption schemes and EUF-ACMA security for signature schemes, into the scheme Π, we first consider an appropriately designed experiment $\mathsf{Expt}_{\mathcal{A}^{H_n},\Pi^{H_n}}$ defined for Π and any adversary $\mathcal{A}$, where Π and $\mathcal{A}$ are both allowed to have an oracle access to H_n. Then a definition of security for Π in the random oracle model takes the following general form: the scheme Π is secure in the random oracle model if, for all probabilistic polynomial-time adversaries $\mathcal{A}$ and all $d \in \mathbb{N}^+$ there exists $N \in \mathbb{N}^+$ such that, for all $n \geq N$,

$$\Pr\left[\mathsf{Expt}_{\mathcal{A}^{H_n},\Pi^{H_n}}(n) = 1\right] \leq \gamma + \frac{1}{n^d}, \tag{1}$$

where the probability is taken over all the possible values assigned to H_n, i.e., all functions in Hash_n, as well as all possible internal coin tosses of the parties running Π and those of the adversary $\mathcal{A}$, with uniform probability distribution. The value γ indicates the maximum desired probability of some "bad" event (e.g., for encryption schemes $\gamma = 1/2$ and for signature schemes $\gamma = 0$). Since the random variable H_n is uniformly distributed over Hash_n for every n, the definition (1) of security in the random oracle model is rewritten into the following form: for all probabilistic polynomial-time adversaries $\mathcal{A}$ and all $d \in \mathbb{N}^+$ there exists $N \in \mathbb{N}^+$ such that, for all $n \geq N$,

$$\frac{\sum_{h \in \mathsf{Hash}_n} \Pr\left[\mathsf{Expt}_{\mathcal{A}^{H_n},\Pi^{H_n}}(n) = 1 \;\middle|\; H_n = h\right]}{\#\mathsf{Hash}_n} \leq \gamma + \frac{1}{n^d}, \tag{2}$$

where $\#\mathsf{Hash}_n$ denotes the number of functions in Hash_n, and the probability is now conditioned on that the ran-

* Research and Development Initiative, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan. E-mail: tadaki@kc.chuo-u.ac.jp, doi@doi.ics.keio.ac.jp WWW: http://www2.odn.ne.jp/tadaki/

dom variable H_n takes a specific function $h \in \mathsf{Hash}_n$ as its value.

Let $\{h_n\}$ be an arbitrary sequence of functions such that $h_n \in \mathsf{Hash}_n$ for all n. In this paper, we introduce the notion of *security of* Π *relative to a specific oracle* $\{h_n\}$, which can be formulated as follows: the scheme Π is secure relative to $\{h_n\}$ if, for all probabilistic polynomial-time adversaries $\mathcal{A}$ and all $d \in \mathbb{N}^+$ there exists $N \in \mathbb{N}^+$ such that, for all $n \geq N$,

$$\Pr\left[\mathsf{Expt}_{\mathcal{A}^{H_n},\Pi^{H_n}}(n) = 1 \;\middle|\; H_n = h_n\right] \leq \gamma + \frac{1}{n^d}. \quad (3)$$

The functions $\{h_n\}$ is an *instantiation* of the random oracle $\{H_n\}$. Note that, in the case where $\{h_n\}$ is polynomial-time computable, the condition (3) implies that the scheme Π is just *secure in the standard model*.

In our former work [6], we investigated the instantiation of the random oracle by a random real in a scheme already proved secure in the random oracle model in this line. We presented equivalent conditions for a specific oracle $\{h_n\}$ instantiating the random oracle to keep a cryptgraphic scheme secure, using a concept of algorithmic randomness, i.e., a variant of Martin-Löf randomness. Based on this, in particular we showed that the security proved in the random oracle model is firmly maintained after instantiating the random oracle by a random real.

In present paper, we introduce the notion of *effective security*, which is a constructive strengthen of normal (non-constructive) notions of security. In terms of the definitions (1) and (3) of security, the "effectiveness" means that the natural number N can be computed from the code of an adversary $\mathcal{A}$ and a natural number d. We consider signature schemes in the random oracle model, and show that some specific *computable* function $\{h_n\}$ can instantiate the random oracle while keeping the effective security originally proved in the random oracle model. We demonstrate that the effective security is a natural alternative to the normal security notions in modern cryptography by reconsidering the security notions required in modern cryptography.

The results of this paper are based only on the general form of the definitions of security notions for a signature scheme in modern cryptography, and depend neither on specific schemes nor on specific security notions.

