RSA hybrid encryption schemes

Louis Granboulan[*]

École Normale Supérieure
Louis.Granboulan@ens.fr

Abstract. This document compares the two published RSA-based hybrid encryption schemes having linear reduction in their security proof: RSA-KEM with DEM1 and RSA-REACT. While the performance of RSA-REACT is worse than the performance of RSA-KEM+DEM1, a complete proof of its security has already been published. This is indeed an advantage, because we show that the security result for RSA-KEM+DEM1 has a small hole. We provide here a complete proof[1] of the security of RSA-KEM+DEM1. We also propose some changes to RSA-REACT to improve its efficiency without changing its security, and conclude that this new RSA-REACT is a generalisation of RSA-KEM+DEM1, with at most the same security, and with possibly worse performance.

Therefore we show that RSA-KEM+DEM1 should be preferred to RSA-REACT.

1 Motivations

Building a secure asymmetric encryption scheme is one of the main goals of public key cryptography. There have been many proposals, some of them have been provided with proofs of security. The recent discoveries about the security of OAEP show that most proofs are subtle and need to be checked in details.

The numerous studies made on the RSA trapdoor one-way function and its good reputation in the industry makes it probably the most suited basis for building a secure asymmetric encryption scheme that could be widely disseminated as a standard.

This document makes an extensive comparison of RSA-REACT and RSA-KEM+DEM1. It is part of the open evaluation of cryptographic primitives done by the NESSIE consortium.

[*] Part of this work has been supported by the Commission of the European Communities through the IST Programme under Contract IST-1999-12324 (NESSIE).

[1] A complete proof of the general KEM+DEM construction can also been found in the full paper of Cramer and Shoup [5, 7], which was not published at the time of this writing.

2 First assumptions

2.1 Exponent 3 RSA

Generic considerations showing that an exponent e RSA problem can be solved if a proportion $1 - 1/e$ of the input is known show that exponent 3 RSA should not be used if the padding can be insecure.
Moreover, extracting a cubic root is less likely to be equivalent to the factorisation than the generic RSA problem.
For these two reasons, we would not recommend a standard that does not allow greater public exponent than 3.

2.2 Hybrid encryption

There exist schemes that allow to encrypt with RSA without the need of a symmetric cipher (OAEP [2], OAEP+ [9] and SAEP+ [4]). They still need a symmetric primitive, based on a hash function, which is modelled as a random oracle. They have inefficient reductions in their security proofs if the public exponent is greater than 3. They can only encrypt messages significantly smaller than the RSA modulus and the encrypted message has the length of the RSA modulus. [2]
We will focus on hybrid encryption. The main disadvantage of hybrid encryption is that the ciphertext length is bigger than for direct encryption. The great advantage is that the security proof is efficient even for large public exponents.
Two RSA-based schemes fulfill these requirements : RSA-KEM+DEM1 and RSA-REACT.

3 Description of RSA-KEM+DEM1 and RSA-REACT

The public key is an integer n of unknown factorisation and a public exponent e. The private key is the exponent $d = e^{-1} \bmod \phi(n)$. Usually, $n = pq$ with p and q of similar size, but these schemes can be extended to the cases where n is a product of three or more primes of similar size.

3.1 RSA-KEM+DEM1

This scheme is completely described in Shoup's ISO proposal [10]. Its parameters are two functions: $KDF : \{0...n-1\} \to \{0,1\}^{s+l}$ and $MAC :$

[2] For more information about proofs in the random oracle model and efficiency of reductions, see [1, 3].

$\{0,1\}^l \times \{0,1\}^* \to \{0,1\}^k$ and a symmetric encryption scheme $SKE = (SE_K, SD_K)$ of keylength s. Usually $l = s$.

