Revisiting the Approaches to Design Secured Hash Functions Using Weakened Random Oracle Models

Fahim Hasan Khan, Syed Muzakkir Ahmed

Abstract— Random oracles are used to model cryptographic hash functions in schemes where strong randomness assumptions are needed of the hash function's output. Random oracles have long been considered in computational complexity theory and many schemes have been proven secure in the random oracle model. In this paper, we studied the recent works for constructing secure cryptosystems and secure hash functions in weakened random oracle models. At first we reviewed the Hash functions used in cryptographic systems for message authentication. Then, we studied the properties of the original Random Oracle Model (ROM). Next, we focused on the weakened variants of the ROM which are called Weak random Oracle Models (WROMs). The WROMs we discussed here are CT-ROM, SPT-ROM and FPT-ROM which are based on the properties of ROM. We also followed an investigation about whether public-key encryption schemes in the random oracle model essentially require the standard security of hash functions by the WROMs. The public-key encryption schemes considered here are Fujisaki-Okamoto conversion (FO), its two artificial variants dFO and wFO and Optimal Asymmetric Encryption Padding (OAEP). The result of the investigation implied that standard encryption schemes such as the OAEP and FO-based one do not always require the standard security of hash functions. Finally, we concluded with our comments and improvement ideas which can be explored in the future.

Index Terms— Cryptographic System, Random Oracle Models, WROMs, Hash Functions, Preimage attacks, Encryption Schemes, Decryption Schemes, Public-key Encryption.

1 INTRODUCTION

CRYPTOGRAPHY has three most important components: message secrecy, confidentiality and integrity [15][16].

There are many circumstances when we don't need of secrecy and confidentiality; rather we need to preserve document by integrity. For instance, somebody write a will to hand over his property to his beneficiary after his death. For this, he doesn't need to encrypt his document; rather he preserves the document through fingerprint. This will ensure the document integrity. Ultimately, the electronic equivalent of both document and fingerprint pair is the message and digest. The digest is prepared through hash function. The hash function is a one-way variation of message authentication code. A hash function accepts a variable-size message M as input and produces a fixed-size output, referred to as a hash code H(M). Unlike other cryptographic system, a hash code does not use a key but is a function only of the input message. The hash code is a function of all the bits of the message and provides an error-detection capability: A change to any bit or bits in the message results in a change to the hash code. The purpose of a hash function is to produce a "fingerprint" of a file, message, or other block of data. Cryptographic hash function requires meeting certain criteria: pre-image resistance, second pre-image resistance and collision resistance. Therefore, the cryptographic hash function is a core issue for message integrity.

A random oracle is a theoretical black box that responds to every query with a truly random response chosen uniformly from its output domain, except that for any specific query, it responds the same way every time it receives that query. Put another way, a random oracle is a mathematical function mapping every possible query to a random response from its output domain. Random oracles are a mathematical abstraction used in cryptographic proofs; they are typically used when no known implementable function provides the mathematical properties required by the proof. A system that is proven secure using such a proof is described as being secure in the random oracle model, as opposed to secure in the standard model. In practice, random oracles are typically used to model cryptographic hash functions in schemes where strong randomness assumptions are needed of the hash function's output. The concept of Random Oracle Model (ROM) has been introduced in 1993 by Bellware and Rogaway [6].

Random Oracle Model is considered as an ideal mathematical model for a hash function. When a new message of any length is given as input, the oracle creates the fixed length message digest with random string of 0's and 1's. That is how, oracle records the message and message digest. In case of existing records, oracle just looks for corresponding digest. Here, another principle is to be strictly adhered i.e. oracle cannot use a formula or an algorithm to calculate the digest. Basically, the implementation schemes of ROM by cryptographic hash function like Message Digest (MD) and Secured Hash Algorithm (SHA). With the fast growing improvement in the attack of cryptography, it is hard to determine the resistivity of hash function to attack like preimage attack, second preimage attack and collisions.

