What Hashes Make RSA-OAEP Secure?

Daniel R. L. Brown*

August 8, 2007

Abstract

Firstly, we demonstrate a pathological hash function choice that makes RSA-OAEP insecure. This shows that at least some security property is necessary for the hash functions used in RSA-OAEP. Nevertheless, we conjecture that only some very minimal security properties of the hash functions are actually necessary for the security of RSA-OAEP. Secondly, we consider certain types of reductions that could be used to prove the OW-CPA (i.e., the bare minimum) security of RSA-OAEP. We apply metareductions that show if such reductions existed, then RSA-OAEP would be OW-CCA2 insecure, or even worse, that the RSA problem would solvable. Therefore, it seems unlikely that such reductions could exist. Indeed, no such reductions proving the OW-CCA2 security of RSA-OAEP exist.

Key Words: RSA, OAEP, Provable Security, Public-key Encryption, IND-CCA2, OW-CPA, Impossibility Results

1 Introduction

The public-key encryption scheme called RSA-OAEP is proven [BR94, Sho01, FOPS01] to be IND-CCA2 secure under the assumptions that the RSA problem is hard and that the adversary treats the hash functions instantiating OAEP as random oracles. This paper examines the security of RSA-OAEP when the adversary is not restricted to treat the hash functions as random oracles, which is often known as the *standard model*. Informally, this means that we consider the possibility that adversaries might be able to exploit weaknesses in hash functions to attack RSA-OAEP. Surprisingly little seems to be known on this matter.

1.1 Necessary Hash Security Properties

Intuitively, one might expect that at least some kind of security properties of the hash function are *necessary* for the security of RSA-OAEP. We demonstrate this in §2 by giving some pathological hash function instantiations under which RSA-OAEP is not secure. (Similarly, pathological trapdoor one-ways functions, instead of RSA, have also been considered [Sho01] to test the assumptions necessary on the trapdoor function.) Despite these pathological examples, we conjecture that the security properties necessary for the hash function are very minimal. Notably, our conjecture implies that collision resistance and preimage resistance are not actually necessary properties for the hash function in RSA-OAEP. If true, this would be fortunate because, although the collision resistance of SHA-1 has succumbed to a series recent attacks, such as [WYY05], the security of RSA-OAEP instantiated with SHA-1 may not be correspondingly affected as a consequence.

*Certicom Research

1.2 Sufficient Hash Security Properties

We seek properties (conjectured or proven) about the hash function that—in conjunction with properties of the RSA problem—are *sufficient* to ensure the security of RSA-OAEP. Ideally, the necessary and sufficient properties would be identical. To date, the only known property is that the adversary treats the hash function like a random oracle. (As usual, this is not strictly a property of the hash function, but rather of the adversary.) In this matter, we do not provide any new security proofs. Rather, we have eliminated certain kinds of proofs.

More precisely, we hypothesize three particular types of security proofs. These types of proofs use reductions, with certain important restrictions. A restriction applying to all of our three types of reductions is that they are *key preserving*, as in [PV06]. This means that the reductions reduce an instance of the RSA problem to the problem of breaking the security RSA-OAEP with the *same* RSA public key. The three types of reduction differ on the second restriction. These are *hash-generic*, *hash-agnostic*, and *hash-specific*, which we are defined formally in §3.1, and summarized below:

- A *hash-agnostic* reduction takes an RSA-OAEP adversary and solves the corresponding RSA problem. Note that this reduction does not attempt to break any problem related to the hash. Nevertheless, a hash-agnostic could potentially only work for a specific hash function.
- A *hash-generic* reduction takes an RSA-OAEP adversary and solves the corresponding RSA problem, but also must treat the hash function as a random oracle. We emphasize that it is the reduction that treats the hash functions as random oracle here, *not* the adversary.
- A *hash-specific* reduction takes an RSA-OAEP adversary and solves the corresponding RSA problem or some problem related to the hash function, such as finding a preimage of a random value in the range of the hash. This is a less restrictive type of reduction than the previous two types, and it would also seem to the most natural type reduction that one would want.

Informally, a hash-agnostic or hash-generic reduction suggests that RSA-OAEP is secure for any hash function, while a hash-specific reduction only suggests that RSA-OAEP is secure if instantiated with hash functions with a given security property.

Generally, the adversaries in the reductions above will be taken to be a OW-CPA adversary. This is the most basic adversary, and resistance to a OW-CPA adversary is the lowest grade of security. Indeed, a public key encryption scheme without OW-CPA security might as well be the identity cleartext encryption scheme. It seems quite reasonable to hope that, if RSA-OAEP is secure at all and it is possible to find for it a security proof in the standard model, then it has a proof of OW-CPA security that uses one of the reductions above.

In §3.2, we find metareductions, which if given an oracle for one of the reductions above, can break the OW-CCA2 security of RSA-OAEP. Furthermore, in the case of hash-generic reductions, the metareductions can solve the RSA problem.

1.3 Related Work

Paillier and Villar [PV06] proved a result similar to one here, independently of this paper. Their *key-preserving black-box* reductions correspond to the *hash-agnostic* reductions given here. Hash agnostic reductions do not tie the security of breaking RSA-OAEP to any assumptions whatsoever about the hash functions with which it is instantiated.

Boldyreva and Fischlin [BF05, BF06] proved the security of OAEP in the (partial and full) standard model, but they do not instantiate the trapdoor one-way function with RSA. Instead they use a construction that maps some bits in the clear, and the rest of the bits with a subsidiary trapdoor one-way function, which could be RSA. Therefore, their result does not contradict the results here and in [PV06], despite being in an opposite direction. Furthermore, they instantiate OAEP with a hash function with security properties that on the chosen trapdoor functions. This is a second reason that their result does not contradict ours about hash-specific reductions.

Paillier and Vergnaud [PV05] gave some impossibility results in the discrete log setting. They gave strong evidence that most digital signatures schemes could not be proven, in the standard model, to be as difficult as the discrete logarithm problem. They imposed, however, a significant restriction on the security proofs: they had to use *algebraic* reductions.[1] Loosely speaking, an algebraic reductions treats the cryptographic group as a generic group, only accessible by means of random oracles. Metaphorically, they put the gloves on the security proof rather than the adversary. This is similar to our *hash-generic* reductions, where we only give the security proof random oracle access to the hash function.