2 Preliminaries

We start with some notation about numbers and strings which will be used in this paper. $\#S$ is the cardinality of S for any set S. $\mathbb{N} = \{0, 1, 2, 3, \dots\}$ is the set of natural numbers, and $\mathbb{N}^+$ is the set of positive integers. $\mathbb{Q}$ is the set of rational numbers.

$\{0,1\}^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, 001, 010, \dots\}$ is the set of finite binary strings where λ denotes the *empty string*, and $\{0,1\}^*$ is ordered as indicated. We identify any string in $\{0,1\}^*$ with a natural number in this order. For any $x \in \{0,1\}^*$, $|x|$ is the *length* of x. For any $n \in \mathbb{N}$, we denote by $\{0,1\}^n$ and $\{0,1\}^{\leq n}$ the sets $\{x \mid x \in \{0,1\}^* \;\&\; |x| = n\}$ and $\{x \mid x \in \{0,1\}^* \;\&\; |x| \leq n\}$, respectively. For any $n, m \in \mathbb{N}$, we denote by Func_n^m and $\mathsf{Func}_{\leq n}^m$ the set of all functions mapping $\{0,1\}^n$ to $\{0,1\}^m$ and the set of all functions mapping $\{0,1\}^{\leq n}$ to $\{0,1\}^m$, respectively. A subset S of $\{0,1\}^*$ is called *prefix-free* if no string in S is a prefix of another string in S. We write "r.e." instead of "recursively enumerable."

$\{0,1\}^\infty$ is the set of infinite binary sequences, where an infinite binary sequence is infinite to the right but finite to the left. For any $\alpha \in \{0,1\}^\infty$ and any $n \in \mathbb{N}$, we denote by $\alpha\restriction_n \in \{0,1\}^*$ the first n bits of α. For any $S \subset \{0,1\}^*$, the set $\{\alpha \in \{0,1\}^\infty \mid \exists\, n \in \mathbb{N}\ \alpha\restriction_n \in S\}$ is denoted by $[S]^\prec$.

Lebesgue outer measure $\mathcal{L}$ *on* $\{0,1\}^\infty$ is a function mapping any subset of $\{0,1\}^\infty$ to a non-negative real. In this paper, we use the following properties of $\mathcal{L}$.

Proposition 2.1.

(i) $\mathcal{L}\left([P]^\prec\right) = \sum_{x \in P} 2^{-|x|}$ *for every prefix-free set* $P \subset \{0,1\}^*$.

(ii) $\mathcal{L}(\mathcal{C}) \leq \mathcal{L}(\mathcal{D})$ *for every sets* $\mathcal{C} \subset \mathcal{D} \subset \{0,1\}^\infty$.

(iii) $\mathcal{L}\left(\bigcup_i \mathcal{C}_i\right) \leq \sum_i \mathcal{L}(\mathcal{C}_i)$ *for every sequence* $\{\mathcal{C}_i\}_{i \in \mathbb{N}}$ *of subsets of* $\{0,1\}^\infty$. □

A function $f\colon \mathbb{N} \to \{0,1\}^*$ is called *computable* if there exists a deterministic algorithm which on every input $n \in \mathbb{N}$ halts and outputs $f(n)$. A computable function is also called a *total recursive function*. A real a is *computable* if there exists a computable function $g\colon \mathbb{N} \to \mathbb{Q}$ such that $|a - g(k)| < 2^{-k}$ for all $k \in \mathbb{N}$. We say that $\alpha \in \{0,1\}^\infty$ is *computable* if the mapping $\mathbb{N} \ni n \mapsto \alpha\restriction_n$ is a computable function, which is equivalent to that the real $0.\alpha$ in base-two notation is computable.

3 Signature Schemes in a General Form

We begin by presenting the general form of signature scheme which we consider in this paper. These are the *full-domain hash* (FDH) signature schemes in a general form. We will give our results for them.

Let $\ell(n)$ be a polynomial. An *ℓ-function* is a function $H\colon \mathbb{N} \times \{0,1\}^* \to \{0,1\}^*$ such that $|H(n,x)| = \ell(n)$ for all $n \in \mathbb{N}$ and $x \in \{0,1\}^*$. For each ℓ-function H and $n \in \mathbb{N}$, we define a function $H_n\colon \{0,1\}^* \to \{0,1\}^{\ell(n)}$ by $H_n(x) = H(n,x)$. An ℓ-function serves as an instantiation of the random oracle, such as a cryptographic hash function.