- *Fahim Hasan Khan is currently working as a faculty member in Computer Science and Engineering Department of Military Institute of Science and Technology (MIST), Bangladesh. E-mail: fhk_rifat@yahoo.com*
- *Syed Muzakkir Ahmed is currently working as a faculty member in Computer Science and Engineering Department of Military Institute of Science and Technology (MIST), Bangladesh. E-mail: syed.muzakkir@gmail.com*

In the past, many efforts have been made to identify the weakness and limitations on the properties of ROM. The non-programmable random oracle model where the random oracle is not programmable is proposed by Nielson [11]. In this model, the values that the random oracle answers to some convenient values cannot be set. Unruh proposed a ROM with oracle-dependent auxiliary inputs [12]. In this model, adversaries obtain an auxiliary input that contains information with respect to the random oracle. He showed that the RSA-OAEP encryption scheme [6] is secure in the ROM even under the presence of oracle-dependent auxiliary inputs. The most important work has been done in identifying several weakened versions of the ROM, called weakened random oracle model (WROMs), which offer additional oracles to break some properties of the random oracle [9]. These models capture the situation that adversaries are given an attack algorithm for breaking some specific property of the functions. Subsequently, we like to explore further works in terms of security encryption schemes.

The rest of the paper is organized as follows. The section 2 deals with the related work on hash function, ROM and WROMs. The section 3 briefly discusses the problems and challenges to be faced in the design of the security schemes. In section 4, we discussed the available solution approaches proposed in different texts. Thereafter, in section 5 a brief comparative analysis of ROMs and the security schemes is presented. Finally, in section 6, we conclude our paper with comments and idea of improvements which can be some prospective future research works.

2 Related Works

In last few years, there are significant works are done on WROMs. Several weakened ideal models were considered [3, 9] and several cryptographic schemes were proposed that are secure in a weakened model. Liskov [9] first proposed weakened random oracle models for compression functions: fixed input length weakened random oracle model (FIL-WROM). In the models, adversaries are given sub-oracles in addition to RO. The sub-oracles return collisions, preimages and so on. He proposed the Zipper hash function which indistinguishably behaves like Random Oracle even when an underlying compression function is modeled by FIL-WROM. WROM offer additional oracles to break some properties of the random oracle. These models capture the situation that adversaries are given an attack algorithm for breaking some specific property of the functions. For example, the first-preimage tractable random oracle model offers the random oracle and the first-preimage oracle associated with the random oracle, which returns a first-preimage of the random oracle to adversaries. This first-preimage oracle then corresponds to the attack to the first preimage property of a hash function. We can replace the additional oracle to others such as the second-preimage and collision ones that correspond to the attack to the properties. Thus, the WROMs can capture vulnerability of hash functions even if the parties are allowed to utilize ideal ones as in the ROM. By using WROMs, Liskov constructed hash functions based on weak ideal compression functions and proved it is not distinguishable from the random oracle (RO).

Hoch and Shamir [1] revised this model and proved that the Double pipe hash function and the parallel hash function are indistinguishable from RO even when an underlying compression function is modeled by FIL-WROM. Pasini and Vaudenay also applied Liskov's idea to the security analysis of digital signature schemes [10]. Fischlin and Lehmann also proposed a weakened random oracle model in a similar way to Liskov's one in the context of secure combiners [14].

Numayama et al formalized the WROMs, which allows us to formally analyze the security of the schemes [3]. By using these models, they classified several digital signature schemes by the properties of the ROM. Naito, Wang, and Ohta, in their paper[13], proposes new conversion that can convert any cryptosystem secure in ROM to a new cryptosystem that is secure in the first preimage tractable random oracle model (FPT-ROM) without re-proof. They also propose two hash constructions that are indifferentiable from RO when the underlying compression function is modeled by two-way partially-specified preimage tractable fixed input length random oracle model (TFILFOM).

3 Problem Discussion

3.1 Preliminaries

All uses of cryptographic hash functions require random oracles schemes that require only some property or properties that have a definition in the standard model, such as resistance against collision attack, preimage attack and second preimage attack.

In cryptography, the preimage attack is a classification of attacks on hash functions for finding a message that has a specific hash value. There are two types of preimage attacks:

Preimage attack or First-preimage attack: given a hash h, find a message m (a preimage) such that hash(m) = h.
Second-preimage attack: given a fixed message m1, find a different message m2 (a second preimage) such that, hash(m2) = hash(m1).