The function KDF should be an *entropy smoothing function* and is modelled as a random oracle. The function MAC should be a *one-time message authentication code.*

Encryption:

```
    input($m$)
$r \leftarrow_{\text{random}} \{0...n-1\}$
$(y, K||K') \leftarrow (r^e \bmod n, KDF(r))$
$c \leftarrow SE_K(m)$
$t \leftarrow MAC_{K'}(c)$
    output($y, c, t$)
```

Decryption:

```
    input($y, c, t$)
$r \leftarrow y^d \bmod n$
$K||K' \leftarrow KDF(r)$
reject if $t \neq MAC_{K'}(c)$
$m \leftarrow SD_K(c)$
    output($m$)
```

3.2 RSA-REACT

This scheme is completely described in Okamoto and Pointcheval's papers [7, 8]. Its parameters are two functions: $KDF : \{0...n-1\} \to \{0,1\}^s$ and $H : \{0...n-1\} \times \{0...n-1\} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^h$ and a symmetric encryption scheme $SKE = (SE_K, SD_K)$ of keylength s.

The function KDF should be an *entropy smoothing function* and is modelled as a random oracle. The function H should be an *entropy smoothing collision resistant hash function* and is modelled as a random oracle.

Encryption:

```
    input($m$)
$r \leftarrow_{\text{random}} \{0...n-1\}$
$(y, K) \leftarrow (r^e \bmod n, KDF(r))$
$c \leftarrow SE_K(m)$
$t \leftarrow H(r, y, m, c)$
    output($y, c, t$)
```

Decryption:

```
    input($y, c, t$)
$r \leftarrow y^d \bmod n$
$K \leftarrow KDF(r)$
$m \leftarrow SD_K(c)$
reject if $t \neq H(r, y, m, c)$
    output($m$)
```

4 Performance comparison

Performance comparison is meaningful only if the symmetric algorithm's sizes and the RSA modulus size have adequate relation. It is still an open problem to find a link between those two parameters[3], but it is of no importance for our comparison: both techniques do the same computations

[3] Lentra and Verheul [6] estimates for equivalent key sizes in 2002 are: a 80-bits security is obtained with 1280-bits RSA, and a 128-bits security is obtained with 3333-bits RSA.
Silverman [11] estimates cost equivalent sizes: a 80-bits security is obtained with 760-bits RSA, and a 128-bits security is obtained with 1620-bits RSA.

modulo n, and have similar requirements for the symmetric encryption scheme SKE.

RSA-REACT has the advantage that its KDF function only outputs s bits while $2s$ bits are needed for RSA-KEM+DEM1. This is only a tiny advantage because the input of KDF has fixed and short length, and there exist good hash functions with output 160 or 256 bits.

RSA-REACT has the disadvantage that the amount of data processed by its symmetric components is slightly above three times the message length, while RSA-KEM+DEM1 only processes twice the message length. For long messages (dozens of kilobytes), RSA-REACT is 50% slower than RSA-KEM+DEM1.

Both RSA-REACT and RSA-KEM+DEM1 can be used for stream processing of messages, but the input of the function H in RSA-REACT needs to alternate fixed sized chunks of m and c.

Also note that MAC are generally faster than hash.

Another (slight) advantage of RSA-KEM+DEM1 is that rejection of invalid messages need only the computation of MAC and not SD.

The conclusion is that RSA-KEM+DEM1 is better than RSA-REACT from a performance point of view.

5 Security comparison

5.1 Security model

An attacker against an encryption scheme can be an *inverter*, a *checker* or a *distinguisher*.

An inverter's goal is, given a ciphertext, to obtain the corresponding plaintext. Its probability of success is taken for a random key and a random plaintext and measures the one-wayness of the scheme.

An checker's goal is, given a plaintext and a ciphertext, to find if the ciphertext encrypts the plaintext. Its probability of success is taken for a random bit (that determines if the ciphertext actually encrypts the plaintext), a random key, a random plaintext and (if the bit is 0) a random ciphertext. [4]

[4] The probability of success of a checker or a distinguisher is $Succ = \Pr[\hat{b} = b]$. Because a random attacker has a probability of success of $\frac{1}{2}$, usually one considers the guessing advantage $Guess = \Pr[\hat{b} = b] - \frac{1}{2}$ or its double, the distinguishing advantage $Dist = \Pr[\hat{b} = 1|b = 1] - \Pr[\hat{b} = 1|b = 0] = 2 \cdot \Pr[\hat{b} = b] - 1$.