Definition 3.1. *Let $\ell(n)$ be a polynomial. A* signature scheme relative to ℓ-functions *is a tuple* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *of three probabilistic polynomial-time algorithms such that, for every ℓ-function H,*

1. *The* key generation algorithm Gen *takes as input a security parameter* 1^n *and outputs a pair of keys* (pk, sk). *These are called the* public key *and the* private key, *respectively. We assume that n can be determined from each of pk and sk.*
2. *The* signing algorithm Sign *takes as input a private key sk and a message $m \in \{0,1\}^*$. It is*

given oracle access to $H_n(\cdot)$, and then outputs a signature σ, denoted as $\sigma \leftarrow \mathsf{Sign}_{sk}^{H_n(\cdot)}(m)$.

3. *The deterministic* verification algorithm Vrfy *takes as input a public key pk, a massage m, and a signature σ. It is given oracle access to $H_n(\cdot)$, and then outputs a bit b, with $b = 1$ meaning* valid *and $b = 0$ meaning* invalid. *We write this as $b := \mathsf{Vrfy}_{pk}^{H_n(\cdot)}(m, \sigma)$.*

It is required that, for every $n \in \mathbb{N}^+$, for every ℓ-function H, for every (pk, sk) output by $\mathsf{Gen}(1^n)$, and for every $m \in \{0,1\}^*$, $\mathsf{Vrfy}_{pk}^{H_n(\cdot)}(m, \mathsf{Sign}_{sk}^{H_n(\cdot)}(m)) = 1$. □

In this paper we consider the existential unforgeability of signature schemes under adaptive chosen-message attacks as an example. We can show the same results for other security notions, such as the existential unforgeability against key only attacks, the existential unforgeability against known-message attacks, and the existential unforgeability against generic chosen-massage attacks. Let $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme relative to ℓ-functions, and consider the following experiment for a probabilistic polynomial-time adversary $\mathcal{A}$, a parameter n, and a function G mapping a superset of $\{0,1\}^{\leq q(n)}$ to $\{0,1\}^{\ell(n)}$ where $q(n)$ is the maximum value between the running time of $\mathcal{A}$ and the running time of Sign on the parameter n:

The signature experiment $\mathsf{Sig\text{-}forge}_{\mathcal{A},\Pi}(n, G)$:

1. $\mathsf{Gen}(1^n)$ *is run to obtain keys (pk, sk).*
2. *Adversary $\mathcal{A}$ is given pk and oracle access to $\mathsf{Sign}_{sk}^{G(\cdot)}(\cdot)$ and $G(\cdot)$. (The first oracle returns a signature $\mathsf{Sign}_{sk}^{G(\cdot)}(m')$ for any message m' of the adversary's choice.) The adversary then outputs (m, σ). Let $\mathcal{Q}$ denotes the set of messages whose signatures were requested by $\mathcal{A}$ during its execution.*
3. *The output of the experiment is defined to be 1 if and only if* **(1)** *$m \notin \mathcal{Q}$, and* **(2)** *$\mathsf{Vrfy}_{pk}^{G(\cdot)}(m, \sigma) = 1$.*

On the one hand, the existential unforgeability of signature schemes under adaptive chosen-message attacks relative to a specific ℓ-function is defined as follows. This form of the definition corresponds to the condition (3) for the security relative to a specific oracle $\{h_n\}$ considered in the introduction.

Definition 3.2. *Let H be an ℓ-function. A signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ relative to ℓ-functions is* existentially unforgeable under an adaptive chosen-message attack *(or* EUF-ACMA secure*)* relative to H *if for all probabilistic polynomial-time adversaries $\mathcal{A}$ and all $d \in \mathbb{N}^+$ there exists $N \in \mathbb{N}^+$ such that, for all $n > N$, $\Pr[\mathsf{Sig\text{-}forge}_{\mathcal{A},\Pi}(n, H_n) = 1] \leq 1/n^d$.* □

On the other hand, the existential unforgeability of signature schemes under adaptive chosen-message attacks in the random oracle model is formulated as follows. This form of the definition corresponds to the condition (2), and is justified based on the consideration in the introduction.

Definition 3.3. *A signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ relative to ℓ-functions is* existentially unforgeable under an adaptive chosen-message attack *(or* EUF-ACMA secure*)* in the random oracle model *if for all probabilistic polynomial-time adversaries $\mathcal{A}$ and all $d \in \mathbb{N}^+$ there exists $N \in \mathbb{N}^+$ such that, for all $n > N$,*

$$\frac{1}{\#\mathsf{Func}^{\ell(n)}_{\leq q(n)}} \sum_{G \in \mathsf{Func}^{\ell(n)}_{\leq q(n)}} \Pr[\mathsf{Sig\text{-}forge}_{\mathcal{A},\Pi}(n, G) = 1] \leq \frac{1}{n^d},$$

where $q(n)$ is the maximum value between the running time of $\mathcal{A}$ and the running time of Sign on the parameter n. □

4 Main Result

Let H be an ℓ-function. We say that H is *computable* if there exists a deterministic algorithm which on every input (n, x) halts and outputs $H(n, x)$. On the other hand, we say that H is *polynomial-time computable* if there exists a deterministic algorithm which on every input $(1^n, x)$ operates and outputs $H(n, x)$ within time polynomial in n and $|x|$.