A collision attack on a cryptographic hash tries to find two arbitrary inputs that will produce the same hash value, i.e. a hash collision. In contrast to a preimage attack, neither the hash value nor one of the inputs is specified. There are roughly two types of collision attacks:

Collision attack: Find two arbitrary different messages m1 and m2 such that hash(m1) = hash(m2).
Prefix collision attack: Given two different prefixes p1, p2 find two appendages m1 and m2 such that

$$\text{hash}(p1 \| m1) = \text{hash}(p2 \| m2)$$

where ‖ is the concatenation operation.

To give formal definitions of the WROMs, we need to define some notation. Let X and Y be finite sets. Let H be a hash function chosen randomly from all of the functions from X to

Y. We denote by T_H the table $\{(x, H(x)) \mid x \in X\}$. We identify the hash function H with the table T_H. We next define the random oracle and the additional oracles associated with $H : X \rightarrow Y$ as follows. (details are explained in [2] and [3]).

Random oracle RO^H: Given x, return y such that $(x, y) \in T_H$.
Collision oracle CO^H: On the query, first pick one entry $(x, y) \in T_H$ uniformly at random. If there is no other entry $(x', y) \in T_H$, then answer $\perp$. Otherwise, pick one entry $(x', y) \in T_H$ satisfying x , x' uniformly at random and answer (x, x').
Second-preimage oracle SRO^H: Given (x, y), if $(x, y) < T_H$ answer $\perp$. If there is no other entry $(x', y) \in T_H$, then answer $\perp$. Otherwise, pick one entry $(x', y) \in T_H$ satisfying x , x' uniformly at random and answer x'.
First-preimage oracle FPO^H: Given y, if there is any entry $(x, y) \in T_H$ then return such an x uniformly at random. Otherwise return $\perp$.

The formal definitions of the WROMs are given as follows. The WROMs consist of three components, a hash function h chosen randomly from all of the functions from X to Y, the random oracle, and the additional oracle associated with h. The models are called the CT-ROM, SPT-ROM, and FPT-ROM, if the additional oracle is the collision, second-preimage, and first-preimage oracle, respectively.

3.2 Difference between ROM and WROM

There are some important differences between the ROM and WROMs by considering the ROM and FPT-ROM. WROMs offer additional oracles to break some properties of the random oracle. In the both models, the function H, i.e., the table TH is uniformly distributed. In the ROM, if one queries some x that has never been queried to the random oracle, the value of H(x) is uniformly distributed regardless of the past queries. That is, the knowledge of the past queries does not affect the entries not queried in the table. This property of the ROM is called uniformity. In contrast to the situation in the ROM, when it comes to the FPT-ROM, this property is not attained. Recall that the first-preimage oracle uniformly returns one of the preimages, say x, of queried value y. If the first-preimage oracle leaks a number of preimages of y, the value of H(x) is not uniformly distributed for an x not queried yet.

3.3 Problem formulation and Simulation Methods

We studied very vividly the relationship between the components of WROMs and the variants of existing public-key encryption schemes and therefore, depict whether public-key encryption schemes in ROM essentially require the standard security of hash functions by the WROMs. In Fig 1 we try to relate the major components discussed in this paper and their role in the design of security schemes.

In almost all the security proofs in the ROM, the reduction algorithms simulate the random oracles. When it comes to the security proofs in the WROMs, the reduction algorithms have to simulate both the random and the additional oracle, which makes differences of the simulation methods in the WROMs from those in the ROM. Numayama et al. [3] proposed the simulation methods for WROMs, but they required an unproven assumption. Under this assumption, they constructed the simulation algorithms, RO, CO, SPO, and FPO, for the security proofs in the WROMs as given in the following proposition.