A distinguisher's goal is, given a ciphertext and two plaintexts, to find which one has been encrypted. The attacker can choose the pair of plaintexts, the probabilities of success are taken for a random key and a random bit that chooses between the two plaintexts. It measures the semantic security of the scheme. [4]

If the attacker has no access to other information that the ciphertext and eventually the public key, then it is a *passive attack*. If it has access to a decryption oracle, it is a *chosen ciphertext attack*. If it has access to an encryption oracle, it is a *chosen plaintext attack*. [5]

The goal of an attacker against a MAC is, given a plaintext and a tag for some key, to obtain at least another pair (plaintext, tag) for the same key. The attacker's power is limited to the proposal of q_M (plaintext, tag) pairs. [6]

5.2 A unified formulation for proofs of security

Let a scheme have two components X and Y. The proof of security considers an attacker $\mathcal{A}$ against the scheme that runs in time t and succeeds with probability ε. It builds an attacker $\mathcal{B}$ that succeeds if it breaks either component X or component Y. $\mathcal{B}$ runs in time t' and succeeds with probability ε'.

Okamoto and Pointcheval [7, 8] then formulate this security result by saying that, for any $0 < \nu < \varepsilon'$, either there exists an attacker against X with success probability ν, or an attacker against Y with success probability $\varepsilon' - \nu$.

Shoup [10] formulates this security result by saying that $\varepsilon' \leq Succ(A_1) + Succ(A_2)$ where A_1 is an attacker against X and A_2 is an attacker against Y.

We use an intermediate but equivalent formulation: for any $Succ_X$ and $Succ_Y$ such that $\varepsilon' \geq Succ_X + Succ_Y$, then either there exists an attacker against X with success $Succ_X$ or an attacker against Y with success $Succ_Y$.

[5] For an asymmetric encryption scheme the attacker can always encrypt without needing an encryption oracle. For a symmetric encryption scheme, access to an encryption oracle must be explicitely stated.

[6] If the MAC outputs h bits, there is a minimal success probability, that of a random attacker: $Succ_{MAC}(q_M) = \frac{q_M}{2^h}$.

5.3 Claimed results

RSA-KEM+DEM1 The claimed security [10] can be rewritten as: suppose there exists a chosen ciphertext distinguisher running in time t that attacks the hybrid public key encryption scheme with guessing advantage $Guess_{Hyb}$ and at most q_D and q_{KDF} queries to the decryption oracle and to the function KDF (modelled as random oracle). Let n' be a lower-bound on n. Then let t', $Succ_{RSA}$, $Guess_{SKE}$ and $Succ_{MAC}(q_D)$ such that $t' \simeq t$ and $Guess_{Hyb} \leq 2(Succ_{RSA} + \frac{q_D}{n'}) + Guess_{SKE} + Succ_{MAC}(q_D)$. [7]
Then there either exists a passive inverter of RSA running in time t' with success $Succ_{RSA}$, or a passive distinguisher against SKE running in time t' with guessing advantage $Guess_{SKE}$ or an attacker against MAC running in time t' with success probability $Succ_{MAC}(q_D)$

RSA-REACT The claimed security [7, 8] can be rewritten as: suppose there exists a chosen ciphertext distinguisher running in time t that attacks the hybrid public key encryption scheme with distinguishing advantage $Dist_{Hyb}$ and at most q_D, q_{KDF} and q_H queries to the decryption oracle and to the functions KDF and H (modelled as random oracles). Then let t', $Succ_{RSA}$ and $Dist_{SKE}$ such that $t' \leq t + q_{KDF}T_{SKE} + (q_H + q_{KDF})T_{RSAenc}$ and $Dist_{Hyb} \leq 2(Succ_{RSA} + \frac{q_D}{2^h}) + Dist_{SKE}$.
Then there either exists a passive inverter of RSA running in time t' with success $Succ_{RSA}$, or a passive distinguisher against SKE running in time t' with distinguishing advantage $Dist_{SKE}$.

