A New Rabin-type Trapdoor Permutation Equivalent to Factoring and Its Applications

Katja Schmidt-Samoa

Technische Universität Darmstadt, Fachbereich Informatik,
Hochschulstr. 10, D-64289 Darmstadt, Germany
samoa@informatik.tu-darmstadt.de

Abstract. Public key cryptography has been invented to overcome some key management problems in open networks. Although nearly all aspects of public key cryptography rely on the existence of trapdoor one-way functions, only a very few candidates of this primitive have been observed yet. In this paper, we introduce a new trapdoor one-way permutation based on the hardness of factoring integers of p^2q-type. We also propose a variant of this function with a different domain that provides some advantages for practical applications. To confirm this statement, we develop a simple hybrid encryption scheme based on our proposed trapdoor permutation that is CCA-secure in the random oracle model.

Keywords: trapdoor one-way permutations, EPOC, hybrid encryption, Tag-KEM/DEM framework

1 Introduction

Informally, a *one-way permutation* is a bijective function that is "easy" to compute but "hard" to invert. If there is some token of information that makes the inversion also an easy task, then we call the function *trapdoor*. Trapdoor one-way permutations are used as building blocks for various kind of cryptographic schemes, e.g. asymmetric encryption, digital signatures, and private information retrieval. There is no doubt that the concept of trapdoor one-way permutations is of particular importance especailly in public key cryptography. Nevertheless, just a relatively small number of promising candidates can be found in the literature. Promising means that the one-wayness of the trapdoor permutation can be reduced to a presumed hard problem such as the integer factorization problem. As not even the pure existence of one-way functions can be proven today[1], this kind of *provable secure* trapdoor permutations is the best alternative solution at present.

1.1 Previous Work

The oldest and still best known candidate trapdoor permutation is the RSA function, i.e. modular exponentiation with exponents coprime to the order of

[1] Interestingly, the current knowledge in complexity theory does not even allow to prove the existence of one-way functions assuming $\mathcal{P} \neq \mathcal{NP}$.

the multiplicative residue group [RSA78]. The factors of the modulus can serve as a trapdoor to invert the RSA function, but the opposite direction is unknown. Thus RSA is not provably equivalent to factoring, and there are serious doubts that this equivalence holds indeed [BV98]. Anyway, as the RSA problem has been extensively studied for decades, nowadays inverting the RSA function is widely accepted as a hard problem itself. Slightly later, M. O. Rabin observed that the special case of modular squaring *can* be reduced to factoring [Rab79]. Modular squaring, however, is not a permutation, it is 4-to-1 (if a two-factor modulus is used). This can be overcome: squaring modulo a Blum integer[2] n is a permutation of the quadratic residues modulo n. The resulting trapdoor permutation is referred to as Blum-Williams function in the literature, and an extension (exponent $2e$, where e is coprime to $\lambda(n)$) is denoted Rabin-Williams function. More factorization-based trapdoor permutations were proposed by Kurosawa et al [KIT88], Paillier [Pai99a,Pai99b], and Galindo et al [GMMV03]. A survey on trapdoor permutations including some less established candidates can be found in [PG97].

1.2 Our Contribution

In this paper, we introduce a rather simple trapdoor one-way permutation equivalent to factoring integers of the shape $n = p^2q$. As many previous candidates, our proposed trapdoor function is also a variant of the RSA function, namely in our case the public exponent is the same as the modulus $n = p^2q$. With the domain $\mathbb{Z}_n^\times$ the function $x \mapsto x^n \bmod n$ is p-to-one, but restricted to the subgroup of n-th residues modulo n, it is indeed a permutation. This property is similar to the Blum-Williams function (where n-th residues are replaced by quadratic residues). Analogical to the quadratic residuosity assumption, we assume that without knowledge of the factorization of n, it is hard to distinguish n-th residues from non-residues, whereas it is efficient if the factors of n are known. However, the restricted domain has some shortcomings that also apply to Blum-Williams and Rabin-Williams functions: in practical applications, the data has to be preprocessed into the set of n-th resp. quadratic residues. Supposably this is one reason why the RSA function (with domain $\mathbb{Z}_n$) is by far more widespread in commercial applications than Rabin-type functions. But fortunately, we can prove that for n of p^2q-type the set of n-th residues is isomorphic to $\mathbb{Z}_{pq}^\times$, thus our proposed trapdoor function is also a bijection between the easy-to-handle domain $\mathbb{Z}_{pq}^\times$ and the set of n-th residues. No such property is known for Rabin-type functions. Indeed, we can show that our proposed trapdoor permutation easily provides practical applications by constructing a hybrid encryption scheme based on Abe et al's Tag-KEM/DEM framework that is chosen-ciphertext (CCA) secure in the random oracle model.

[2] A *Blum integer* is a product of two distinct primes each congruent to 3 modulo 4.

2 A Trapdoor One-way Permutation Equivalent to Factoring

In this section, we introduce a new trapdoor one-way permutation. We also give a short account on its mathematical background in order to deepen the understanding about the special properties of the group $\mathbb{Z}_n^\times$ for n of p^2q-type.