Dodis, Oliviera and Pietrzak [DOP05] examine RSA-FDH signatures. They show that, in the standard model, it cannot be proved secure unless the trapdoor one-way function used (such as RSA) has a property untrue of random functions. Their approach is slightly different from a metareduction, but they obtain a very general result. Their result, however, does not apply directly to the specific RSA trapdoor function because its homomorphic and random self-reducibility properties make it distinct enough from a random function to exclude their separation results from applying to it. In this work, we focus on the specific RSA trapdoor functions rather than some general class of security properties of trapdoor functions. Indeed, the homomorphic and self-reducible properties of RSA are pivotal to our result.

Shoup gives [Sho01] an argument that the design of OAEP cannot be proven secure when RSA is replaced by an arbitrary trapdoor one-way function. He hypothesizes a pathological trapdoor one-way function for which OAEP becomes insecure. This is more akin to the insecure instantiations of RSA-OAEP discussed in §2.

The phenomenon of security proofs leading to attacks is not new. Rabin's digital signatures and public key encryption [Rab79] had reductions showing that their the basic security against passive adversaries was equivalent to the integer factorization problem. It was soon discovered, though, that these reduction algorithms also led to an active attacks [Wil80, GMR88].[2] Countermeasures, such as OAEP, have since been discovered to thwart the active attacks. The results in this paper, however, show that, despite OAEP appearing to thwart attacks and enabling security proofs in the random oracle model, it does not completely overcome the apparent paradox that certain kinds of security proofs can lead to attacks.

2 Weak Hashes for RSA-OAEP

In this section, we examine security properties are needed for the hash functions used to instantiate RSA-OAEP by considering a pathological choice: $G \equiv H \equiv 0$, that is, the constant zero-value functions. Note that RSA-OAEP allows correct deciphering for any deterministic choice of G and

[1] Their algebraic reductions are similar to those that Boneh and Venkatesan [BV98] used in comparing the RSA problem to factoring.

[2] Metareductions have also appeared in other contexts, such as Boneh and Venkatesan's [BV98] metareduction that takes a reduction of factoring to the RSA problem and uses it to factor.

H, so our pathological choice does not interfere with the private key holder's ability to decrypt legitimate RSA-OAEP ciphertexts. Note, obviously, that our choice of G and H are not collision resistant, not preimage resistant, and not pseudorandom.

Then, with our choice of G and H are RSA-OAEP ciphertext has the form

$$c = (m\|0^k\|t)^e \bmod n \tag{1}$$

where m is the plaintext message, 0^k is a fixed bit string (of all zero bits), r is a random bit string (chosen by the encryptor), and (n, e) is the RSA public key (of the decryptor). More generally, if we can allow an arbitrary choice of hash function H, then we replace that the t above by $t = r \oplus H(m\|z)$.

Theorem 1. *Suppose now that we use RSA public exponent $e = 3$. Suppose further that the length of t is at most one third or less[3] than that of the ciphertext c. With this choice of $G \equiv 0$ and the sufficiently small length of t, RSA-OAEP succumbs to the following IND-CPA attack.*

Proof. The adversary chooses a message $m_0 = 0^l$, that is, a message of all zero bits of the correct length, and another message m_1, which is arbitrary. Now $b \in \{0, 1\}$ and the adversary is given the challenge of finding b from c_b where c_b is an encryption of m_b. The adversary can test if $b = 0$ as follows. If $b = 0$, then c_b will be a perfect cube, because $t^3 < n$, so no modular reduction will take place. □

Furthermore, we conjecture the following worse attack.

Conjecture 2. *The RSA-OAEP instantiation may even succumb to a more severe OW-CPA attack.*

Sketch. Algorithms similar to Coppersmith's [Cop96] may exist that can solve for m given c. Coppersmith's algorithm inverts the RSA function given certain sufficient information about the input to the RSA function. In this case the information available is the fixed value 0^l in the middle of the padded plaintext. □

Importantly, it not should be ignored that in three standards [Ame07, Ins00, RSA02], the definition of RSA-OAEP is slightly different from the original definition. Arguably, the standardized version is more important for study because it is more likely to be deployed than the academic version. In the version of RSA-OAEP, the message is placed on the very right, while the padding string remains in the middle. There is also a requirement that $G \equiv H$. In this case (1), with the pathological choice of $G \equiv H \equiv 0$, becomes

$$c = (r\|f\|0^k\|1\|m)^e \bmod n, \tag{2}$$

where f is another distinct feature of [RSA02] consisting of the hash of some *label* attached to the ciphertext. The label will generally not be a secret, so we will regard f as known to the adversary, and perhaps even fixed in practice. Since it is the output of a hash function, we could even consider the pathological possibility that $f = 0$, consistent with our choices of $G \equiv H \equiv 0$. Regardless, our IND-CPA attack against the original RSA-OAEP can be made to work, by observing that if $m = 0^l$, then

$$c2^{-3(k+l)} = (r\|f)^3 \bmod n. \tag{3}$$

[3] In fact, a typical choice of t is equal to the length of the hash function whose collision resistance matches the strength of the RSA modulus, which results in a length for t or one sixth or less of the length of c.

Provided that $r\|f < \sqrt[3]{n}$, so that the combined length of r and f is at most third of that of c, then the left hand side will be a perfect cube. The conjectural OW-CPA attack based on a hypothesized variant of Coppersmith's algorithm may be also effective.

Note that these attacks do not appear to exploit the failure in G of standard security properties of hash functions, such as collision resistance and preimage resistance. Indeed, the attacks described do not seem to work for larger values of e, such as $e = 2^{16} + 1$, so the insecurity is not entirely confined to the pathological hash function. Furthermore, concatenating a nonzero leading or trailing one-valued bit before exponentiation also renders some of the attacks useless. Nevertheless, it does demonstrate that if we want security for RSA with $e = 3$ and the existing specifications of OAEP, then some minimal security property of the hash function, especially the masking function G, is necessary. Minimally, we need that the property that the probability of giving a result of all zero bits is negligible.