Comparison We can see that the claimed securities of both schemes are similar. There are still some differences.
If everything is written in terms of guessing advantage, then the security is:

RSA-REACT	$Guess_{Hyb} \leq Guess_{SKE} + Succ_{RSA} + \frac{q_D}{2^h}$
RSA-KEM+DEM1	$Guess_{Hyb} \leq Guess_{SKE} + 2 \cdot Succ_{RSA} + \frac{2 \cdot q_D}{n'} + Succ_{MAC}(q_D)$

Because MAC is not modelled as a random oracle, any comparison of the claimed securities of those schemes is fallacious. Nevertheless, since $Succ_{MAC}(q_D) \geq \frac{q_D}{2^h}$, the value $\frac{2 \cdot q_D}{n'}$ cannot been seen as an advantage for RSA-KEM+DEM1. And the success probability of a RSA inverter has a factor of 2 for RSA-KEM+DEM1 security, which might be an advantage for RSA-REACT.

[7] In fact, [10, p52] wrongly says $\frac{nBound}{q_D}$, where it should be $\frac{q_D}{nBound}$. This is probably a typo.

5.4 Proof of security for RSA-REACT

The proof for the generic REACT construction can be found in [8] and is even valid if the underlying encryption scheme is randomised. We rewrite this proof here, specialised to RSA:

> **Outline of the proof.** Suppose that there exists an attacker $\mathcal{A}$ against the semantic security of RSA-REACT, that runs in time t with q_D, q_{KDF} and q_H queries to a decryption oracle, and the two hash functions. Then we build an attacker $\mathcal{B}$ running in time t' that either solves the RSA problem or attacks the semantic security of SKE.
>
> **Description of the attacker $\mathcal{B}$.** The attacker $\mathcal{B}$ makes one call to the distinguisher $\mathcal{A}$ which sends a pair (m_0, m_1) of plaintexts. Then $\mathcal{B}$ transmits this pair and receives a ciphertext $c = SE_K(m_b)$ for unknown and random values b and K. Then $\mathcal{B}$ provides to $\mathcal{A}$ the ciphertext (y, c, t) where y has unknown e-th root and t is random. $\mathcal{B}$ will either extract $r = y^d$ from the queries that $\mathcal{A}$ makes to the oracles, or find the value b. The attacker $\mathcal{B}$ needs to simulate all oracle answers until attacker $\mathcal{A}$ makes a query that allows to find r, or $\mathcal{A}$ returns a bit $\hat{b}$.
>
> Either the attacker $\mathcal{B}$ outputs $\texttt{RSA}(r)$ which means that he solved the RSA problem with answer r, or it outputs $\texttt{SKE}(b)$ which means that he broke the semantic security of SKE and the answer is b.
>
> For all queries r' that $\mathcal{A}$ makes to KDF, the attacker $\mathcal{B}$ outputs $\texttt{RSA}(r')$ if $(r')^e \stackrel{?}{=} y$. For all queries (r', y', m', c') that $\mathcal{A}$ makes to H, the attacker $\mathcal{B}$ outputs $\texttt{RSA}(r')$ if $(r')^e \stackrel{?}{=} y$. If $\mathcal{A}$ returns $\hat{b}$, then the attacker $\mathcal{B}$ outputs $\texttt{SKE}(\hat{b})$.
>
> **Simulating the oracle calls.** If an oracle query does not allow $\mathcal{B}$ to find r, then it must answer a valid value.
>
> - Queries r' to KDF are answered with a new random value K' if r' was not previously asked.
> - Queries (r', y', m', c') to H are answered with a new random value t' if it was not previously asked.
> - Queries (y', c', t') to the decryption oracle are rejected, unless t' was an answer made to a query (r_i, y_i, m_i, c_i) to H and $(y', c') = (y_i, c_i)$. For all queries such that r_i was queried to KDF with answer K_i, the attacker checks if $SE_{K_i}(m_i) = c_i$. In the positive case, m_i is the decrypted value and is returned.
>
> An invalid oracle answer only happens if a query to the decryption oracle is rejected while it should be accepted. This happens if t' is

valid but was not an answer to a query to H. This happens at most with probability 2^{-h} because t' is h bits long.