2.1 Notations and definitions

Let n be a positive integer. We write $\mathbb{Z}_n$ for the ring of residue classes modulo n, and $\mathbb{Z}_n^\times$ for its multiplicative group, i.e. the set of invertible elements modulo n. For $x \in \mathbb{Z}_n^\times, \mathrm{ord}_n(x)$ denotes the multiplicative order of x modulo n, i.e. the smallest positive integer k with $x^k = 1 \bmod n$. Furthermore, $\varphi : \mathbb{N} \to \mathbb{N}$ means Euler's totient function.
For any homomorphism h, we denote the kernel and the image with $\ker(h)$ and $\mathrm{im}(h)$, respectively.
As usual, a probability $\Pr(k)$ is called *negligible* if $\Pr(k)$ decreases faster than the reciprocal of any polynomial in k, i.e. $\forall c \exists k_c (k > k_c \Rightarrow \Pr(k) < k^{-c})$.
Unless indicated otherwise, all algorithms are randomized, but we don't mention the random coins as an extra input. If A is a probabilistic algorithm, then $A(y_1, \ldots, y_n)$ refers to the probability space which to the string x assigns the probability that A on input $y_1, \ldots, y_n$ outputs x. For any probability space S, the phrase $x \leftarrow\!\shortmid S$ denotes that x is selected at random according to S. In particular, if S is a finite set, then $x \leftarrow\!\shortmid S$ is the operation of picking x uniformly at random from S.
Finally, we write $|n|_2$ for the bit-length of the integer n.
For the sake of completeness, we formally define the notion of trapdoor one-way permutation.

Definition 1 (Collection of trapdoor one-way permutations). *Let I be a set of indices such that for each $i \in I$ the sets D_i and Let $\mathcal{F} = \{f_i | f_i : D_i \to \tilde{D}_i\}_{i \in I}$ be a family of permutations. Then $\mathcal{F}$ is said to be a* collection of trapdoor one-way permutations *if*

1. *There exists a polynomial p and a probabilistic polynomial time key generator* KeyGen *such that* KeyGen *on input 1^k (the security parameter) outputs a pair (i, t_i) where $i \in \{0,1\}^k \cap I, |t_i|_2 < p(k)$. The data t_i is denoted the trapdoor information of f_i.*
2. *The domains D_i are samplable: There exists a probabilistic polynomial time sampling algorithm* S *that on input $i \in I$ outputs $x \in D_i$ uniformly chosen at random.*
3. *The members of $\mathcal{F}$ are easy to evaluate: There exists a deterministic polynomial time evaluator* Eval *that on input $i \in I, x \in D_i$ outputs $f_i(x)$.*
4. *Inverting the members of $\mathcal{F}$ is easy if the trapdoor information is known: There exists a deterministic polynomial time inverter* Inv *such that for all $x \in D_i$ we have $\mathsf{Inv}(t_i, f_i(x)) = x$.*

5. *Inverting the members of $\mathcal{F}$ is hard if the trapdoor information is unknown: For every probabilistic polynomial time algorithm $\mathcal{A}_I$ the following probability is negligible in k:*

$$\Pr[(i, t_i) \leftarrow 1^k; x \leftarrow D_i : \mathcal{A}_I(f_i(x)) = x].$$

Note that in contrast to strictly mathematical parlance we do not require that permutations are maps onto itself.

2.2 Our proposed trapdoor one-way permutation

Throughout this section, let p, q be primes with $p \nmid q-1$ and $q \nmid p-1$ and define $n = p^2 q$.

All of our constructions are based on the following group homomorphism:

Definition 2 (The homomorphism h). *Let p, q be primes with $p \nmid q-1, q \nmid p-1$ and $n = p^2 q$. Then we define:*

$$\begin{aligned} h : \mathbb{Z}_n^\times &\longrightarrow \mathbb{Z}_n^\times \\ x &\mapsto x^n \bmod n \end{aligned}$$

The reason why we don't use standard RSA moduli is the observation that in $\mathbb{Z}_n^\times$ with $n = p^2 q$ there are members of order p:

Lemma 1. *Let p, q be primes with $p \nmid q-1$ and $n = p^2 q$. Define the set $\mathcal{S}$ as*

$$\mathcal{S} := \{x \in \mathbb{Z}_n^\times \,|\, x = 1 + kpq \text{ for an integer } k, 0 < k < p\}.$$

Then $\mathcal{S}$ consists of exactly the elements of multiplicative order p in $\mathbb{Z}_n^\times$.

Proof. Let x be an element of multiplicative order p in $\mathbb{Z}_n^\times$. Then we have

$$\begin{aligned} x^p = 1 \bmod n &\Rightarrow (x^p = 1 \bmod p \wedge x^p = 1 \bmod q) \\ &\Rightarrow (x = 1 \bmod p \wedge x = 1 \bmod q). \end{aligned}$$

Hence $pq | x-1$ must hold, and we conclude $x \in \mathcal{S}$.
On the other hand, from the binomial expansion formula it is obvious that for all $x \in \mathcal{S}$ we have $x^p = 1 \bmod n \wedge x \neq 1$, thus the assertion follows.

From Lemma 1 we can easily deduce that each element of order p in $\mathbb{Z}_n^\times$ reveals the factorization of n. On this fact we will base the one-wayness of our proposed trapdoor permutations. Next, we analyze the relationship between the homomorphism h and the set $\mathcal{S}$:

Lemma 2. *Let h and $\mathcal{S}$ be defined as above. Then we have*

$$\ker(h) = \{1\} \cup \mathcal{S}.$$

Proof. Note that as p is the only non-trivial common factor of n and $\varphi(n) = p(p-1)(q-1)$, we must have

$$x^n = 1 \bmod n \iff x = 1 \vee \mathrm{ord}_n(x) = p$$

Hence the kernel of h consists of 1 and exactly the elements of multiplicative order p in $\mathbb{Z}_n^\times$, i.e the elements of $\mathcal{S}$ as defined in Lemma 1.

As the magnitude of the kernel of h is exactly p, we obtain

Corollary 1. *The homomorphism h as defined above is p-to-1.*

Now we will prove that h is collision-resistant, because a collision leads to a non-trivial element of $\ker(h)$.

Theorem 1. *For $x, y \in \mathbb{Z}_n^\times$ we have*

$$x^n = y^n \bmod n \iff h(x) = h(y) \iff x = y \bmod pq.$$

Proof. "if": Let $y = x + kpq$ for $k \in \mathbb{Z}$. Then, $(x+kpq)^n = x^n + nx^{n-1}kpq = x^n \bmod n$.
"only if": $x^n = y^n \bmod n$ leads to $xy^{-1} \in \ker(h)$, consequently $xy^{-1} = 1 \bmod pq$ using Lemma 1 and Lemma 2.