Some may still inclined to object to this analysis as obvious. After all, it is no surprise that a weak hash implies a weak version of OAEP. If nobody has observed this before in print, an objector may say that this is because it is a trivial, inconsequential fact. This view, however, is a rather myopic one, induced by over reliance on the random oracle model. Excessive exposure to the ideal hash model causes many to turn a blind eye to the security of hash functions. Cryptanalytic muscles have atrophied.[4] Doubtlessly, a proof in the random oracle model is a superb starting point, but it should not be the stopping point. The most logical next step in cryptanalysis after a proof in the ideal hash model is to consider a real world hash. This could entail a re-design of the cryptosystem tailoring it to a proof without idealizing the hash as a random oracle. Or, just as easily, one may seek to work with the existing design, in this case RSA-OAEP, striving to conduct a security analysis with real world hash functions. Arguably, the latter approach is better, since it will leverage the security already established, and perhaps the efficiency gained by working random oracle paradigm. In the case of RSA-OAEP, standardization and deployment should encourage further cryptanalysis of the existing design, rather than starting from scratch with a new design.

Our analysis is a first step, albeit an admittedly obvious step, in the cryptanalysis of RSA-OAEP without hash functions. To provoke deeper cryptanalysis, we assert the following conjecture, which appears non-obvious.

Conjecture 3. *If the RSA problem is hard, and a fixed arbitrary function G has negligible probability of being all zeros on the portion masking the message, then RSA-OAEP is IND-CCA2 secure.*

Note that the purpose of this conjecture is not to challenge cryptologists not to *prove* it, but rather challenge them to *disprove* it. A disproof could consist of some pathological hash function.[5] If this is an obvious issue, then it should be straightforward for a skilled cryptanalyst to find such a pathological hash. Such a cryptanalysis would be a welcome step for the understanding of the security of RSA-OAEP, with respect to reliance of hash function security, as so far we only understand the impossible hypothesis that the hash functions are random oracles. There is a tremendous gap between the understood security and apparent security, and hopefully this gap can be narrowed.

[4] More generally, in the discipline of provable security, it appears that blind eye is turned to necessary security conditions, whether obvious or not.

[5] Strictly speaking, a hash whose pathology varies with the public key does not disprove the conjecture, but it could be a good first step.

3 Impossible Security Reductions for RSA-OAEP

In this section, we will examine what types of reductions can be used prove that RSA-OAEP is OW-CPA secure against adversary that do not treat the hash function as a random oracle. We only focus on certain types of reductions. All the reductions are to the *RSA problem*.

We first recall the basic RSA problem. An *instance* of the RSA problem is a triple of integers (n, e, y). A solution to this instance is an integer x such that $x^e \equiv y \bmod n$. The RSA assumption says that solving the RSA problem is infeasible for the following distribution of instances. The *public exponent* e is fixed. The *public modulus* is $n = pq$, where p and q are randomly chosen primes of a given bit length with the extra condition that $\gcd(e, (p-1)(q-1)) = 1$. The bit lengths are chosen such that finding p from n using the best factoring algorithms known[6] is infeasible. Note that (n, e) is the data typically used as the *public key* in RSA based signature and encryption schemes, such as RSA-OAEP. The *challenge* value y is a random integer, or equivalently a random integer in the range $[0, n-1]$. Throughout this paper, we assume that (n, e, y) has the distribution described above.

3.1 Three Types of Reductions

Definition 1 (Hash-Agnostic Reduction)**.** *An algorithm R that takes input of an RSA public key (n, e) and random integer $y \in [0, n-1]$. The algorithm R is given access to an oracle A. The oracle A is a OW-CPA adversary to RSA-OAEP, when instantiated with public key (n, e) and some fixed pair of hash functions. That is, on input of a valid RSA-OAEP ciphertext c, the oracle A will output the decrypted plaintext m. The algorithm R then outputs an integer $x \in [0, n-1]$. If $x^e \equiv y \bmod n$, then R is successful.*

One may also quantify a hash-agnostic reduction by its success rate and computational cost, and by the success rate and computational cost of its oracle A.

Definition 2 (Hash-Generic Reduction)**.** *An algorithm R that takes input of an RSA public key (n, e) and random integer $y \in [0, n-1]$. The algorithm is given access to three oracles A, G, and H. The oracles G and H are random oracle hash functions. The oracle A is an OW-CPA adversary to the RSA-OAEP, when instantiated with public key (n, e) and hash functions G and H. That is, on input of a valid RSA-OAEP ciphertext C, the oracle A will output the decrypted plaintext. The algorithm R then output an integer $x \in [0, n-1]$. If $x^e \equiv y \bmod n$, then R is successful.*

One may also quantify a hash-generic reduction by its success rate and computational cost, and by the success rate and computational cost of its oracle. The oracle A models an adversary that is specific to G and H. In theory, if specific choices of G and H make RSA-OAEP secure, such an adversary may not exist. Nevertheless the reduction may exist. Indeed, given an explicit algorithm R, it is relatively easy to test if it is truly hash-generic reduction—independently of whether any actual adversaries A exist—because one can simulate A using the factorization of n, and simulate G and H using random values.

We now argue that a hash-generic reduction is a reasonable thing not only to desire but to also consider as something feasible. If RSA-OAEP has very good OW-CPA security in the sense that instantiating it with any random functions G and H makes it secure—which we already have very strong evidence [BR94] for in the ideal hash model—then surely there are ought to be a reduction that is *universal* with respect to hash functions. That is, the security proof ought to work independently of the choice of hash function. Although, a hash-agnostic may seem to achieve

[6] Such as the generalized number field sieve.

this, because it does not try to break the security of the hash function, a hash-agnostic reduction may be an algorithm that only succeeds for a particular hash function. To completely remove dependence of the reduction on any particular hash function, we can restrict the reduction to only have random oracle access to the hash functions.