Conjecture 1 below means that, in the case where a signature scheme Π satisfies a certain condition $\mathcal{C}$, the existential unforgeability of Π proved to be EUF-ACMA secure in the random oracle model can be firmly maintained in the standard model after instantiating the random oracle by some polynomial-time computable ℓ-function.

Conjecture 1. *Let $\ell(n)$ be a polynomial. Suppose that a signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ relative to ℓ-functions is EUF-ACMA secure in the random oracle model. If Π satisfies $\mathcal{C}$, then there exists a polynomial-time computable ℓ-function (or a polynomial-time computable family of ℓ-functions) relative to which Π is EUF-ACMA secure.* □

Note that an appropriate restriction on a signature scheme Π, i.e., the condition $\mathcal{C}$ on Π, might be necessary to prove Conjecture 1, due to the negative results in the secure instantiation of the random oracle by Canetti, et al. [2]. At present, however, it would seem very difficult to prove it with identifying an appropriate nontrivial condition $\mathcal{C}$.

The second best thing is to investigate whether Conjecture 2 below holds true or not, where we consider the instantiation of the random oracle by simply a computable ℓ-function, which is not necessarily polynomial-time computable.

Conjecture 2. *Let $\ell(n)$ be a polynomial. Suppose that a signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ relative to ℓ-functions is EUF-ACMA secure in the random oracle model. Then there exists a computable ℓ-function H such that Π is EUF-ACMA secure relative to H.* □

In what follows, we show that an "effective" variant of Conjecture 2 holds true. We introduce the notion

of *effective EUF-ACMA security*, which is a constructive strengthen of normal (non-constructive) notions of EUF-ACMA security. In terms of Definitions 3.2 and 3.3 for the normal EUF-ACMA security, the "effectiveness" means that the number N in the definitions can be computed, given the code of an adversary $\mathcal{A}$ and a number d. To begin with a formal definition, we choose a particular recursive enumeration $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \dots$ of all probabilistic polynomial-time adversaries as the standard one for use throughout the rest of this paper. It is easy to show that such an enumeration exists.

On the one hand, the effective EUF-ACMA security relative to a specific ℓ-function is defined as follows.

Definition 4.1. *Let H be an ℓ-function. A signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ relative to ℓ-functions is* effectively existentially unforgeable under an adaptive chosen-message attack *(or* effectively EUF-ACMA secure*)* relative to H *if there exists a computable function $f\colon \mathbb{N}^+ \times \mathbb{N}^+ \to \mathbb{N}^+$ such that, for all $i, d, n \in \mathbb{N}^+$, if $n \geq f(i,d)$ then $\Pr[\mathsf{Sig\text{-}forge}_{\mathcal{A}_i,\Pi}(n, H_n) = 1] \leq 1/n^d$.* □

Obviously, if a signature scheme Π relative to ℓ-functions is effectively EUF-ACMA secure relative to H, then Π is simply EUF-ACMA secure relative to H.

On the other hand, the effective EUF-ACMA security in the random oracle model is defined as follows.

Definition 4.2. *A signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ relative to ℓ-functions is* effectively existentially unforgeable under an adaptive chosen-message attack *(or* effectively EUF-ACMA secure*)* in the random oracle model *if there exists a computable function $f\colon \mathbb{N}^+ \times \mathbb{N}^+ \to \mathbb{N}^+$ such that, for all $i, d, n \in \mathbb{N}^+$, if $n \geq f(i,d)$ then*

$$\frac{1}{\#\mathsf{Func}^{\ell(n)}_{\leq q(n)}} \sum_{G \in \mathsf{Func}^{\ell(n)}_{\leq q(n)}} \Pr[\mathsf{Sig\text{-}forge}_{\mathcal{A}_i,\Pi}(n, G) = 1] \leq \frac{1}{n^d},$$

where $q(n)$ is the maximum value between the running time of $\mathcal{A}_i$ and the running time of Sign *on the parameter n.* □

Obviously, if a signature scheme Π relative to ℓ-functions is effectively EUF-ACMA secure in the random oracle model, then Π is simply effectively EUF-ACMA secure in the random oracle model.

The effective variant of Conjecture 2 is then presented as follows.