Thus we have

Corollary 2. *If factoring integers of the shape p^2q is hard, then the homomorphism h is collision-resistant.*

Proof. Assume that $\mathcal{A}$ is a polynomial time algorithm that on input n determines $x, y \in \mathbb{Z}_n^\times$ with $x \neq y$ and $h(x) = h(y)$. From Theorem 1 we conclude $\gcd(x - y, n) = pq$, which completely reveals the factorization of n.

Next, we formally define the set of n-th residues modulo n.

Definition 3 ($\text{N-R}(n)$)**.** *Let* $\text{N-R}(n) = \{x \in \mathbb{Z}_n^\times | x = y^n \bmod n$ *for a* $y \in \mathbb{Z}_n^\times\} = \mathrm{im}(h)$ *denote the set of the n-th residues modulo n.*

$\text{N-R}(n)$ is a subgroup of $\mathbb{Z}_n^\times$ of order $(p-1)(q-1)$ (as there are exactly $\varphi(pq) = (p-1)(q-1)$ pairwise different n-th residues modulo n, namely the elements $\{x^n \bmod n | x \in \mathbb{Z}_{pq}^\times\}$).

Now we can state the main results of this section:

Theorem 2. *1. Let $I = \{n | n = p^2q, |p|_2 = |q|_2, p \nmid q-1, q \nmid p-1\}$ be a set of indices. The family $\mathcal{F}_{\text{N-R}} = \{f_{\text{N-R}}^{(n)}\}_{n \in I}$ is a collection of trapdoor one-way permutations, where $f_{\text{N-R}}^{(n)}$ is defined as*

$$\begin{aligned} f_{\text{N-R}}^{(n)} : \text{N-R}(n) &\to \text{N-R}(n) \\ x &\mapsto x^n \bmod n. \end{aligned}$$

2. Let I be defined as above. The family $\mathcal{F}_{pq} = \{f_{pq}^{(n)}\}_{n \in I}$ is a collection of trapdoor one-way permutations, where $f_{pq}^{(n)}$ is defined as

$$\begin{aligned} f_{pq}^{(n)} : \mathbb{Z}_{pq}^{\times} &\to \text{N-R}(n) \\ x &\mapsto x^n \bmod n. \end{aligned}$$

In both cases, the trapdoor is the factorization of n and the one-wayness is based on the factorization assumption. For individual members, we omit the superscript (n) whenever it is clear from the context.

Proof. 1. We first show that the $f_{\text{N-R}}$ are indeed permutations. Define $d = n^{-1} \bmod \varphi(pq)$ (note that $\gcd(n, \varphi(pq)) = 1$). Let x be an element of N-R(n), i.e. $x = y^n \bmod n$ for an appropriate $y \in \mathbb{Z}_n^{\times}$. Then we have $(x^n)^d = y^{n^2 d} = x \bmod n$, because of $n^2 d = n \bmod \varphi(n)$ (equality holds modulo p and modulo $\varphi(pq)$). Thus, $x \mapsto x^n \bmod n$ is a permutation of N-R(n). Properties 1. to 3. of Definition 1 are obviously fulfilled. It is clear that d (resp. the factorization of n) can be used as a trapdoor to invert $f_{\text{N-R}}$. The one-wayness (property 5.) is a consequence of Corollary 2: To factor n with access to an oracle that inverts $f_{\text{N-R}}$, we choose an element $x \in \mathbb{Z}_n^{\times}$ at random and query the oracle on $h(x) = x^n \bmod n$. With probability $1 - 1/p$ we have $x \notin$ N-R(n) and the oracle will answer $x' \in$ N-R(n) with $x \neq x' \bmod n$ such that x and x' collide under h. Hence $\gcd(x - x', n) = pq$ reveals the factorization of n.

2. Define d as above. Then it is easy to see that $(f_{pq}(x))^d = x \bmod pq$ holds for all $x \in \mathbb{Z}_{pq}$. Thus f_{pq} is a bijection. The remaining properties can be shown along the lines of the proof given above.

Remark 1. The fact that modular exponentiation with $n = p^2 q$ can be inverted uniquely modulo pq has been implicitely exploited in [Pai99b], where P. Paillier introduced a trapdoor permutation based on the Okamoto-Uchiyama trapdoor mechanism. However, the results and the proof techiques used in [Pai99b] are substantial different from our proposal.

We want to point out the similarities among exponentiation modulo $n = p^2 q$ and Rabin-type modular squaring. In both cases, we have a group homomorphism with a non-trivial kernel. Moreover, one-wayness holds because each non-trivial kernel element reveals the factorization of the modulus. Obviously, the Rabin-Williams permutation on quadratic residues corresponds to our permutation $f_{\text{N-R}}$ on n-th residues. In the case of modular squaring, however, there is no analogue to the bijection f_{pq}. The latter is interesting for practical applications, as no preprocessing into the set of n-th residues is necessary. In particular, f_{pq} can be used to encrypt *arbitrary* strings like keys. We provide an application in Section 3. Further advantages of our proposal are due to the fact that the magnitude of the kernel is larger. For instance, it is possible to construct fail-stop signature schemes [SS04] and trapdoor commitments [SST05] from the homomorphism h. To emphasize the analogy to modular squaring even more, we assume that without knowledge of the factors of n distinguishing N-R(n) from $\mathbb{Z}_n^\times$ is hard[3] (*cf.* the well-known quadratic residuosity assumption). Given p and q, however, deciding n-th residuosity is efficient.

Theorem 3. *For all $x \in \mathbb{Z}_n^\times, x > 1$ we have*

$$x \in \text{N-R}(n) \iff x^{p-1} = 1 \bmod p^2.$$

Proof. See Appendix A.