Definition 3 (Hash-Specific Reduction)**.** *An algorithm R that takes input of an RSA public key (n, e) and random integer $y \in [0, n-1]$ and (random) hash challenge values g and h. The algorithm is given access to oracles A. The oracle A is an OW-CPA adversary to the RSA-OAEP, when instantiated with public key (n, e) and some fixed hash functions G and H. That is, on input of a valid RSA-OAEP ciphertext C, the oracle A will output the decrypted plaintext. The algorithm R then output an integer $x \in [0, n-1]$, and values γ and η. If $x^e \equiv y \bmod n$, then R is successful. If $\Pi(G, g, \gamma) = 1$, then R is successful. If $\Pi(H, g, \eta) = 1$, then R is successful.*

The function Π represents a difficult-to-solve problem characterizing the property of the hash functions G and H that must we prove or assume to be true in order for R to be used in a "security proof". For example, if we want to assume collision resistance in our security proof, then we could take $g = 1$, and the $\gamma = (M, M')$, with Π defined as $\Pi(G, 1, (M, M')) = 1$ if and only if $G(M) = G(M')$. If we want to assume preimage resistance, we can assume define g to a random element of the range of G and define $\Pi(G, g, \gamma) = 1$ if and only if $G(\gamma) = g$. A security proof for RSA-OAEP that uses a hash-specific reduction would need to assume or prove that Π is indeed a hard problem.

Note that, just to be clear, here we are assuming that G and H are fixed, public, unkeyed functions, which is essentially what they need to be implement RSA-OAEP. For properties such as collision resistance, we can adopt the view Rogaway [Rog06] espouses. Because G and H are assumed to be known, the problem defining their security is non-interactive. An oracle for the functions G and H is unnecessary. Note also, the challenge problem Π to solve for the hash functions does not depend on the RSA public key (n, e) corresponding to the RSA problem instance and OW-CPA adversary.

Note that a hash-specific reduction may also be described as three reductions: one for each of the three goals of the hash-specific reductions. More precisely, one would then state a result in which the sum of success probabilities of the three reductions exceeds some minimum. Conversely, it seems likely that any security proof stated in terms of such three reductions could be restated in terms of a single coalesced reduction, such as the hash-specific reduction above.

Note that our formal definitions are reductions against OW-CPA adversaries, which will be the type that we consider primarily. We may also consider secondarily, the generalization to similar reductions against IND-CPA, OW-CCA2 and IND-CCA2 adversaries.

3.2 Metareductions

Theorem 4. *Suppose that R is a hash-agnostic reduction for the OW-CPA security of RSA-OAEP. Then we can find an algorithm M that uses R as an oracle. Algorithm M breaks the OW-CCA2 (and thus IND-CCA2) security of RSA-OAEP.*

Proof. Algorithm M makes no "lunchtime" chosen ciphertext queries between receiving the challenge RSA public key (n, e) and the challenge ciphertext c_b. The challenge ciphertext is an RSA-OAEP encryption under public key (n, e) of a random message m_b in the OW-CCA2 variant. In the IND-CCA2 variant of M, we have the further restriction $m_b \in \{m_0, m_1\}$, where m_0 and m_1 are arbitrary distinct messages chosen by M. Figure 1 illustrates the IND-CCA2 variant of M.

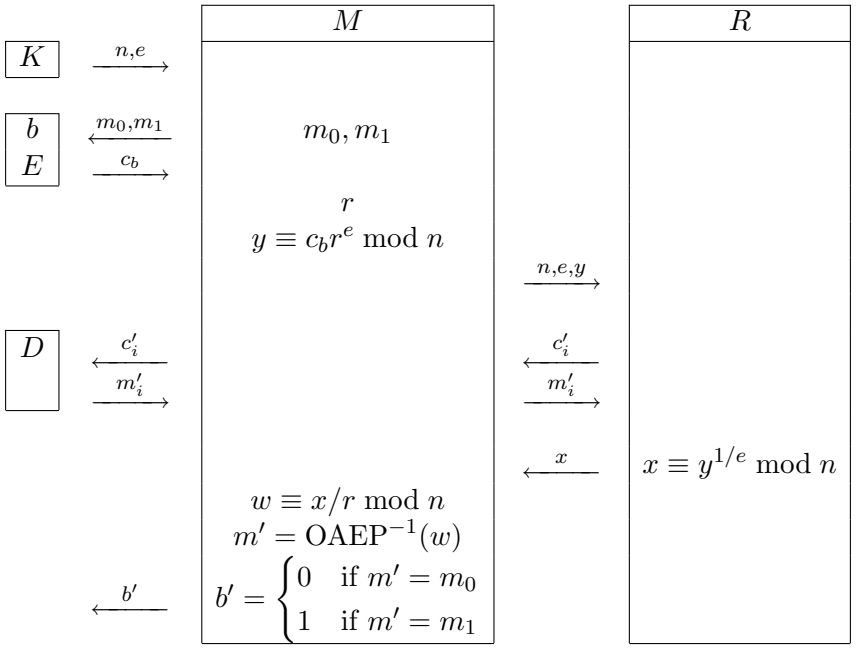

Figure 1: Converting a hash-agnostic OW-CPA reduction into an IND-CCA2 adversary

Algorithm M randomizes the challenge ciphertext to generate a random RSA problem instance (n, e, y), where $y = c_b r^3 \bmod n$, where r is random value chosen by M. This problem instance is given to hash-agnostic reduction algorithm R. By definition, to solve this instance of the RSA problem, a hash-agnostic reduction R expects to use an oracle A, which is a OW-CPA adversary to the given RSA public key (n, e). Therefore, reduction may expect to query its oracle A with some possible RSA-OAEP ciphertext c'_i. Reduction R also expects the oracle A to decrypt these, or reject them as invalid.

Although M does not have the private decryption key, it can decrypt the ciphertext queries c'_i as follows. Because M is playing the role of an OW-CCA2 or IND-CCA2 adversary, M is given an oracle that decrypts any ciphertext except the challenge ciphertext c_b.[7] Accordingly, metareduction M forwards the ciphertext query c'_i from R over to its chosen-ciphertext oracle. (Note that the nature of the ciphertext query has changed: first it was from a good guy R to a bad guy A, but now it is from a bad guy, M, to a good guy, the decryption oracle D.)