Theorem 4.3 (main result)**.** *Let $\ell(n)$ be a polynomial. Suppose that a signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ relative to ℓ-functions is effectively EUF-ACMA secure in the random oracle model. Then there exists a computable ℓ-function H such that Π is effectively EUF-ACMA secure relative to H.* □

In order to prove Theorem 4.3, we need Lemmas 4.4, 4.5, and 4.6 below. The first two can be easily proved. The last one is Exercise 1.9.21 of Nies's textbook [5] of algorithmic randomness. See [5] for the proof of Lemma 4.6.

Lemma 4.4. *Let $f_1, \dots, f_N$ be reals. Suppose that $\frac{1}{N}\sum_{i=1}^{N} f_i \leq \varepsilon$. Then, for every $\alpha > 0$, the number of i for which $\alpha\varepsilon < f_i$ is less than N/α.* □

Lemma 4.5. *Let $d \geq 2$. Then $\sum_{k=n}^{\infty} 1/k^d \leq 2/n$ for every $n \in \mathbb{N}^+$.* □

Lemma 4.6. *Let S be an r.e. subset of $\{0,1\}^*$. Suppose that $\mathcal{L}\left([S]^{\prec}\right) < 1$ and $\mathcal{L}\left([S]^{\prec}\right)$ is a computable real. Then there exists $\alpha \in \{0,1\}^\infty$ such that α is computable and $\alpha \notin [S]^{\prec}$.* □

Proof of Theorem 4.3. Suppose that a signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ relative to ℓ-functions is effectively EUF-ACMA secure in the random oracle model. Then there exists a computable function $f\colon \mathbb{N}^+ \times \mathbb{N}^+ \to \mathbb{N}^+$ such that, for all $i, d, n \in \mathbb{N}^+$, if $n \geq f(i,d)$ then

$$\frac{1}{\#\mathsf{Func}^{\ell(n)}_{\leq q_i(n)}} \sum_{G \in \mathsf{Func}^{\ell(n)}_{\leq q_i(n)}} \Pr[\mathsf{Sig\text{-}forge}_{\mathcal{A}_i,\Pi}(n, G) = 1] \leq \frac{1}{n^d},$$

where $q_i(n)$ is the maximum value between the running time of $\mathcal{A}_i$ and the running time of Sign on the parameter n. Note that the value $q_i(n)$ can be computed, given i and n. It follows from Lemma 4.4 that, for all $i, d, n \in \mathbb{N}^+$, if $n \geq f(i, 2d)$ then

$$\#\left\{G \in \mathsf{Func}^{\ell(n)}_{\leq q_i(n)} \,\middle|\, \Pr[\mathsf{Sig\text{-}forge}_{\mathcal{A}_i,\Pi}(n, G) = 1] > \frac{1}{n^d}\right\} < \frac{\#\mathsf{Func}^{\ell(n)}_{\leq q_i(n)}}{n^d}. \tag{4}$$

In order to apply the method of algorithmic randomness, i.e., Lemma 4.6, we identify an ℓ-function with an infinite binary sequence in the following manner: We first choose a particular bijective total recursive function $b\colon \mathbb{N} \to \mathbb{N}\times\mathbb{N}$ with $b(k) = (b_1(k), b_2(k))$ as the standard one for use throughout the rest of this paper. We assume for convenience that, for every $k, l \in \mathbb{N}$, if $b_1(k) = b_1(l)$ and $k < l$ then $b_2(k) < b_2(l)$. For example, the inverse function of a function $c\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ with $c(m,n) = (m+n)(m+n+1)/2+n$ can serve as such a function b. Then each ℓ-function $H\colon \mathbb{N} \times \{0,1\}^* \to \{0,1\}^*$ is identified with the infinite binary sequence

$$H(b(0))H(b(1))H(b(2))H(b(3))\cdots\cdots. \tag{5}$$

Recall here that we identify $\{0,1\}^*$ with $\mathbb{N}$, as explained in Section 2, and therefore each $b_2(k)$ is regarded as a finite binary string in (5). In what follows, we work with this intuition of the identification.

For each $i, d, n \in \mathbb{N}^+$ we define a subset $[C_{i,d,n}]^{\prec}$ of $\{0,1\}^\infty$ as the set of all ℓ-functions H such that $\Pr[\mathsf{Sig\text{-}forge}_{\mathcal{A}_i,\Pi}(n, H_n) = 1] > 1/n^d$. Namely, we define a subset $C_{i,d,n}$ of $\{0,1\}^*$ as the set of all finite binary strings of the form $x_0 G(\lambda) x_1 G(0) x_2 G(1) x_3 \cdots x_L$ $G(1^{q_i(n)})$ for which the following properties (i), (ii), and (iii) hold for L, $x_0, x_1, x_2, x_3, \dots, x_L$, and G:

(i) $L + 1 = \#\{0,1\}^{\leq q_i(n)}$ (i.e., $L = 2^{q_i(n)+1} - 2$).