3 An Exemplary Application: Chosen-ciphertext Secure Hybrid Encryption

Although the concept of public key cryptography (aka asymmetric cryptography) is pretty appealing and has many organizational advantages, secret key cryptography (aka symmetric cryptography) is still much more efficient. Thus for practical applications the combination of both concepts, i.e. hybrid encryption is quite popular.

In this section, we will prove that our proposed trapdoor function is not only of theoretical interest by constructing a simple chosen-ciphertext secure hybrid encryption scheme as an exemplary application. In particular, we show that our novel scheme offers notable advantages compared to the members of the well-known EPOC family [FKM$^+$,OP00]. We choose these schemes as a candidate because they all rely on the Okamoto-Uchiyama trapdoor mechanism that like ours is based on the hardness of factoring integers $n = p^2q$ [OU98]. However, as EPOC-1 has a worse security reduction than EPOC-2 and a similar performance, we focus on EPOC-2 and EPOC-3.

3.1 The Okamoto-Uchiyama trapdoor mechanism and EPOC-2/3

For the sake of self-containedness of this paper, we briefly sketch the Okamoto-Uchiyama trapdoor mechanism (see [OU98] for details). Let n be of the shape $n = p^2q$ for two large primes p, q. Consider the Sylow group $\Gamma_p = \{x \in \mathbb{Z}_{p^2} | x = 1 \bmod p\}$ of $\mathbb{Z}_{p^2}^\times$. The crucial observation is that the L-function defined on Γ_p as For fixed $h \in \text{N-R}(n)$ and $g \in \mathbb{Z}_n^\times$ with $p | \operatorname{ord}_{p^2}(g)$ the Okamoto-Uchiyama encryption of $m \in \{0, 1, \ldots, p-1\}$ is as follows: choose randomness $r \in \mathbb{Z}_n$ and compute $c = g^m h^r \bmod n$. If p is known, then m can be recovered from c in the following way: $c' = c^{p-1} \bmod p^2, g_p = g^{p-1} \bmod p^2, m = L(c')L(g_p)^{-1} \bmod p$. The correctness is deduced from the additive homomorphic properties of the L-function because we have $c' = g_p^m \bmod p^2$ and $g_p \in \Gamma_p$. In [OU98] it is shown that breaking the one-wayness of this scheme is as hard as factoring the modulus.

[3] In case of RSA modulus n, this assumption is known as *Decisional Composite Residuosity Assumption*, and it is the basis for the semantic security of Paillier's homomorphic encryption scheme [Pai99a].

EPOC-3 is obtained by applying the REACT-conversion [OP01] to the Okamoto-Uchiyama encryption scheme. The REACT-conversion builds an CCA-secure (in the random oracle model) hybrid encryption scheme from any one-way-PCA secure asymmetric encryption scheme combined with a symmetric cryptosystem semantically secure against passive attacks. Here, PCA denotes plaintext-checking attack. In this model, the adversary has access to an oracle that on input a message m and a ciphertext c answers if c is a possible encryption of m. Of course, this oracle is only helpful if the encryption is probabilistic, otherwise the adversary can answer the queries himself. Thus, in the deterministic scenario, one-wayness-PCA is equivalent to one-wayness under the weakest attack, i.e. chosen-plaintext-attack (CPA). The benefit of REACT is that the security reduction is tight and that the decryption process is very fast, as only the computation of a single hash-value is necessary to check if the ciphertext is well-formed. In previous conversion techniques, a costly re-encryption was necessary to fulfill this purpose. In the case of EPOC-3, note that although the one-wayness of the Okamoto-Uchiyama encryption is equivalent to factoring integers p^2q, the security of the converted scheme is only based on the probably stronger Gap-High-Residuosity assumption. This is due to the fact that Okamoto-Uchiyama is probabilistic and thus one-way-PCA is not equivalent to one-way-CPA[4].

EPOC-2 is the outcome of combining the Okamoto-Uchiyama encryption and a semantically secure (against passive adversaries) symmetric encryption scheme using the Fujisaki-Okamoto conversion technique [FO99]. In contrast to EPOC-3, EPOC-2 is CCA secure under the p^2q-factoring assumption in the ROM. Although in general the security reduction of the Fujisaki-Okamoto conversion technique is not very tight, Fujisaki observed a tight reduction proof tailored to the special application EPOC-2 [Fuj01]. A disadvantage of EPOC-2 is that in the decryption phase a re-encryption is necessary as an integrity check. For efficiency reasons, this re-encryption is only performed modulo q instead modulo n (accepting a small error probability). Nevertheless, the decryption is less efficient than in case of EPOC-3. There is also a second drawback due to the re-encryption: poor implementation makes EPOC-2 vulnerable against reject-timing attacks [ST03,Den02]. In this attack, the adversary can find the secret key if he is able to distinguish the two different kinds of rejections of invalid ciphertexts (if the enciphered text does not meet length restrictions on the one hand, or if the re-encryption test fails on the other hand). As the re-encryption involves the costly public key operations and hence takes a suitable amount of time, careless implementation makes it possible for an adversary to distinguish between the two cases by measuring the time of rejection.

[4] One could ask why the randomization is not removed before applying REACT (this would lead to the enciphering $c = g^m \bmod n$, and the same decryption as in the original scheme). But note that in this case, we cannot reduce one-wayness to factoring as before, because the distributions of $\{g^m \bmod n | m > p\}$ and $\{g^m \bmod n | m < p\}$ are not necessarily the same.