The chosen ciphertext oracle will decrypt ciphertext query c'_i to give either a plaintext message m'_i or a rejection (indicating that c'_i was invalid), provided that $c'_i \neq c_b$. Metareduction M may fail if R somehow chooses $c'_i = c_b$. But it easy to see that M reveals no information about c_b to R, so the probability that R determines c_b (not that it is even trying to), is negligible. This is because y is blinded version of c_b, and y has a uniform distribution over RSA challenge instances for the given public key (n, e).

Therefore, except with negligible probability, the chosen ciphertext oracle of M provides a decryption m'_i, which M forwards back to R. From the perspective of R, the metareduction now appears to be a valid A oracle, that is, an OW-CPA adversary. If it is the case that reduction R

[7] Or, to allow for benign malleability of ciphertexts, any ciphertext that does not decrypt to the critical ciphertexts m_0 or m_1.

works by making many queries to its A oracle, then M iterate the procedure above (with variable i being a counter).

Once R has gotten enough access to its alleged OW-CPA oracle A, which is being provided by M by way its chosen ciphertext oracle, the reduction will solve its given RSA problem instance (n, e, y), yielding a value x, such that $x^e \equiv y \bmod n$. When metareduction receives x, it computes $w = x/r \bmod n$. From

$$w^e = x^e r^{-e} = y r^{-e} = c_b r^e r^{-e} = c_b \bmod n, \tag{4}$$

we see that w has the proper OAEP formatting, and when a plaintext message m' is extracted from w, we shall have $w' = m_b$. In the OW-CCA2 variant of M, we are done: M has successfully completed an OW-CCA2 attack. In the IND-CCA2 variant of M, metareduction M chooses the index b such that $m_b = m'$ and successfully completes the IND-CCA2 attack. □

Metareduction M does not rule out, *per se*, a hash-agnostic reduction for the OW-CPA security of RSA-OAEP. It does, however, rule out a hash-agnostic reductions for the OW-CCA2 security of RSA-OAEP. A OW-CCA2 reduction is also a OW-CPA reduction, since an OW-CPA adversary is just a special case of an OW-CCA2 reduction. A proof of OW-CCA2 or IND-CCA2 security for RSA-OAEP would apparently need to fall outside the scope of the reductions that we have formulated.[8]

Note that Paillier and Villar [PV06] prove an equivalent result to ours about hash agnostic reductions. Furthermore, they extend the result to cover reductions that are not key-preserving. Their extension requires, however, a novel, but reasonable, assumption that the ability to solve the RSA problem for one public key does not help to solve the RSA problem for another public key.

Theorem 5. *Suppose that R is a hash-specific reduction for property Π. Then there is an algorithm M_2 that uses R as an oracle to solve the Π problem for G or H or can break the IND-CCA2 security of RSA-OAEP, provided that the hash challenges g and h are independent of the public key.*

Proof. Algorithm M_2 is modification of the algorithm M from the proof of the previous theorem. First of all, note that algorithm M_2 has some challenges to solve that M does not: the g and h challenges for the hash functions G and H. What M_2 does with these hash challenges is to forward them to algorithm R. As before, M_2 randomizes the challenge ciphertext and forwards this to the reduction algorithm R. Figure 2 illustrates M_2.

Note that the public key (n, e) and hash challenges (g, h) given to M are independent. When these challenges are forwarded to R they will remain independent, as required by the definition of R. Now R has its independent challenges. By definition, R uses an OW-CPA oracle A, to solve either its given RSA problem instance or its hash challenge instances. To answer these queries, metareduction M_2 forwards this to its chosen-ciphertext oracle, just as M did.

The responses of the chosen ciphertext oracle are taken by M_2 and forwarded back to R, as was done by M. As before, the decryptions will be valid, and the requirements for R will be complete. Therefore R will succeed in either solving the RSA problem instance or in solving the one of the hash challenges. That is, R will output: x with $x^e \equiv y \bmod n$, or γ with $\Pi(G, g, \gamma) = 1$, or η with $\Pi(H, h, \eta) = 1$.

Given x from R, then M_2, as M did, takes x uses it win the OW-CCA2 or IND-CCA2 game. Given γ from R, then M_2 outputs γ as the solution to the challenge for G. Given η from R, then M_2 outputs η as the solution to the challenge for the hash function H. □

[8] It could for example, rely on a stronger assumption than the hardness of the RSA problem, it could not be key-preserving, or it could escape our result by some other means not contemplated by the author.

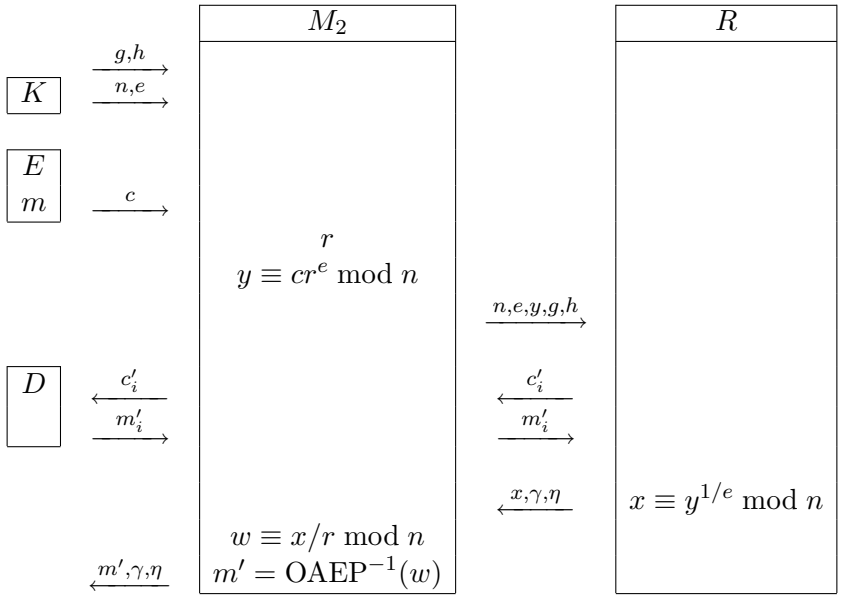

Figure 2: Converting a hash-specific OW-CPA reduction into a OW-CCA2 adversary or hash problem solver

Note that the metareduction M_2 either breaks the OW-CCA2 security of RSA-OAEP or falsifies the security assumptions made about the hash functions G and H in a security proof relying on reduction R. Intuitively, this is slightly less bad than the effect of M, which was to imply a weakness in RSA-OAEP if a OW-CPA security proof is found. Nevertheless, logically, it is equivalent, because M_2 finds a OW-CCA2 attack if the security proof has both a valid reduction and valid assumptions.