(ii) For each $j = 0, \dots, L$, $x_j \in \{0,1\}^*$ and $|x_0 G(\lambda) x_1 G(0) x_2 G(1) x_3 \cdots x_j| = \sum_{k<k_j} \ell(b_1(k))$ where k_j is a natural number such that $b(k_j) = (n, j)$.

(iii) $G\colon \{0,1\}^{\leq q_i(n)} \to \{0,1\}^{\ell(n)}$ and $\Pr[\mathsf{Sig\text{-}forge}_{\mathcal{A}_i,\Pi}(n, G) = 1] > 1/n^d$.

Since $\#\mathsf{Func}^{\ell(n)}_{\leq q_i(n)} = 2^{\ell(n)\#\{0,1\}^{\leq q_i(n)}}$, it follows from (i) of Proposition 2.1 and (4) that, for each $i, d, n \in \mathbb{N}^+$, if $n \geq f(i, 2d)$ then

$$\begin{aligned} \mathcal{L}\left([C_{i,d,n}]^{\prec}\right) &= \sum_{s \in C_{i,d,n}} 2^{-|s|} \\ &< \frac{\#\mathsf{Func}^{\ell(n)}_{\leq q_i(n)}}{n^d} 2^{-\ell(n)\#\{0,1\}^{\leq q_i(n)}} = \frac{1}{n^d}. \end{aligned} \tag{6}$$

We choose a particular computable bijection $\varphi\colon \mathbb{N}^+ \to \{\, (i,d) \mid i \in \mathbb{N}^+ \ \&\ d \geq 2 \,\}$, and define $(\varphi_1(m), \varphi_2(m)) = \varphi(m)$. We then define a computable function $g\colon \mathbb{N}^+ \to \mathbb{N}^+$ by $g(m) = \{f(\varphi_1(m), 2\varphi_2(m)) + 1\}^{m+1}$. For each $m \in \mathbb{N}^+$, we define a subset C_m of $\{0,1\}^*$ by

$$C_m = \bigcup_{n=g(m)}^{\infty} C_{\varphi_1(m),\varphi_2(m),n}. \tag{7}$$

It follows from (iii) of Proposition 2.1, (6), and Lemma 4.5 that, for each $m \in \mathbb{N}^+$,

$$\begin{aligned} \mathcal{L}\left([C_m]^{\prec}\right) &\leq \sum_{n=g(m)}^{\infty} \mathcal{L}\left(\left[C_{\varphi_1(m),\varphi_2(m),n}\right]^{\prec}\right) \\ &< \sum_{n=g(m)}^{\infty} \frac{1}{n^{\varphi_2(m)}} \leq \frac{2}{g(m)} \leq \frac{1}{2^m}. \end{aligned} \tag{8}$$

We then define C by

$$C = \bigcup_{m=1}^{\infty} C_m. \tag{9}$$

Therefore, using (iii) of Proposition 2.1,

$$\mathcal{L}\left([C]^{\prec}\right) \leq \sum_{m=1}^{\infty} \mathcal{L}\left([C_m]^{\prec}\right) < \sum_{m=1}^{\infty} \frac{1}{2^m} = 1. \tag{10}$$

Next we show that C is an r.e. subset of $\{0,1\}^*$. It is easy to see that, given i, d, and n, one can decide the finite subset $C_{i,d,n}$ of $\{0,1\}^*$, since the dyadic rational $\Pr[\mathsf{Sig\text{-}forge}_{\mathcal{A}_i,\Pi}(n, G) = 1]$ is computable, given i, n, and $G\colon \{0,1\}^{\leq q_i(n)} \to \{0,1\}^{\ell(n)}$. Thus, since φ and g are computable functions, it follows from (7) and (9) that C is an r.e. subset of $\{0,1\}^*$.

We then show that $\mathcal{L}\left([C]^{\prec}\right)$ is a computable real. For each $k \in \mathbb{N}$, we define a finite subset D_k of C by

$$D_k = \bigcup_{m=1}^{k} \bigcup_{n=g(m)}^{g(m)2^k - 1} C_{\varphi_1(m),\varphi_2(m),n}.$$

Given $k \in \mathbb{N}$, one can decides the finite set D_k, since φ and g are computable functions and moreover one can decide the finite set $C_{i,d,n}$, given i, d, and n. Therefore, given $k \in \mathbb{N}$, one can calculate the dyadic rational $\mathcal{L}\left([D_k]^{\prec}\right)$ based on (i) of Proposition 2.1. On the other hand, note that

$$C \setminus D_k \subset \left(\bigcup_{m=1}^{k} \bigcup_{n=g(m)2^k}^{\infty} C_{\varphi_1(m),\varphi_2(m),n} \right) \cup \bigcup_{m=k+1}^{\infty} C_m.$$