3.2 The Tag-KEM/DEM framework for hybrid encryption

Beside the technique of applying specific generic constructions to suitable asymmetric and symmetric primitives, a more general solution of hybrid encryption has been introduced by Cramer and Shoup in [CS04]. In this paper, Cramer and Shoup formalize the so-called KEM/DEM framework where KEM is a probabilistic asymmetric *key-encapsulation mechanism*, and DEM is a symmetric encryption scheme (a *data encapsulation mechanism*) used to encrypt messages of arbitrary length with the key given by the KEM. Needless to say, such combinations of public and secret key schemes have been folklore for years, but Cramer and Shoup for the first time gave a rigorously analyzed formal treatment of this subject. Note that a KEM is not the same as a key agreement protocol: the encapsulated key is designated to be used once only, therefore the DEM is only required to be secure in the one-time scenario. For more details on security definitions and requirements the reader is referred to [CS04]. Roughly speaking, if both the KEM and the DEM part are CCA-secure, then the same holds for the whole KEM/DEM scheme.

At this year's Eurocrypt, Abe et al enhanced Cramer and Shoup's framework by introducing the notion of a *Tag*-KEM, which is a KEM equipped with a special piece of information, the tag [AGK05]. In their novel framework for hybrid encryption, this tag as part of an CCA-secure Tag-KEM is assigned to protect the non-malleability of the DEM part. Consequently, for the CCA-security of the whole Tag-KEM/DEM hybrid scheme with an CCA-secure Tag-KEM, it is only required that the DEM part is secure against *passive* adversaries. This is an obvious improvement compared to the KEM/DEM framework, but the flip-side of the coin is that proving a Tag-KEM to be CCA-secure is somewhat more involved than the analogue proof for a "plain" KEM. In [AGK05,AGKS05], the authors provide some generic constructions for Tag-KEMs built from combinations of primitives like KEM, MAC, hash-functions and public key encryption.

In the following, we construct a new Tag-KEM based on our proposed trapdoor permutation and prove its CCA-security in the ROM. Then we show how this leads to a CCA-secure hybrid encryption scheme in the Tag-KEM/DEM framework. Finally, we compare this novel scheme with EPOC-2/3.

3.3 The proposed Tag-KEM

In [AGK05,AGKS05], the notion of Tag-KEM is formally defined. Here – to prevent redundancy – we only give the concrete description of our proposed Tag-KEM that, in our opinion, should be self-explanatory.

TKEM.Gen(1^k): Let k be a security parameter. Choose two distinct k bit primes p, q with $p \nmid q-1, q \nmid p-1$ such that each of $p-1, q-1$ has a large prime factor[5]. Build the product $n = p^2q$, compute $d = n^{-1} \bmod \varphi(pq)$ and

[5] meaning that the bit-length of $p-1$ (resp. $q-1$) divided through its largest prime factor is $\mathcal{O}(\log k)$

define $rLen = 2k - 2$. Select a key derivation function KDF that maps bit-strings into the key-space of the designated DEM and a hash-function H, which outputs bit-strings of length $hashLen$. Return a pair (pk, sk) of public and secret key, where $\mathrm{pk} = (n, rLen, \mathrm{KDF}, H)$ and $\mathrm{sk} = (d, p, q)$.

TKEM.Key(pk): Choose $\omega \in \{0, 1, \ldots, 2^{rLen} - 1\}$ uniformly at random, compute $\mathrm{dk} = \mathrm{KDF}(\omega)$ and return (ω, dk).

TKEM.Enc(ω, τ): Given the key carrier ω and a tag τ, compute $c_1 = \omega^n \bmod n$, $c_2 = H(\omega, \tau)$ and return $\Psi = (c_1, c_2)$.

TKEM.Dec$_{\mathrm{sk}}$(Ψ, τ): Given the encapsulated key Ψ and a tag τ, parse Ψ to c_1, c_2 and compute $r = c_1^d \bmod pq$. If $|r|_2 > rLen$ or $H(r, c_1) \neq c_2$, then return $\perp$, return $\mathrm{KDF}(r)$, otherwise.

In the first step, a key pair is generated. Then a one-time key dk for the DEM part is constructed by applying a key derivation function to a random bit string. Note that the role of KDF is not only to format the bit string according to the key space of the designated DEM, but also KDF is required to destroy all algebraic relations between its input and the encapsulated key. In the security proof, both of KDF and H are modeled as random oracles [BR93]. In the step TKEM.Enc, the one-time key (which in some sense is embedded in ω) is encrypted together with the tag τ. Finally, using TKEM.Dec$_{\mathrm{sk}}$ the one-time key dk can be recovered from the encapsulation Ψ and the tag τ.

Remark 2. In the decapsulation procedure, it is necessary to check if r indeed meets the length requirements ($|r|_2 \leq rLen = 2k - 2$), because otherwise a simple chosen-ciphertext attack can be mounted to obtain the secret factor pq by binary search [JQY01].

CCA-security of a Tag-KEM requires that an adversary with adaptive oracle access to TKEM.Dec$_{\mathrm{sk}}$ has no chance to distinguish whether a given one-time key dk is encapsulated in a challenge (Ψ, τ) or not, even if the tag τ is chosen by the adversary himself. As usual, this is defined via an appropriate game. The following definition is almost verbatim from [AGKS05]:

Definition 4 (Security of Tag-KEM). *Let $\mathcal{K}_D$ be the key space of an appropriate DEM, $\mathcal{O}$ be the decapsulation oracle TKEM.Dec$_{\mathrm{sk}}(.,.)$ and let $\mathcal{A}_T$ be an adversary against Tag-KEM playing the following game:*
GAME.TKEM:
Step 1. $(\mathrm{pk}, \mathrm{sk}) \leftarrow TKEM.Gen(1^k)$
Step 2. $\nu_1 \leftarrow \mathcal{A}_T^{\mathcal{O}}(\mathrm{pk})$
Step 3. $(\omega, \mathrm{dk}_1) \leftarrow TKEM.Key(\mathrm{pk}), \mathrm{dk}_0 \leftarrow \mathcal{K}_D, b \leftarrow \{0, 1\}$
Step 4. $(\tau, \nu_2) \leftarrow \mathcal{A}_T^{\mathcal{O}}(\nu_1, \mathrm{dk}_b)$
Step 5. $\Psi \leftarrow TKEM.Enc(\omega, \tau)$
Step 6. $\tilde{b} \leftarrow \mathcal{A}_T^{\mathcal{O}}(\nu_2, \Psi)$