Note that the theorem above may be extended to cover reductions that are not key-preserving, using the same technique as Paillier and Villar use in [PV06]. That is, if we assume that an oracle for solving the RSA problem for one choice of RSA public modulus n' does not help to solve the RSA problem for a distinct RSA public modulus n. Here, we will assume that e is fixed throughout. We will allow n' to depend on n by any feasibly computible function. This extra assumption is called *non-malleability* of the RSA problem.

We now summarize the essential idea from [PV06] about making such an extension. Suppose reduction R, when given an RSA challenge (n, e, y), determines a different RSA public key (n', e) for which it can use a OW-CPA adversary with respect to that key (n', e) to solve the RSA challenge (n, e, y). We can use this algorithm R to solve to break the non-malleability of the RSA problem. To answer the ciphertext queries for R, we invoke the available RSA problem solving oracle for RSA public key (n', e).

Theorem 6. *Suppose that R is a hash-generic reduction for RSA-OAEP. There exists an algorithm M_3 that uses R as an oracle to solve the RSA problem.*

Proof. The challenge input for M_3 is an instance (n, e, y) of the RSA problem. Algorithm M_3 forwards this R. By definition, reduction algorithm R expects access to oracles G, H, and A which are, respectively, two random oracle hash functions and an OW-CPA adversary to RSA-OAEP under public key (n, e) when instantiate G and H as the hash functions.

Metareduction M_3, illustrated in Figure 3, answers the oracles queries from R as follows. For a new G query, M_3 responds with a random output value of the correct length. For an old G query, M_3 responds with the previous output. In other words, the G oracle is an unmodified random oracle. Similarly, for a new H query, M_3 responds with a random output value of the correct length. For an old H query, M_3 responds with the previous output. In other words, the H oracle is an unmodified random oracle, just as is G, but with a different length.

For a new A query c, M_3 examines all past G and H queries. For each pair of queries, say r to G and s to H, algorithm M_3 computes

$$c_{r,s} = (s \| (r \oplus H(s)))^e \bmod n \tag{5}$$

If $c_{r,s} = c$, then M_3 computes $t = s \oplus G(r)$, parses this as $t = m \| z$, where y has bit length k. If $z = 0^k$, then M_3 answers the A query for R with the plaintext message m. If $z \neq 0^k$, or if no value of c matches the A-query input, then M_3 responds to the A query of R with the answer that the ciphertext query c is an invalid ciphertext. For an old A query, M_3 responds with its previous output.

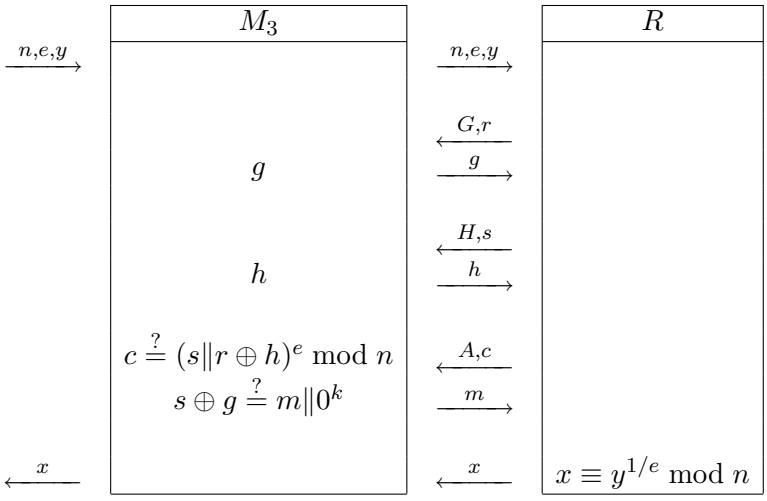

Figure 3: Converting a hash-generic OW-CPA reduction into an RSA problem solver

We next prove that the oracle responses M_3 provides are indistinguishable from true random oracles for G and H and a true adversary A. As already noted, M_3 actually implements G and H as true random oracles, so they are indistinguishable from random oracles.

In the A oracle queries for which M_3 finds match $c_{r,s} = c$, the metareduction M_3 responds with a decryption m which is verifiably valid with respect to everything that R knows about G and H up to the point of making the A query.

Consider an A query for which no match is found. Recall that M_3 responds that this is an invalid ciphertext. There is a possibility, however, that afterwards, R will have at some point made queries r, s to G and H such that $c = c_{r,s}$, in the notation above. In this case, c will have become a valid ciphertext, and the earlier response that M_3 provided will be false. In this event, reduction R is no longer guaranteed to succeed in its goal because its oracles did not operate correctly.

We therefore examine the probability of a later match occurring with c. Note that the value of s in the match is determined by uniquely by c, as follows. Let $d = c^{1/e} \bmod n$, which of course M_3

cannot compute, although it does exist uniquely. Parsing this as $d = s\|u$, where s of the correct bit length, we see s is determined by c. Of course, u is also determined by c, and $r = u \oplus H(s)$. Therefore, $H(s)$ is also determined uniquely by c. Furthermore $t = s \oplus G(r)$ parses as $t = m\|0^k$ for some message k.

Suppose that the first H query with input s was made after the A query of ciphertext c. Now c determines both s and $H(s)$, but since the query s appears for the first time, metareduction M_3 will choose $H(s)$ purely at random, and there is negligible probability that it will be the value determined by c. This event therefore has negligible probability. Essentially, reduction R had to have queried s to G before querying c to A.

Consider now the first G query with input r, supposing that this has occurred after the A query with input c. Note that r is determined uniquely by c and $H(s)$. But when M_3 outputs the value of $G(r)$, it will do so purely at random, because it chooses the G outputs of new queries completely at random. In this case, we will have that $t = s \oplus G(r)$ is a uniformly random bit string. In particular, it will have negligible probability (precisely 2^{-k}) of parsing correctly as $m\|0^k$.