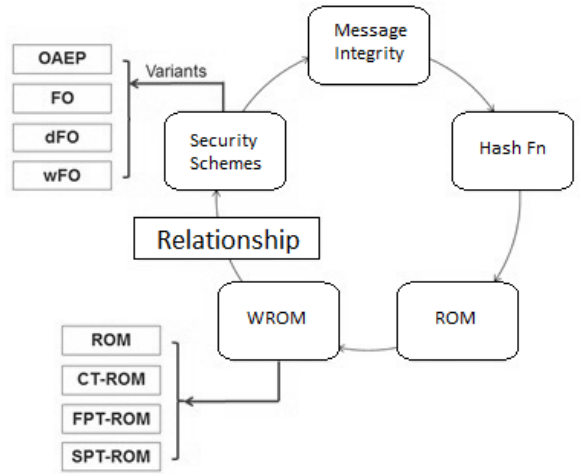

Fig. 1 Role of WROMs in Security Schemes

There is an efficient sampling algorithm appropriate for this purpose in the real-number computation model [4]. The algorithm is modified and rigorously analyzed the error bound in the bit computation model. Finally a simulation method without assumption is obtained [2]. It can statistically simulate the random oracle, collision oracle, second-preimage oracle, and first-preimage oracle in the WROMs. That is, the output distributions of the oracles in the WROMs are statistically close to the output distributions of the algorithms RO, CO, SPO, and FPO, respectively.

4. Security of Encryption Schemes in WROMs

4.1 Notation and Definitions

Public-key encryption schemes: We first give notation and notions for public-key encryption schemes briefly. A public-key encryption scheme PKE = (Gen, Enc, Dec) over a plaintext space M and a random coin space R is defined by the following three algorithms. Let k denote the security parameter.

Key Generation: On input 1^k, the key generation algorithm Gen(1^k) produces a public/secret key pair (pk, sk).
Encryption: Given a public key pk, a plaintext $m \in M$, and a random string $r \in R$, the encryption algorithm $Enc_{pk}(m; r)$ outputs a ciphertext *c* corresponding to the plaintext m.
Decryption: Given a secret key sk and ciphertext c, the decryption algorithm $Dec_{sk}(c)$ outputs the plaintext $m \in M$ or the special symbol $\perp \notin M$ corresponding to the ciphertext c.
We require the perfect completeness, that is, for every (pk, sk) generated by Gen(1^k), every plaintext $m \in M$, and every random string $r \in R$, it should be satisfied that
$Dec_{sk}(Enc_{pk}(m; r)) = m$.

We only consider three standard security notions for public-key encryption schemes, the one-wayness against chosen-

plaintext attack (OW-CPA), the indistinguishability against chosen-plaintext attack (IND-CPA), and the indistinguishability against adaptive chosen-ciphertext attack (IND-CCA2). For $\gamma = \gamma(k)$, we say PKE is γ-uniform if for any key pair (pk, sk) generated by Gen(1^k), any m ∈ M, and c ∈ {0, 1}*, we have $Pr_{r \leftarrow R}[c = Enc_{pk}(m; r)] \leq \gamma$. There exists an OW-CPA public-key encryption scheme with γ-uniformity.

4.2 Brief review for Fujisaki-Okamoto (FO) conversion

The Fujisaki-Okamoto (FO) conversion was proposed by Eiichiro Fujisaki and Tatsuaki Okamoto, to obtain highly secure public-key encryption schemes in the ROM [5]. As the standard one-time pad satisfies the requirement of the FO conversion, the one-time pad is selected as the symmetric-key encryption scheme used in the FO conversion for simplicity.

Considering PKE be an OW-CPA secure and γ-uniform public-key encryption scheme over a plaintext space M and a randomness space R. Now the FO conversion converts PKE to an IND-CCA2 secure one PKE′ = FO(PKE) over a plaintext space M′= {0, 1}$^{k'}$ and a randomness space R′ = M. Here k′ denotes the length of plaintexts, which is polynomially related to the security parameter k. The encryption technique of PKE′ is explained as follows: For a plaintext m ∈ M′ = {0, 1}k′ and a random string r ∈ R′= M, the ciphertext is

$$(c1, c2) = (Enc_{pk}(r; H(m, r)), G(r) \oplus m)$$

Here H: {0, 1}$^{k'}$ × M → R and G : M → {0, 1}$^{k'}$ are hash functions modeled as the random oracles. The decryption technique is specified as follows: For a given ciphertext (c1, c2), c1 decrypted by sk and r is obtained. Then, m is extracted by c2 ⊕ G(r)