Running time of $\mathcal{B}$. Each query to KDF needs the computation of $(r')^e$. Each query to H needs the computation of $(r')^e$. Each query to the decryption oracle may need the computation of SE. Therefore, the total time $t' \leq t + (q_{KDF} + q_H)T_{RSAenc} + \min(q_{KDF}, q_D)T_{SKE}$.

Success probability. The probability that there has been at least one invalid oracle answer is $\frac{q_D}{2^h}$. If $\mathcal{A}$ is given valid oracle answers and $\mathcal{A}$ succeeds, then $\mathcal{B}$ succeeds. Its success probability $Succ(\mathcal{B}) \geq (1 - \frac{q_D}{2^h})Succ(\mathcal{A}) \geq Succ(\mathcal{A}) - \frac{q_D}{2^h}$. This proves the inequality $Succ_{SKE} + Succ_{RSA} \geq Succ_{Hyb} - \frac{q_D}{2^h}$, which is equivalent to the formulation of [7]: $Dist_{SKE} + 2 \cdot (Succ_{RSA} + \frac{q_D}{2^h}) \geq Dist_{Hyb}$.

5.5 An improvement of RSA-REACT

We can improve RSA-REACT by minimizing the input of H. We can also withdraw y from the input of H, because it can be recomputed. If m is also not included in the input of H, this new scheme has similar efficiency to RSA-KEM+DEM1, and exactly the same security as RSA-REACT. [8] The new scheme's parameters are two functions: $KDF : \{0...n-1\} \rightarrow \{0,1\}^s$ and $H : \{0...n-1\} \times \{0,1\}^* \rightarrow \{0,1\}^h$ and a symmetric encryption scheme $SKE = (SE_K, SD_K)$ of keylength s.

Encryption:

```
    input($m$)
$r \leftarrow_{\text{random}} \{0...n-1\}$
$(y, K) \leftarrow (r^e \bmod n, KDF(r))$
$c \leftarrow SE_K(m)$
$t \leftarrow H(r, c)$
    output($y, c, t$)
```

Decryption:

```
    input($y, c, t$)
$r \leftarrow y^d \bmod n$
$K \leftarrow KDF(r)$
$m \leftarrow SD_K(c)$
reject if $t \neq H(r, c)$
    output($m$)
```

The proof of security of this scheme is essentially the proof of security of RSA-REACT. Only the oracle simulation needs to be adapted.

Simulating the oracle calls. If an oracle query does not allow $\mathcal{B}$ to find r, then it must answer a valid value.

- Queries r' to KDF are answered with a new random value K' if r' was not previously asked.

[8] The inclusion of y is needed for the generic REACT conversion, because the underlying asymetric encryption scheme may be randomised. The inclusion of m is not needed, even for the generic REACT conversion.

- Queries (r', c') to H are answered with a new random value t' if it was not previously asked.
- Queries (y', c', t') to the decryption oracle are rejected, unless t' was an answer made to a query (r_i, c_i) to H where $c' = c_i$ and $y' = {r_i}^e$. For one query such that r_i was queried to KDF with answer K_i, the attacker computes and returns $SD_{K_i}(c_i)$.

Now we can notice that if we change the notations in RSA-KEM+DEM1 by splitting $K = KDF(r)$ and $K' = KDF'(r)$ and by setting $H(r, c) = MAC_{KDF'(r)}(c)$, then it is the above improved RSA-REACT.

5.6 The proof of security of the hybrid construction KEM+DEM

The proof of RSA-KEM+DEM1 in [10] is split in three parts: the construction of an hybrid scheme from some KEM and some DEM, the proof of security of DEM1, and the proof of security of RSA-KEM. The first two proofs are left to the reader and the explicit running time of the attackers is not included.
The (generic) security result for the hybrid construction in [10, p17] does not explicitely state that the choice of the DEM should be independent of the key of the KEM. We show below a counter-example where an insecure KEM+DEM is built from secure, but related, KEM and DEM.