Therefore, except with negligible probability, M_3 will answer the queries made by R in a consistent manner indistinguishable from the oracles required by the definition of R. Therefore R will succeed in solving the RSA problem instance (n, e, y) given to it by M_3. When R reports the answer x to the RSA problem, then M_3 reports x as the answer to the RSA problem it was given. □

3.3 Interpretation

It does seem rather ironic, if not paradoxical, that a certain type of formal proof of a lower grade of security, such as OW-CPA security, implies failure of a higher grade of security, namely OW-CCA2 security. This suggests that, for RSA-OAEP, it is too ambitious either to prove even the most basic security (with a certain class of proof) or to satisfy the higher grade of security. Proofs in the random oracle of the IND-CCA2 security of RSA-OAEP are highly suggestive that RSA-OAEP does indeed have IND-CCA2 security. This in turn, is highly suggestive that RSA-OAEP does not have a security proof, in the standard model, of the most basic kind of security: OW-CPA. Therefore, proofs of advanced security in the random oracle are providing evidence of no proofs of any security at all, in the standard model.

On the other hand, the following less ironic alternative viewpoint can be formulated. The random oracle model is too idealized, and as such, proofs therein should be given little to no weight as evidence of anything. Secondly, the formal definition of IND-CCA2 is quite contrived, and overly strong for any realistic application. What real world decryptors would want to check every ciphertext before before decrypting it, to see if it matches a ciphertext that some other entity encrypted?

A better reason for providing an adversary a chosen ciphertext oracle is actually to model the event that an adversary obtains temporary unauthorized access to the decryptor's decryption device. For example, the device could a tamper resistant module inside of a personal computer. The adversary has injected malware into the personal computer, with the malware being able to access the decryption device module, but not being able to actually extract the private key. The adversary can choose arbitrary ciphertexts, send them to the malware, and then the malware can send back the adversary the decrypted plaintexts. Later, however, the legitimate user, after running a sweep of the laptop, discovers and removes the malware. Of course, any damage already done, such as the adversary learning the content of messages encrypted while the malware was installed, cannot be undone, upon removal of the malware, it should be the case that new ciphertexts are safe from the adversary. The formal definition of IND-CCA1 models this realistic situation well.

In summary, the alternative viewpoint is that IND-CCA1 security should be the true benchmark of public key encryption.

Perhaps not coincidentally, the difficulties in proving the security of RSA-OAEP over the years have only been in the IND-CCA2 setting. The original proof of IND-CCA1 security in the random oracle is recognized as rigorous. Again, perhaps not coincidentally, the impossibility results given here and elsewhere [PV06], apply only to the CCA2 attacks, so say nothing about CCA1. Taking this viewpoint on the greater relevance of CCA1 over CCA2, combined with the lesser relevance of idealized random oracle hash model, then one is led to take the natural objective that RSA-OAEP should be proven IND-CCA1 secure in the standard model, and that one should ignore the question of whether it is actually IND-CCA2 secure.

Recall from §2 pathological hashes for which RSA-OAEP is insecure. Given this result, and the informal interpretation of hash-agnostic or hash-generic reduction as security proofs requiring no security properties form the hash function, one could easily view the metareductions M and M_3 as redundant, and perhaps relatively obvious and unimportant. Nevertheless, there are some technical distinctions, however. For example, both M and M_3 have a wider scope than the result about pathological hash functions, since they apply to presumably strong hash functions, such as SHA-256, not just pathological ones.

3.4 A New Terminology: "Unprovable" Security

The discipline *provable security* entails proving, under certain assumed conjectures, that a cryptographic system is secure against precisely defined classes of adversaries. A specific cryptographic system is often said to have *provable security* or to have a *security proof* if a result in provable security applies to it. Arguably, *security proof* sounds too conclusive, whereas *provable security* more easily admits a softer, and more accurate interpretation, namely that some aspect of security of security has been proven. A deficiency with the term *provable security* is the suffix *-able* in combination with the negative: if some scheme does not *yet* have any provable security results, then the statement that it does not have provable security sounds too strongly as though one will *never* able be to prove anything about its security. Except for this small deficiency, the term *provable security* is fairly useful and understandable.

Just as the reductions used in provable security rule out certain classes of adversaries, metareductions rule out certain classes of reductions. Perhaps, the aptest term for both the discipline of such metareductions, and the ensuing property of cryptographic systems affected by such metareductions, is *unprovable security*.[9] As with the term *provable security*, overly strong interpretations of *unprovable security* are certainly possible, so the term is definitely not perfect by any means. Nevertheless, softer, more accurate interpretations, are also easily admitted: namely, that certain types of security proofs are impossible. This interpretation is analogous the interpretation of provable security, that certain types of adversaries are impossible. One can then make statements that RSA-OAEP enjoys aspects of both provable security and unprovable security. The latter may be a cause for concern, so it would not be unreasonable to seek an alternative that avoids or lessens unprovable security, assuming that one were willing to sacrifice the commitment and effort already put into RSA-OAEP.

[9]The term *undecidable* has been used in formal logic, but as the author is no logician, this term has been forgone here.

4 Analysis of RSA-KEM

An alternative to RSA-OAEP is RSA-KEM [KR07, Sho04, Ame07, BR93]. In essence, RSA-KEM applies the raw RSA public key operation to a random integer. The random integer is hashed with the hash used as a symmetric key to encrypt the message. The primary advantage of RSA-KEM over RSA-OAEP is tighter and simpler proof of IND-CCA2 security. A disadvantage is greater message expansion. Metareductions M, M_2 and M_3 are easily adapted to RSA-KEM. We note also that the RSA-KEM becomes succumbs to an unfixable OW-CPA attack if the hash function with which it is instantiated is a constant function. Indeed, from this, it appears that RSA-KEM requires a greater amount of security from its hash function than RSA-OAEP does. From these two viewpoints, RSA-KEM does not offer a decisive advantage over RSA-OAEP.