TABLE 1
PKE′ OBTAINED BY THE FO CONVERSION

Key Generation	Input:	1^k
	1:	$(pk, sk) \leftarrow Gen(1^k)$
	Output:	(pk, sk)
Encryption	Input:	$m \in \{0, 1\}^k$
	1:	$r \leftarrow \mathcal{M}$
	2:	$g \leftarrow G(r)$
	3:	$h \leftarrow H(m, r)$
	4:	$c_1 \leftarrow Enc_{pk}(r; h)$
	5:	$c_2 \leftarrow m \oplus g$
	Output:	(c_1, c_2)
Decryption	Input:	(c_1, c_2)
	1:	$r \leftarrow Dec_{sk}(c)$
	2:	$g \leftarrow G(r)$
	3:	$m \leftarrow c_2 \oplus g$
	4:	$h \leftarrow H(m, r)$
	5:	If $c_1 = Enc_{pk}(r; h)$ set $o \leftarrow m$
	6:	Otherwise set $o \leftarrow \perp$
	Output:	o

and c1 = Enc_{pk}(r; H(m, r)) is verified. If not, output is⊥. In a brief, H(m, r) ensures that if a ciphertext (c1, c2) is valid then the encryptor producing (c1, c2) knows corresponding m and r.

4.3 Security of the Original FO Conversion

Here it is showed that the obtained scheme by the conversion FO with the one-time pad is secure in the SPT-ROM, but not secure in the FPT-ROM in some parameter setting.

Let G : M → {0, 1}$^{k'}$ and H : {0, 1}$^{k'}$ × M → R be hash functions modeled as the random oracles.

Recall the encryption procedure of PKE′ = FO(PKE). For a plaintext m ∈ M′= {0, 1}$^{k'}$ and a random string r ∈ R′ = M, the ciphertext is (Enc_{pk}(r; H(m, r)),G(r) ⊕ m). The scheme is summarized in Table 1.

Theorem 1. *Suppose that PKE is OW-CPA secure and γ-uniform for some negligible γ. Then, PKE′ = FO(PKE) is IND-CCA2 secure in the SPT-ROM.*

However, the presence of the first-preimage oracle for G violates the IND-CPA security of PKE′ in some parameter settings. Note that if m is $0^{k'}$, the second component of the ciphertext is G(r), which is vulnerable the first-preimage oracle of G.

Theorem 2. *Let C = #M/$2^{k'}$. Assume that C = $k^{O(1)}$. Then, PKE′ = FO(PKE) is not IND-CPA secure in the FPT-ROM.*

The detail proofs of Theorem 1 and 2 appear in [2].

4.4 Security of the First Variant dFO

The first artificial variant dFO is introduced here and showed that dFO is secure in the ROM, but not secure in general in the CT-ROM. Like FO, the variant dFO converts a public-key encryption scheme PKE, with the one-time pad, to another public-key encryption scheme PKE′ = dFO(PKE). The encryption procedure of PKE′ is explained as follows. For a plaintext m ∈ M′= {0, 1}$^{k'}$ and a random string r ∈ R′= M, the ciphertext of PKE′ is

$$(c1, c2) = (Encpk(r; H(F(m), r)), G(r) \oplus m)$$

Here F : {0, 1}$^{k'}$ → P, G : M → {0, 1}$^{k'}$, and H : P × M → R, for an appropriate set P, are hash functions modeled as the random oracle. Formal description is given in Table 2.

The idea to weaken the conversion is briefed as follows: H(m, r) in the FO conversion can be considered as encryptor's signature on m and r. To make it vulnerable by a collision, a new random oracle F is introduced and replace H(m, r) with H(F(m), r). The replacement does not harm the security in the random oracle model, while it can be exploited by the presence of the collision oracle COF.

The following theorems are on the security and weakness of dFO. The proofs are elaborated in [2].

Theorem 3. *Assume that PKE is an OW-CPA secure and γ-uniform public-key encryption scheme for some negligible γ. Then, PKE ′= dFO(PKE) is IND-CCA2 secure in the ROM if #P = $2^{\omega(\log k)}$.*