Thus, using (ii) and (iii) of Proposition 2.1, (6), Lemma 4.5, and (8) we see that, for each $k \in \mathbb{N}$,

$$\begin{aligned} &\mathcal{L}\left([C \setminus D_k]^{\prec}\right) \\ &\leq \sum_{m=1}^{k} \sum_{n=g(m)2^k}^{\infty} \mathcal{L}\left(\left[C_{\varphi_1(m),\varphi_2(m),n}\right]^{\prec}\right) + \sum_{m=k+1}^{\infty} \mathcal{L}\left([C_m]^{\prec}\right) \\ &< \sum_{m=1}^{k} \frac{2}{g(m)2^k} + \sum_{m=k+1}^{\infty} \frac{1}{2^m} \leq \sum_{m=1}^{k} \frac{1}{2^{m+k}} + \frac{1}{2^k} < \frac{1}{2^{k-1}}. \end{aligned}$$

Therefore, since $[C]^{\prec} = [D_{k+1}]^{\prec} \cup [C \setminus D_{k+1}]^{\prec}$, using (ii) and (iii) of Proposition 2.1 we have

$$\left|\mathcal{L}\left([C]^{\prec}\right) - \mathcal{L}\left([D_{k+1}]^{\prec}\right)\right| \leq \mathcal{L}\left([C \setminus D_{k+1}]^{\prec}\right) \leq 2^{-k}$$

for each $k \in \mathbb{N}$. Hence, $\mathcal{L}\left([C]^{\prec}\right)$ is a computable real.

Now, it follows from Lemma 4.6 that there exists $H \in \{0,1\}^\infty$ such that H is computable and $H \notin [C]^{\prec}$. Since H is computable as an infinite binary sequence, it is easy to see that H is also computable as an ℓ-function. On the other hand, let $i, d, n \in \mathbb{N}^+$ with $n \geq g(\varphi^{-1}(i, d+1))$. We then define $m = \varphi^{-1}(i, d+1)$, i.e., $\varphi(m) = (i, d+1)$. Since $H \notin [C]^{\prec}$ and $n \geq g(m)$, it follows from (9) and (7) that $H \notin \left[C_{\varphi_1(m),\varphi_2(m),n}\right]^{\prec} = [C_{i,d+1,n}]^{\prec}$. Therefore, based on the identification (5) of an ℓ-function with an infinite binary sequence, we see that the function $H_n\colon \{0,1\}^* \to \{0,1\}^{\ell(n)}$ satisfies that $\Pr[\mathsf{Sig\text{-}forge}_{\mathcal{A}_i,\Pi}(n, H_n) = 1] \leq 1/n^{d+1} < 1/n^d$. Thus, since the mapping $\mathbb{N}^+ \times \mathbb{N}^+ \ni (i, d) \mapsto g(\varphi^{-1}(i, d+1))$ is a computable function, it follows from Definition 4.1 that Π is effectively EUF-ACMA secure relative to H. □

5 Discussion

In this section, we show that the effective security introduced in the previous section is a natural alternative to the normal security notions in modern cryptography.

In Definitions 3.2 and 3.3 for the normal EUF-ACMA security, the number N is only required to exist, depending on an adversary $\mathcal{A}$ and a number d, that is, the success probability of the attack by an adversary $\mathcal{A}$ on a security parameter n is required to be less than $1/n^d$ for all sufficiently large n, where the lower bound of such n is not required to be computable from $\mathcal{A}$ and d. On the other hand, in Definitions 4.1 and 4.2 for the effective EUF-ACMA security, it is required that the lower bound N of such n can be computed from the code of $\mathcal{A}$ and d.

In modern cryptography based on computational security, it is important to choose the security parameter n of a cryptographic scheme as small as possible to the extent that the security requirements are satisfied, in order to make the efficiency of the scheme as high as possible. For that purpose, it is desirable to be able to calculate a concrete value of N, given the code of $\mathcal{A}$ and d, since N gives a lower bound of the security parameter for which the security requirements specified by $\mathcal{A}$ and d are satisfied. This results in the notion of effective security.