In Step 6. the adversary is restricted not to query the decapsulation oracle on the challenge Ψ, τ, but queries $\Psi, \tilde{\tau}$ for $\tau \neq \tilde{\tau}$ are permitted. The values ν_1, ν_2 are internal state informations. We define the advantage *of the adversary $\mathcal{A}_T$*

as $\epsilon_{\mathcal{A}_T} = \left|\Pr[\tilde{b} = b] - \frac{1}{2}\right|$ *and* ϵ *as* $\max_{\mathcal{A}_T}(\epsilon_{\mathcal{A}_T})$*, where the maximum is taken over all adversaries modeled as polynomial time Turing machines. Tag-KEM is said to be* CCA-secure*, if* ϵ *is negligible in the security parameter* k*.*

In Appendix B, we prove the following theorem:

Theorem 4. *If factoring integers of the shape* $n = p^2q$ *is hard, then the Tag-KEM defined above is CCA-secure in the random oracle model.*

More formally: If there exists an adversary $\mathcal{A}_T$ *attacking the proposed Tag-KEM in the random oracle model as in Definition 4*

- *in time* t*,*
- *with advantage* ϵ*,*
- *querying the random oracle representing the key derivation function at most* q_K *times,*
- *querying the random oracle representing the hash function at most* q_H *times,*
- *invoking the decapsulation oracle at most* q_D *times,*

then there exists an adversary $\mathcal{A}_{Fact}$ *who factors* $n = p^2q$ *in time* t' *and with advantage* ϵ'*, where*

$$t' \leq t + t_{\mathrm{gcd}}(q_H) + t_{f_{pq}}q_D,$$
$$\epsilon' \geq \left(\epsilon - \frac{q_K}{KLen} - \frac{2q_D}{HashLen} - \frac{q_D}{n + HashLen}\right)\left(1 - \frac{1}{p}\right),$$

where t_{gcd} *is the time needed to perform a* gcd *computation with inputs* $\mathcal{O}(n)$ *and* $t_{f_{pq}}$ *is the time needed to evaluate* f_{pq}*.*

3.4 The proposed hybrid encryption scheme

As before, to avoid lengthy recurrences, we do not review the generic Tag-KEM/DEM framework, but we only describe the concrete hybrid encryption scheme that is obtained when combining our proposed Tag-KEM with an appropriate DEM. The interested reader is referred to [AGKS05,AGK05] for the general treatment. Let $(\mathcal{E}_K^{sym}, \mathcal{D}_K^{sym})$ be any symmetric cryptosystem with key K that is secure in the sense of [AGKS05,AGK05] (roughly speaking, this means that $(\mathcal{E}_K^{sym}, \mathcal{D}_K^{sym})$ is semantically secure against passive adversaries when K is used once only). Assume that the message space of $\mathcal{E}_K^{sym}$ is given as $\{0,1\}^{mLen}$.

Key Generation: The key generation is the same as in TKEM.Gen(.).

Encryption and **decryption** is performed as follows:

$\mathcal{E}_{\mathrm{pk}}(m)$:	$\mathcal{D}_{\mathrm{sk}}(\Psi, \tau)$:
$\omega \leftarrow \{0,1\}^{RLen}$	$(c_1, c_2) := \Psi$
$\mathrm{dk} := \mathrm{KDF}(\omega)$	$r := c_1^d \bmod pq$
$\tau \leftarrow \mathcal{E}_{\mathrm{dk}}^{sym}(m)$	if $\lvert r\rvert_2 > RLen$ or $H(r,\tau) \neq c_2$ return $\perp$
$\Psi := (\omega^n \bmod n, H(\omega, \tau))$	$m := \mathcal{D}_{\mathrm{KDF}(r)}^{sym}(\tau)$
Return (Ψ, τ)	Return m

Note that the DEM ciphertext of the message m encrypted with the encapsulated one-time key serves as the tag. Thus non-malleability of the DEM part is intuitively fulfilled because a CCA-secure Tag-KEM provides integrity of the tag. From the results of [AGK05,AGKS05] we derive

Theorem 5. *If factoring integers of the type p^2q is hard and if $(\mathcal{E}_K^{sym}, \mathcal{D}_K^{sym})$ is one-time secure, then the proposed hybrid encryption scheme is CCA-secure in the random oracle model. More precisely, we have $\epsilon_{hy} \leq 2\epsilon_{KEM} + \epsilon_{DEM}$, where ϵ_{hy}, ϵ_{KEM} and ϵ_{DEM} denote the maximum advantage of an attack against the CCA security of proposed hybrid encryption scheme, against the CCA security of our new Tag-KEM, resp. against the one-time security of $(\mathcal{E}_K^{sym}, \mathcal{D}_K^{sym})$.*

In the above theorem, CCA-security of hybrid encryption is defined in the standard sense, i.e. indistinguishability of ciphertexts under adaptive chosen-ciphertext attacks. Note that the reduction to factoring is tight.

Remark 3. Interestingly, the proposed hybrid encryption scheme is very similar to the scheme obtained from applying the REACT-conversion to f_{pq} [OP01]. The only difference in the REACT-case is that the inputs of the hash function H would be m, ω and c_1. This is a small disadvantage because it decreases the efficiency and it is necessary to recover m before the second integrity check can be performed. As noted above, reject-timing attacks like [ST03,Den02] are possible if the timing difference between two different reject events is too large. However, this threat can be easily dealt with on the implementation level (for instance by using a flag). As the trapdoor function f_{pq} is deterministic, the REACT version of our scheme can be tightly reduced to factoring, too.