TABLE 2
PKE′ OBTAINED BY THE DFO CONVERSION

Key Generation	Input: 1^k 1: $(pk, sk) \leftarrow Gen(1^k)$ Output: (pk, sk)
Encryption	Input: $m \in \{0, 1\}^{k'}$ 1: $r \leftarrow \mathcal{M}$ 2: $g \leftarrow G(r)$ 3: $h \leftarrow H(F(m), r)$ 4: $c_1 \leftarrow Enc_{pk}(r; h)$ 5: $c_2 \leftarrow m \oplus g$ Output: (c_1, c_2)
Decryption	Input: (c_1, c_2) 1: $r \leftarrow Dec_{sk}(c)$ 2: $g \leftarrow G(r)$ 3: $m \leftarrow c_2 \oplus g$ 4: $h \leftarrow H(F(m), r)$ 5: If $c1 = Enc_{pk}(r; h)$ set $o \leftarrow m$ 6: Otherwise set $o \leftarrow \perp$ Output: o

Theorem 4. *Let PKE be a public-key encryption scheme. If* $\#P \leq 2^{k'}$ *then PKE'= dFO(PKE) is not IND-CCA2 secure in the CT-ROM.*

4.5 Security of the Second Variant wFO

Next, the second artificial variant wFO is introduced and showed that the obtained scheme by wFO is secure in the CT-ROM, however not generally secure in the SPT-ROM.

The encryption procedure of PKE'= wFO(PKE) is given as follows. For a plaintext

$m \in M' = \{0, 1\}^{k'}$ and random strings $(r, s) \in R' = M \times S$, the ciphertext of PKE′ is

$$(c1, c2, c3) = (Encpk(r; H(F(m, s), r)), G(r) \oplus m, s)$$

Here $F : \{0, 1\}^{k'} \times S \rightarrow P$, $G : M \rightarrow \{0, 1\}^{k'}$, and $H : P \times M \rightarrow R$ are hash functions modeled as the random oracles. The formal definition is given in Table 3.

It can be noted that (H(F(m, s), r), s) is a proof of knowledge on (m, r, s) which resists a collision on F however is vulnerable by a second-preimage attack against F as in Numayama et al. [3]. It can be shown that the obtained scheme is IND-CCA2 secure in the CT-ROM and the detail proofs of theorem 5 and 6 is in [2]

Theorem 5. *Suppose that PKE is a OW-CPA secure and γ-uniform public-key encryption scheme for some negligible γ. Then, PKE'= wFO(PKE) is IND-CCA2 secure in the CT-ROM if #P−1 and #S−1 are negligible in k.*

However, its security is broken under the presence of the second-preimage oracle for F.

Theorem 6. *Let PKE be a public-key encryption. If* $\#P \leq 2^{k'} \bullet \#S$, *then the scheme PKE'= wFO(PKE) is not IND-CCA2 secure in the SPT-ROM.*

TABLE 3
PKE′ OBTAINED BY THE WFO CONVERSION

Key Generation	Input: 1^k 1: $(pk, sk) \leftarrow Gen(1^k)$ Output: (pk, sk)
Encryption	Input: $m \in \{0, 1\}^{k'}$ 1: $r \leftarrow \mathcal{M}$ 2: $g \leftarrow G(r)$ 3: $s \leftarrow S$ 4: $h \leftarrow H(F(m, s), r)$ 5: $c_1 \leftarrow Enc_{pk}(r; h)$ 6: $c_2 \leftarrow m \oplus g$ 7: $c_3 \leftarrow s$ Output: (c_1, c_2, c_3)
Decryption	Input: (c_1, c_2, c_3) 1: $r \leftarrow Dec_{sk}(c)$ 2: $g \leftarrow G(r)$ 3: $m \leftarrow c_2 \oplus g$ 4: $h \leftarrow H(F(m, c_3), r)$ 5: If $c_1 = Enc_{pk}(r; h)$ set $o \leftarrow m$ 6: Otherwise set $o \leftarrow \perp$ Output: o

4.6 Brief Review for Optimal Asymmetric Encryption Padding (OAEP)

Optimal Asymmetric Encryption Padding (OAEP) is a padding scheme often used together with RSA encryption. OAEP was introduced by Bellare and Rogaway[7]. The OAEP algorithm is a form of Feistel network which uses a pair of random oracles G and H to process the plaintext prior to asymmetric encryption. When combined with any secure trapdoor one-way permutation f, this processing is proved in the random oracle model to result in a combined scheme which is semantically secure under chosen plaintext attack (IND-CPA). When implemented with certain trapdoor permutations (e.g., RSA), OAEP is also proved secure against chosen ciphertext attack. OAEP can be used to build an all-or-nothing transform. OAEP satisfies the following two goals:

1. Add an element of randomness which can be used to convert a deterministic encryption scheme (e.g., traditional RSA) into a probabilistic scheme.
2. Prevent partial decryption of ciphertexts (or other information leakage) by ensuring that an adversary cannot recover any portion of the plaintext without being able to invert the trapdoor one-way permutation f.