Does the replacement of the normal security notion by the corresponding effective security notion bring difficulties to modern cryptography in general ? We do not think so. It would seem plausible that all the normal security notions can be replaced by the corresponding effective security notions in modern cryptography with little cost. As an example, let us consider the EUF-ACMA security of the RSA-FDH signature scheme under the RSA assumption and its effective counterpart. Let $\mathsf{Succ}^{\mathsf{RSA}}_{\mathcal{A}}(n)$ be the success probability of an algorithm $\mathcal{A}$ in solving the RSA problem on a security parameter n. On the one hand, the (normal) RSA assumption is defined as the condition that, for all probabilistic polynomial-time algorithms $\mathcal{A}$ and all $d \in \mathbb{N}^+$ there exists $N \in \mathbb{N}^+$ such that, for all $n \geq N$, $\mathsf{Succ}^{\mathsf{RSA}}_{\mathcal{A}}(n) \leq 1/n^d$. On the other hand, the *effective* RSA assumption is defined as the condition that there exists a computable function $f\colon \mathbb{N}^+ \times \mathbb{N}^+ \to \mathbb{N}^+$ such that, for all $i, d, n \in \mathbb{N}^+$, if $n \geq f(i,d)$ then $\mathsf{Succ}^{\mathsf{RSA}}_{\mathcal{A}_i}(n) \leq 1/n^d$, where $\mathcal{A}_i$ is the ith algorithm in a particular recursive enumeration of all probabilistic polynomial-time algorithms.

Now, recall the following theorem.

Theorem 5.1 (Bellare and Rogaway [1])**.** *RSA-FDH is EUF-ACMA secure in the random oracle model under the RSA assumption.* □

By analyzing the proof of Theorem 5.1 given in [1], we can see that the following effective version of Theorem 5.1 holds. We can do this task very easily, compared with the non-triviality of the original proof itself.

Theorem 5.2. *RSA-FDH is* effectively *EUF-ACMA secure in the random oracle model under the* effective *RSA assumption.* □

Note that the effective RSA assumption seems more difficult to prove than the RSA assumption. However, in modern cryptography based on computational security, we must make a computational assumption somehow to guarantee the security of a cryptographic scheme. Since making any computational assumption does not cost at all in the development of theory of cryptography, making the effective RSA assumption instead of the RSA assumption would not seem to bring any trouble to modern cryptography. In this manner, we would expect that all the normal security notions can be replaced by the corresponding effective security notions in modern cryptography with little cost. Thus, it would seem plausible that we can easily reconstruct the theory of cryptography based on the effective security notions instead of the normal security notions.

6 Furure Direction

In the previous section, we consider the validity of the effective security notions in modern cryptography. However, it would seem more natural to require that the functions $f\colon \mathbb{N}^+ \times \mathbb{N}^+ \to \mathbb{N}^+$ in Definitions 4.1 and 4.2 are polynomial-time computable rather than simply computable. We call this type of effective security *polynomial-time effective security*. Conjecture 3 below is a polynomial-time effective version of Conjecture 1, and states that the security in the random oracle model implies one in the standard model. In the future, it would be challenging to prove Conjecture 3 (or its appropriate modification) with identifying an appropriate computational assumption COMP and an appropriate nontrivial condition $\mathcal{C}$ on a signature scheme Π.

Conjecture 3. *Let $\ell(n)$ be a polynomial. Suppose that a signature scheme Π relative to ℓ-functions is* polynomial-time effectively *EUF-ACMA secure in the random oracle model. Under the assumption* COMP*, if Π satisfies the condition $\mathcal{C}$, then there exists a* polynomial-time computable *ℓ-function (or a* polynomial-time computable *family of ℓ-functions) relative to which Π is* polynomial-time effectively *EUF-ACMA secure.* □

We conclude this paper with the mention that our result is valid only if the security in the random oracle model is confirmed already. This may imply that the random oracle model is not necessarily an imaginary framework to discuss the security of a cryptographic scheme, but may have substantial implications for the security in the standard model.

Acknowledgments. This work was supported by the Ministry of Economy, Trade and Industry of Japan. The first author was also supported by JSPS KAKENHI Grant Numbers 23340020, 23650001, 24540142.

References

[1] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," Proceedings of the 1st ACM Conference on Computer and Communications Security, ACM, pp.62–73, 1993.

[2] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," *J. ACM*, vol. 51, pp. 557–594, 2004.

[3] R. G. Downey and D. R. Hirschfeldt, *Algorithmic Randomness and Complexity*. Springer-Verlag, New York, 2010.

[4] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Chapman & Hall/CRC Press, 2007.

[5] A. Nies, *Computability and Randomness*. Oxford University Press, Inc., New York, 2009.

[6] K. Tadaki and N. Doi, "Instantiating the random oracle using a random real," Proceedings of the 29th Symposium on Cryptography and Information Security (SCIS2012), 2A3-4, January 30–February 2, 2012, Kanazawa, Japan.