Definitions.
A DEM (Data Encapsulation Mechanism) is a symmetric scheme, that should be secure (for a random key) against a distinguisher having access to a decryption oracle for that key. Note that access to an encryption oracle is not required.
A KEM (Key Encapsulation Mechanism) is an asymmetric scheme that generates random pairs of plaintext-ciphertext, and that should be secure against a checker having access to a decryption oracle.

Hybrid construction. The private and public keys of the hybrid scheme are those of the KEM. The hybrid encryption of m first calls the KEM to obtain a pair (K, y), then encrypts m with the DEM using K to obtain c. The result is the pair (y, c). The hybrid decryption of (y, c) first calls the DEM to decrypt y and obtain K, then decrypts c with the DEM using K to obtain m.

A counter-example for the generic hybrid construction. We show how to build an insecure hybrid encryption scheme from a secure KEM and a secure DEM. The trick is that the KEM and the DEM will be related in some way that will allow to break the hybrid construction.

KEM. Let (E_{pk}, D_{sk}) be any bijective trapdoor one-way permutation of $\{0,1\}^n$ and KDF_0 be any one-way compression function from $\{0,1\}^n$ to $\{0,1\}^h$, with $h \ll n$. Let also $H : \{0,1\}^h \rightarrow \{0,1\}^{n-h}$ be some one-way function. Let KDF be identical to KDF_0, with the exception that for any value K, we fix $KDF(K||H(K)) = K$. For this new key derivation function, it is easy to compute one of the preimages: $KDF^{-1}(K) = K||H(K)$.

Suppose that the KEM is built as usual: a random r is computed, the output is $(KDF(r), E_{pk}(r))$, Decryption of this KEM computes $KDF \circ D_{sk}$. The attacker's advantage against this KEM is increased by the probability 2^{h-n} that a random r is of the form $K||H(K)$. Because the KEM based of KDF_0 and (E_{pk}, D_{sk}) is secure and $h \ll n$, this KEM is secure.

DEM. Remember that the security of a DEM relies on the fact that the secret key K is kept secret, and that the encryption function DEM_K is secure against a distinguisher having access to a decryption oracle. Suppose that DEM is built such that the one-wayness of the mapping $K \rightarrow DEM_K$ relies on the one-wayness of $E_{pk} \circ KDF^{-1}$. More precisely, we begin with any secure DEM, and we change its definition for one point: for any key K, the encryption of $y_0 = E_{pk} \circ KDF^{-1}(K)$ is the value 0.

This new DEM is exactly as secure as the previous one, because $E_{pk} \circ KDF^{-1}$ is one-way.

Attack of the hybrid scheme. Then the hybrid scheme built from these schemes is not secure. An attacker of the hybrid scheme knows a ciphertext (y, c) that encrypts one of m_0, m_1.

He begins by requesting $(y, 0)$ to the decryption oracle which answers $y_0 = E_{pk} \circ KDF^{-1}(K)$. Then he requests (y_0, c) to the decryption oracle, which will answer the solution m_b. This attack works because $KDF \circ D_{sk}(y) = KDF \circ D_{sk}(y_0)$.

A proof for the construction KEM+DEM. The theorem we prove is that, if the KEM is secure against a checker having access to a decryption oracle for the KEM **and access to a decryption oracle for the**

DEM, [9] and if the DEM is secure against a distinguisher having access to a decryption oracle for the DEM **and access to a decryption oracle for the KEM**, [10] then the resulting hybrid scheme is secure against a distinguisher under chosen ciphertext attack.