4.7 Security of OAEP

The final focus is on the OAEP and its IND-CCA2 security in the FPT-ROM is presented. For the security parameter k, k0

and k1 is considered to be functions in k, where k0 < k − k0. F is considered to be a family of partial-domain one-way trapdoor permutations of a domain $\{0,1\}^{k-k0} \times \{0,1\}^{k0}$. (The definition of the partial-domain one-wayness is described in [8].) Also, let G and H be hash functions such that
$G : \{0,1\}^{k0} \rightarrow \{0,1\}^{k-k0}$ and $H : \{0,1\}^{k-k0} \rightarrow \{0,1\}^{k0}$.
Now, the OAEP encryption scheme based on F is described in Table 4.

TABLE 4
PKE′ OBTAINED BY OAEP

Key Generation	Input:	1^k
	1:	$(f_{pk}, g_{sk}) \leftarrow F$
	Output:	(f_{pk}, g_{sk})
Encryption	Input:	$m \in \{0,1\}^{k-k_0-k_1}, f_{pk}$
	1:	$r \leftarrow \{0,1\}^{k_0}$
	2:	$s \leftarrow (m \parallel 0^{k_1}) \oplus G(r)$
	3:	$t \leftarrow H(s) \oplus r$
	4:	$c \leftarrow f_{pk}(s \parallel t)$
	Output:	c
Decryption	Input:	c, g_{sk}
	1:	$s \parallel t \leftarrow g_{sk}(c)$
	2:	$r \leftarrow t \oplus H(s)$
	3:	$M \leftarrow s \oplus G(r)$
	4:	If $M = m \parallel 0^{k_1}$ set $o \leftarrow m$
	5:	Otherwise set $o \leftarrow \perp$
	Output:	o

The following theorem is obtained from [2] that state the security of the OAEP encryption scheme in the FPT-ROM.

Theorem 7. *Let F be a family of partial-domain one-way trapdoor permutations. Then, the OAEP encryption scheme based on F is IND-CCA2 secure in the FPT-ROM.*

5. Comparative Analysis

In [2] it is investigated whether public-key encryption schemes constructed in the ROM essentially require the standard security of hash functions by further extending the direction originated from Liskov [9]. In particular, their security in ROM, CT-ROM, SPT-ROM, and FPT-ROM is considered. Here, they are ordered according to their strengths. For instance, the security of encryption schemes in the FPT-ROM implies that in the SPT-ROM and such implications hold between each adjacent two models.

The security of four schemes is summarized in Table 5.

TABLE 5
SECURITY OF FOUR SCHEMES

scheme/model	OAEP	FO	wFO	dFO
ROM	*secure*	*secure*	*secure*	*secure*
CT-ROM				*insecure*
SPT-ROM			*insecure*	
FPT-ROM		*insecure*		

It is demonstrate in [2] that the security notions in the four WROMs can be strictly separated in the context of encryption schemes. For the separation, the security of the encryption schemes obtained by the Fujisaki-Okamoto conversion (FO) [5], its two artificial variants (dFO and wFO), and the OAEP is focused. Precisely, the following four statements are proved:

1. OAEP is IND-CCA2 secure in the FPT-ROM.
2. FO is IND-CCA2 secure in the SPT-ROM, but not IND-CPA secure in the FPT-ROM.
3. wFO is IND-CCA2 secure in the CT-ROM, but not IND-CCA2 secure in the SPT-ROM.
4. dFO is IND-CCA2 secure in the ROM, but not IND-CCA2 secure in the CT-ROM.