3.5 Comparison

In this section, we give a brief comparison of EPOC-2/3 and our proposed hybrid encryption scheme.Table 1 summarizes the most important parameters regarding security and performance. The efficiency of encryption and decryption is measured in modular multiplications, where $MM(k)$ denotes a modular multiplication modulo a k-bit number. We do not distinguish between multiplications and squarings, and we assume that a modular exponentiation with a k bit exponent takes approximately $3k/2$ modular multiplications, whilst a double exponentiation as necessary for performing the Okamoto-Uchiyama encryption takes approximately $7k/4$ modular multiplications using standard techniques. We have not considered exponent recoding techniques, which are applicable in our scheme due to the fixed exponents. Chinese remaindering is taken into account if possible. Hashing, evaluations of the key derivation function and the symmetric key operations are not measured, because these magnitudes are comparable in all schemes. For evaluating the public key sizes, we compare n, g, h on the EPOC-2/3 side with n in our proposed scheme. The secret key sizes are the same (p, g_p for EPOC-2/3, resp. p, d for our proposed scheme). All quantities are measured in terms of the security parameter k (i.e. the bit-length of the prime

Scheme	Assumption	encrypt	decrypt	sk	pk
EPOC-2	FACT	$\geq 7k/2\ MM(3k)$	$\approx 3k/2\ MM(2k) + 7k/4\ MM(k)$	$3k$	$9k$
EPOC-3	Gap-HR	$\geq 7k/2\ MM(3k)$	$\approx 3k/2\ MM(2k)$	$3k$	9k
Proposed	FACT	$\approx 9k/2\ MM(3k)$	$\approx 3k\ MM(k)$	$3k$	$3k$

Table 1. Comparison of important parameters

factors p, q). In case of EPOC-2/3, we assume that $rLen = k$ and $HashLen \geq 2k$ hold (these are the values determining the exponent sizes).
As modular multiplication is quadratic in the length of the modulus, we conclude that our scheme is the most efficient one in decryption, whilst in encryption it is slightly less efficient than EPOC-2/3. Furthermore, the public key is 3 times shorter in our proposed scheme, and the underlying security assumption is optimal (as it is the case for EPOC-2). Another advantage of our scheme is the following: If one-time pad is used for the symmetric part, then the message length in our scheme is $2k$ compared to k in EPOC-2/3. This is because the bandwidth of f_{pq} is twice as large as the bandwidth of the Okamoto-Uchiyama trapdoor function.

4 Conclusion

In this paper we introduced a new simple trapdoor one-way permutation based on the hardness of factoring. As provable secure trapdoor one-way permutations are so rare and nevertheless of outstanding importance in public key cryptography, the development of new candidates is a fundamental issue on its own. Moreover, to constitute the claim that our proposed trapdoor function is not only of theoretical interest, we constructed a novel CCA-secure hybrid encryption scheme as an exemplary application. To do so, we made use of the recently published Tag-KEM/DEM framework for hybrid encryption. We were able to show that already our proposed ad-hoc construction compares favorably with the members of the well-known EPOC family which are based on the same intractability assumption as our proposal.

References

[AGK05] S. Abe, R. Gennaro, and K. Kurosawa. Tag-KEM/DEM: A new framework for hybrid encryption. Cryptology ePrint Archive: Report 2005/027, 2005.

[AGKS05] M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup. Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 128–146. Springer, 2005.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, New York, 1993. ACM Press.

[BV98] D. Boneh and R. Venkatesan. Breaking RSA may not be equivalent to factoring. In Nyberg [Nyb98], pages 59–71.

[CS04] R. Cramer and R. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.*, 33(1):167–226, 2004.

[Den02] A. W. Dent. An implementation attack against the EPOC-2 public-key cryptosystem. *Electronics Letters*, 38(9):412–413, 2002.

[FKM+] E. Fujisaki, T. Kobayashi, H. Morita, H. Oguro, T. Okamoto, S. Okazaki, D. Pointcheval, and S. Uchiyama. EPOC: Efficient probabilistic public-key encryption. Submitted to ISO and NESSIE.

[FO99] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer, 1999.

[Fuj01] E. Fujisaki. Chosen-chipertext security of EPOC-2. Technical report, NTT Corporation, 2001.

[GMMV03] D. Galindo, S. M. Molleví, P. Morillo, and J. L. Villar. A practical public key cryptosystem from Paillier and Rabin schemes. In Yvo Desmedt, editor, *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 279–291. Springer, 2003.

[JQY01] M. Joye, J.-J. Quisquater, and M. Yung. On the power of misbehaving adversaries and security analysis of the original EPOC. In Naccache [Nac01], pages 208–222.

[KIT88] K. Kurosawa, T. Ito, and M. Takeuchi. Public key cryptosystem using a reciprocal number with the same intractability as factoring a large number. *CRYPTOLOGIA*, 12(4):225–233, 1988.

[Nac01] David Naccache, editor. *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings*, volume 2020 of *Lecture Notes in Computer Science*. Springer, 2001.

[Nyb98] Kaisa Nyberg, editor. *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, volume 1403 of *Lecture Notes in Computer Science*. Springer, 1998.

[OP00] T. Okamoto and D. Pointcheval. EPOC-3 - efficient probabilistic public-key encryption, 2000. Submitted to IEEE P1363.

[OP01] T. Okamoto and D. Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In Naccache [Nac01], pages 159–175.

[OU98] T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In Nyberg [Nyb98], pages 308–318.

[Pai99a] P. Paillier. Public key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology – Proceedings of EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 223 – 238. Springer-Verlag, 1999.