> **Outline of the proof.** Suppose that there exists an attacker $\mathcal{A}$ against the semantic security of the hybrid scheme, that runs in time t with q_D queries to a decryption oracle. Then we build an attacker $\mathcal{B}$ running in time t' that will attack the semantic security of DEM.
> **Description of the attacker $\mathcal{B}$.** The attacker $\mathcal{B}$ makes one call to the distinguisher $\mathcal{A}$ which sends a pair (m_0, m_1) of plaintexts. Then $\mathcal{B}$ transmits this pair and receives a ciphertext $c = DEMenc_K(m_b)$ for unknown and random values b and K. Then $\mathcal{B}$ provides to $\mathcal{A}$ the ciphertext (y, c) where y is random. The attacker $\mathcal{B}$ needs to simulate all oracle answers to $\mathcal{A}$, and he can make queries to two oracles that compute — $KEMdec(y')$ if $y' \neq y$ for the first oracle — $DEMdec_K(c')$ if $c' \neq c$ for the other one.
> **Simulating the oracle calls.** When $\mathcal{A}$ queries (y', c'), if $y' \neq y$ then $\mathcal{B}$ asks for $K' = KEMdec(y')$ and returns $m' = DEMdec_{K'}(c')$. If $y' = y$ then $c' \neq c$ and $\mathcal{B}$ asks for $m' = DEMdec_K(c')$ and returns m'.
> Oracle answers for $y' = y$ are invalid, because y was randomly chosen independantly of K, but the probability that it is detected (i.e. the probaility that the fact that these answers are invalid influences the result of $\mathcal{A}$) is at most the best distinguishing advantage against KEM.
> **Running time of $\mathcal{B}$.** Each query to the decryption oracle may need the computation of $DEMdec$, and also needs one call to one of the oracles. Therefore, the total time $t' \leq t + q_D(T_{DEM} + T_{slowest\ oracle})$.
> **Success probability.** The probability that there has been at least one invalid oracle answer is $Dist_{KEM}$. If $\mathcal{A}$ is given valid oracle answers and $\mathcal{A}$ succeeds, then $\mathcal{B}$ succeeds. We have $Succ(\mathcal{B}) \geq (1 - Dist_{KEM})Succ(\mathcal{A}) \geq Succ(\mathcal{A}) - Dist_{KEM}$. This proves the inequality

[9] This condition can easily be improved. Any checker against KEM has to find if a pair (K, y) is valid. Therefore the checker knows the value of K and a decryption oracle for DEM cannot help the attack of KEM.

[10] This condition is mandatory. The counter-example above is built on the lack of this security requirement. Note that a decryption oracle against the KEM can only help an attack of the DEM is the DEM is related to the (secret) key implied by that decryption oracle. Therefore the counter-example is representative of all possible counter-examples.

$Succ_{DEM} \geq Succ_{Hyb} - Dist_{KEM}$, which is equivalent to the formulation of [10, p17]: $Guess_{DEM} + Dist_{KEM} \geq Guess_{Hyb}$.

5.7 Proof of security for DEM1

The straightforward construction of DEM1 is in [10, p19].
Suppose there exists a chosen ciphertext distinguisher against DEM1 running in time t with guessing advantage $Guess_{DEM1}$ and at most q_D queries to the decryption oracle. Let $Guess_{DEM1} \leq Guess_{SKE} + Succ_{MAC}(q_D)$ and $t' \leq T_{MAC}$. Then there exists a passive distinguisher against SKE running in time t' with guessing advantage $Guess_{SKE}$ or an attacker against MAC running in time t' with success probability $Succ_{MAC}(q_D)$

Outline of the proof. Suppose that there exists an attacker $\mathcal{A}$ against the semantic security of the DEM1, that runs in time t with q_D queries to a decryption oracle. Then we build an attacker $\mathcal{B}$ running in time t' that will attack the semantic security of SKE.
Description of the attacker $\mathcal{B}$. The attacker $\mathcal{B}$ makes one call to the distinguisher $\mathcal{A}$ which sends a pair (m_0, m_1) of plaintexts. Then $\mathcal{B}$ transmits this pair and receives a ciphertext $c = SKEenc_K(m_b)$ for unknown and random values b and K. Then $\mathcal{B}$ computes a random K' and computes $t = MAC_{K'}(c)$. He provides to $\mathcal{A}$ the ciphertext (c, t).
Simulating the oracle calls. $\mathcal{B}$ rejects all queries (c', t') from $\mathcal{A}$.
Running time of $\mathcal{B}$. The total time $t' \leq t + T_{MAC}$.
Success probability. The probability that at least one oracle answer is invalid is bounded by $Succ_{MAC}(q_D)$, the probability that a valid MAC can be forged. If $\mathcal{A}$ is given valid oracle answers and $\mathcal{A}$ succeeds, then $\mathcal{B}$ succeeds. We have $Succ(\mathcal{B}) \geq (1 - Succ_{MAC}(q_D))Succ(\mathcal{A}) \geq Succ(\mathcal{A}) - Succ_{MAC}(q_D)$. This proves that $Succ_{SKE} \geq Succ_{DEM1} - Succ_{MAC}(q_D)$, which is equivalent to $Guess_{DEM1} \leq Guess_{SKE} + Succ_{MAC}(q_D)$.