6. Conclusion and Future Works

In this literature we have discussed the basic working principles of Hash Functions and Random Oracle Model [6]. We also discussed the weakened variants of ROM the weak Random Oracle Models [9] and Hash Functions in wROMS. Then we walked through the previous and recent works on constructing secure cryptosystems and secure hash functions based on ROM and wROMs. In the latter half of this literature we compared some public-key encryption schemes and compare their securities for ROM and three variants of wROMs by revisiting works of Kawachi et al [2]. The summary of the result of the comparison is also mentioned briefly.

From this study several drawbacks are found which can be addressed as future works. First, most of the recent works like [2][3][10] is based on the simplified version of wROM proposed by Liskov[9]. The working principle of the wROM can be further analyzed. Also, these works discussed the monolithic random oracle H and the additional oracles associated with H. The monolithic H is considered as a black-box which can be simply used without considering the internal structure. But, the realistic implementation of H is far more complicated. So, the works can be done addressing the gap between H and realistic instantiation of ROM and wROM. Except for the OAEP and FO conversion, there are several other conversion methods in the ROM. Also, investigating the security of these conversion methods in the WROMs can be considered as a very interesting future work.

References

[1] Hoch, Jonathan J., and Adi Shamir. "On the strength of the concatenated hash combiner when all the hash functions are weak." *Automata, Languages and Programming*, pp. 616-630. Springer Berlin Heidelberg, 2008.

[2] Kawachi, Akinori, Akira Numayama, Keisuke Tanaka, and Keita Xagawa. "Security of encryption schemes in weakened random oracle models." *Public Key Cryptography–PKC 2010*, pp. 403-419. Springer Berlin Heidelberg, 2010.

[3] Numayama, Akira, Toshiyuki Isshiki, and Keisuke Tanaka. "Security of digital signature schemes in weakened random oracle models." *Public Key Cryptography–PKC 2008*, pp. 268-287. Springer Berlin Heidelberg, 2008.

[4] Relles, Daniel A. "A simple algorithm for generating Binomial ran-

dom variables when N is large." *Journal of the American Statistical Association 67*, no. 339 (1972): 612-613.

[5] Fujisaki, Eiichiro, and Tatsuaki Okamoto. "Secure integration of asymmetric and symmetric encryption schemes." *Advances in Cryptology – CRYPTO'99*, pp. 537-554. Springer Berlin Heidelberg, 1999.

[6] Bellare, Mihir, and Phillip Rogaway. "Random oracles are practical: A paradigm for designing efficient protocols."*Proceedings of the 1st ACM conference on Computer and communications security*, pp. 62-73. ACM, 1993.

[7] Bellare, Mihir, and Phillip Rogaway. "Optimal asymmetric encryption." *Advances in Cryptology – EUROCRYPT'94,* pp. 92-111. Springer Berlin Heidelberg, 1995.

[8] Fujisaki, Eiichiro, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. "RSA-OAEP is secure under the RSA assumption." *Advances in Cryptology – CRYPTO 2001*, pp. 260-274. Springer Berlin Heidelberg, 2001.

[9] Liskov, Moses. "Constructing an ideal hash function from weak ideal compression functions." *Selected Areas in Cryptography*, pp. 358-375. Springer Berlin Heidelberg, 2007.

[10] Pasini, Sylvain, and Serge Vaudenay. "Hash-and-sign with weak hashing made secure." *Information Security and Privacy*, pp. 338-354. Springer Berlin Heidelberg, 2007.

[11] Nielsen, Jesper Buus. "Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case." *Advances in Cryptology – Crypto 2002*, pp. 111-126. Springer Berlin Heidelberg, 2002.

[12] Unruh, Dominique. "Random oracles and auxiliary input."*Advances in Cryptology-CRYPTO 2007*, pp. 205-223. Springer Berlin Heidelberg, 2007.

[13] Naito, Yusuke, Lei Wang, and Kazuo Ohta. "How to construct cryptosystems and hash functions in weakenend random oracle models." *Cryptology ePrint Archive*, Report 2009/550, 2009.

[14] Fischlin, Marc, and Anja Lehmann. "Security-amplifying combiners for collision-resistant hash functions." *Advances in Cryptology-CRYPTO 2007*, pp. 224-243. Springer Berlin Heidelberg, 2007.

[15] Forouzan, B. A. Cryptography & Network Security. McGraw-Hill, Inc. 2007

[16] William, S. Cryptography and Network Security, 4/E. Pearson Education India. 2006