5 Further Work

Further work should be done to understand what security properties of the hash functions instantiating RSA-OAEP. Under our analysis, a tremendous gulf exists between the known necessary and sufficient conditions. Narrowing this gap would be desirable. Unfortunately, the metareductions suggest that finding a security proof based on the RSA problem would be difficult. One could try to devise a security proof that does not use a reduction belonging to the three types analyzed here. This would thus entail a creative exercise. A possibly easier strategy would be to abandon the RSA assumption, and instead use a stronger assumption. Taking a cue from the original design rationale of OAEP, we can idealize the trapdoor function, that is, model it by a random oracle, instead of the hash function. Or, if ones wishes to focus on RSA-OAEP, instead of OAEP in general, one can model the RSA component by a generic ring [LR06].

6 Conclusion

More work could be done to determine the security properties of hash function necessary and sufficient to securely instantiate RSA-OAEP. We have shown here that at least very minimal property is necessary: that the masking hash function is fully zero with only negligible probability. That said, we have conjectured that the design of RSA-OAEP is fairly robust, in the sense that no other security properties of the hash function are needed in order for RSA-OAEP to be secure. On the downside, we have provided evidence here that no reasonable property is sufficient, in conjunction with the standard hardness of RSA problem assumption, in the sense that of certain natural types of reductions cannot exist, or would imply that RSA-OAEP is OW-CCA2 insecure.

Acknowledgments

Comments from Alfred Menezes, Scott Vanstone, René Struik, John Goyo, Adrian Antipa, Rob Lambert, Rob Gallant, Nigel Smart, Dan Bernstein, Mark Wooding, Dan Boneh, Alexandra Boldyreva, Jorge Villar, Pascale Paillier, Berry Schoenmakers, Shai Halevi, Mihir Bellare and two anonymous journal reviewers influenced my presentation of this work.

References

[Ame07] American National Standards Institute, *ANS X9.44-2007: Key establishment using integer factorization cryptography*, 2007.

[BF05] Alexandra Boldyreva and Marc Fischlin, *Analysis of random oracle instantiation scenarios for OAEP and other practical schemes*, in Shoup [Sho05], pp. 412–419.

[BF06] ———, *On the security of OAEP*, in Lai and Chen [LC06], pp. 210–225.

[BR93] Mihir Bellare and Phillip Rogaway, *Random oracles are practical: A paradigm for designing efficient protocols*, 1st Annual Conference on Computer and Communications Security, ACM, 1993.

[BR94] ———, *Optimal asymmetric encryption*, Advances in Cryptology — EUROCRYPT '94 (Alfredo De Santis, ed.), LNCS, no. 950, IACR, Springer, May 1994, pp. 92–111.

[BV98] Dan Boneh and Ramarathnam Venkatesan, *Breaking RSA may be easier than factoring*, Advances in Cryptology — EUROCRYPT '98 (Kaisa Nyberg, ed.), LNCS, no. 1403, IACR, Springer, May 1998, pp. 59–71.

[Cop96] Don Coppersmith, *Finding a small root of a univariate modular equation*, Advances in Cryptology — EUROCRYPT '96 (Ueli Maurer, ed.), LNCS, no. 1070, IACR, Springer, May 1996, pp. 155–165.

[DOP05] Yevgneiy Dodis, Roberto Oliviera, and Krzystof Pietrzak, *On the generic insecurity of full domain hash*, in Shoup [Sho05], pp. 449–466.

[FOPS01] Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern, *RSA-OAEP is secure under the RSA assumption*, in Kilian [Kil01], pp. 260–274.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest, *A digital signature scheme secure against adaptive chosen-message attacks*, SIAM J. Comput. **17** (1988), no. 2, 281–308.

[Ins00] Institute for Electrical and Electronics Engineers, *IEEE Std 1363-2000: Specifications for public-key cryptography*, August 2000.

[Kil01] Joe Kilian (ed.), *Advances in cryptology — CRYPTO 2001*, LNCS, no. 2139, IACR, Springer, August 2001.

[KR07] Burt Kaliski and James Randall, *Use of the RSA-KEM key transport algorithm in CMS*, Internet Engineering Task Force, July 2007, Draft.

[LC06] Xuejia Lai and Kefei Chen (eds.), *Advances in cryptology — ASIACRYPT 2006*, LNCS, no. 4284, IACR, Springer, December 2006.

[LR06] Gregor Leander and Andy Rupp, *On the equivalence of RSA and factoring regarding generic ring algorithms*, in Lai and Chen [LC06], pp. 241–251.

[PV05] Pascal Paillier and Damien Vergnaud, *Discrete-log-based signatures may not be equivalent to discrete log*, Advances in Cryptology — ASIACRYPT 2005 (Bimal Roy, ed.), LNCS, no. 3788, IACR, Springer, December 2005, pp. 1–20.

[PV06] Pascal Paillier and Jorge L. Villar, *Trading one-wayness against chosen-ciphertext security in factoring-based encryption*, in Lai and Chen [LC06], pp. 252–266.

[Rab79] Michael O. Rabin, *Digitalized signatures and public-key functions as intractable as factorization*, LCS/TR 212, MIT, 1979.

[Rog06] Phillip Rogaway, *Formalizing human ignorance: Collision-resistant hashing without the keys*, Progress in Cryptology — VIETCRYPT 2006 (Phong Q. Nguyen, ed.), LNCS, no. 4341, Springer, September 2006, http://eprint.iacr.org/2006/281, pp. 221–228.

[RSA02] RSA Laboratories, *PKCS #1 v2.1: RSA cryptography standard*, June 2002.

[Sho01] Victor Shoup, *OAEP reconsidered*, in Kilian [Kil01], pp. 239–259.

[Sho04] ———, *IS 18033-2: Encryption algorithms — part 2: Asymmetric ciphers*, International Standards Organization, December 2004.

[Sho05] Victor Shoup (ed.), *Advances in cryptology — CRYPTO 2005*, LNCS, no. 3621, IACR, Springer, August 2005.

[Wil80] Hugh C. Williams, *A modification of the RSA public-key encryption procedure*, IEEE Transactions on Information Theory **IT-26** (1980), no. 6, 726–729.

[WYY05] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu, *Finding collisions in the full SHA-1*, in Shoup [Sho05], pp. 17–36.