5.8 Proof of security for RSA-KEM+DEM1

RSA-KEM construction. The proof in [10, p52] is complete and shows that if $Guess_{RSA-KEM} \leq Succ_{RSA} + \frac{q_D}{n'}$ and $t' \leq t + q_{KDF}T_{RSAenc}$, then a chosen ciphertext checker against RSA-KEM in time t reduces to a passive RSA inverter in time t'.

Merging all the proofs. We need to adapt the security proof for DEM1 to a proof that DEM1 is still secure when the attacker has access to a decryption oracle for RSA-KEM. Due to the fact that KDF is modelled as a random oracle, a decryption oracle for RSA-KEM cannot help that attacker.

In conclusion, the proven security of RSA-KEM+DEM1 is identical to the claimed security. The running time of the passive RSA inverter is bounded by $t' \leq t + q_{KDF} T_{RSAenc} + q_D T_{SKE} + (q_D + 1) T_{MAC}$.

6 Conclusion

An analysis of the security of RSA-KEM+DEM1 with modelling the function $(r, c) \mapsto MAC_{KDF'(r)}(c)$ as a random oracle proves that its security is at least the same as RSA-REACT. Because of its additionnal security proof where MAC is modelled as a MAC and because of its better performance, RSA-KEM+DEM1 should be preferred to RSA-REACT.

Acknowledgements

I'd like to thank David Pointcheval for his suggestion of removing m from the input of H in REACT, and for fruitful discussions. I'd also like to thank Victor Shoup for his comments, and the anonymous referees of PKC'02, who suggested useful corrections.

References

1. M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCS*, pages 62–73. ACM Press, New York, 1993.
2. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Proc. of EUROCRYPT '94*, LNCS 950, pages 92–111. Springer-Verlag, Berlin, 1995.
3. M. Bellare and P. Rogaway. The exact security of digital signatures: how to sign with RSA and Rabin. *Proc. Eurocrypt'96*, LNCS 1070, pages 399-416, May 1996. Revised version available at `http://www-cse.ucsd.edu/users/mihir/crypto-research-papers.html`.
4. Dan Boneh. Simplified OAEP for the RSA and Rabin functions. In *Advances in Cryptology – CRYPTO 2001*, August 2001. Available at `http://crypto.stanford.edu/~dabo/abstracts/saep.html`.
5. R. Cramer and V. Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. Available at `http://eprint.iacr.org/2001/108/`, December 2001.

6. A. Lentra and E. Verheul. Selecting cryptographic key sizes. *Journal of cryptology*, 14:4, 255-293, Aug. 2001. Applet computing equivalent key sizes available at `http://www.cryptosavvy.com/suggestions.htm`.
7. T. Okamoto and D. Pointcheval. RSA-REACT: An Alternative to RSA-OAEP. *Proc. second open NESSIE workshop*, Egham, Sept. 2001. Available at `http://www.di.ens.fr/~pointche/`.
8. T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. *CT-RSA'2001*, LNCS 2020, pages 208–222, April 2001. Available at `http://www.di.ens.fr/~pointche/`.
9. V. Shoup. OAEP Reconsidered. In *Proc. of CRYPTO '2001*, LNCS 2139, pages 239–259. Springer-Verlag, Berlin, 2001. Available at `http://eprint.iacr.org/2000/060/`.
10. V. Shoup. A proposal for an ISO standard for public key encryption (version 2.0). September 2001. Available at `http://eprint.iacr.org/2001/112/`.
11. R. Silverman. A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths *RSA Labs bulletin*, 13, Apr. 2000. Available online at `http://www.rsasecurity.com/rsalabs/bulletins/bulletin13.html`.