[Pai99b] P. Paillier. A trapdoor permutation equivalent to factoring. In *Proceedings of the 1999 International Workshop on Practice and Theory in Public Key Cryptography PKC 1999*, volume 1560 of *Lecture Notes in Computer Science*, pages 219 – 222. Springer-Verlag, 1999.

[PG97] J. Patarin and L. Goubin. Trapdoor one-way permutations and multivariate polynominals. In Y. Han, T. Okamoto, and S. Qing, editors, *Information and Communications Security – ICICS*, volume 1334 of *Lecture Notes in Computer Science*, pages 356–368. Springer, 1997.

[Rab79] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report 212, MIT Laboratory of Computer Science, Cambridge, 1979.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Sho04] V. Shoup. Sequences of games: a tool for taming complexity in security proofs. Available from http://shoup.net/papers/, November 2004.

[SS04] K. Schmidt-Samoa. Factorization-based fail-stop signatures revisited. In Javier Lopez, Sihan Qing, and Eiji Okamoto, editors, *ICICS*, volume 3269 of *Lecture Notes in Computer Science*, pages 118–131. Springer, 2004.

[SST05] K. Schmidt-Samoa and T. Takagi. Paillier's cryptosystem modulo p^2q and its applications to trapdoor commitment schemes. In *Proceedings of MYCRYPT 2005*, volume 3715 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[ST03] K. Sakurai and T. Takagi. A reject timing attack on an IND-CCA2 public-key cryptosystem. In Pil Joong Lee and Chae Hoon Lim, editors, *Information Security and Cryptology – ICISC 2002*, volume 2587 of *Lecture Notes in Computer Science*, pages 359–379, Berlin, 2003. Springer-Verlag.

A Proof of Theorem 3

First we show an auxiliary proposition:

$$x \in \text{N-R}(n) \iff x^{(p-1)(q-1)} = 1 \bmod n.$$

From $p \nmid q-1, q \nmid p-1$ we deduce $\gcd((p-1)(q-1), n) = 1$. Hence there exists $z \in \mathbb{Z}$ with $z(p-1)(q-1) = 1 \bmod n$, leading to $-z(p-1)(q-1)+1 = kn$ for a suitable $k \in \mathbb{Z}$. Thus we have

$$\begin{aligned} x^{(p-1)(q-1)} = 1 \bmod n &\Rightarrow x^{-z(p-1)(q-1)+1} = x \bmod n \\ &\Rightarrow x^{nk} = x \bmod n. \end{aligned}$$

This finishes the proof of the auxiliary proposition, as the opposite direction is straightforward.

Therefore we have the following for $x > 1$:

$$\begin{aligned} x \in \text{N-R}(n) &\iff x^{(p-1)(q-1)} = 1 \bmod n \\ &\iff x^{(p-1)(q-1)} = 1 \bmod p^2 \text{ and } x^{(p-1)(q-1)} = 1 \bmod q && (1) \\ &\iff x^{(p-1)(q-1)} = 1 \bmod p^2 && (2) \\ &\iff x^{(p-1)} = 1 \bmod p^2. && (3) \end{aligned}$$

Note that (2) $\Rightarrow$(1) holds because $x^{(p-1)(q-1)} = 1 \bmod q$ is true for all $x \in \mathbb{Z}_n^\times$. (2) $\Rightarrow$(3) is deduced from $\gcd(q-1, \varphi(p^2) = 1)$.

B Proof of Theorem 4

Proof. We prove Theorem 4 using a series of games. Let $\mathcal{A}_T$ be an attacker against the CCA-security of the proposed Tag-KEM and let y^* be defined as $y^* = f_{pq}(\omega^*)$ for $\omega^* \in \mathbb{Z}_{pq}^{\times}$. Furthermore, let $KLen$ denote the length of valid DEM keys. In each game "Game i", let S_i be the event that the attacker correctly guesses the hidden bit b. Without loss of generality, we assume that the adversary does not ask the same query more than once to the decapsulation oracle.

Game 0: This is the original attack game as defined in Definition 4. Thus we have

$$\epsilon_{\mathcal{A}_T} = |\Pr[S_0] - 1/2|. \tag{4}$$

Game 1: This game is the same as above, with the exception that the random oracles are simulated as follows[6]:

KDF**:** An initially empty list KList is prepared. Given a query x, the following steps are performed: If (x, K) is in KList, then return K. Otherwise randomly generate K', a bit-string of length $KLen$, add (x, K') to KList, and return K'.

H: An initially empty list HList is prepared. Given a query (x, τ), the following steps are performed: If $((x, \tau), hash)$ is in HList, then return $hash$. Otherwise randomly generate $hash'$, a bit-string of length $HashLen$, add $((x, \tau), hash')$ to HList, and return $hash'$.

From the ideal assumptions to the random oracles in Game 0 we deduce that Game 0 and Game 1 are perfectly indistinguishable. Hence, we conclude

$$\Pr[S_0] = \Pr[S_1]. \tag{5}$$

Game 2: This game is the same as above, with the only difference that the challenge $\Psi^* := (y^*, c_2^*)$ is fixed at the beginning of the game, where c_2^* is a randomly generated bit-string of length $HashLen$. It is easy to see that Game 1 and Game 2 are indistinguishable from the adversary's point of view if none of the following events takes place:

- The adversary ever queries the oracle H on ω^*. We call this event $Ask2$.
- The adversary ever queries the KDF oracle on x with $\mathrm{KDF}(x) = \mathrm{dk}_b$, where dk_b is the challenge key given to the adversary in Step 4. The probability of this event is upperbounded by $q_K/KLen$.
- The decapsulation oracle is queried on (Ψ^*, τ), where τ is the tag on which the adversary wishes to be challenged. From the restrictions on the adversary, this is only allowed before Step 6. However, in this case Ψ^* is hidden to the adversary and thus the probability of this event is upperbounded by $q_D/(n + HashLen)$.
- The decapsulation oracle is queried on (Ψ^*, τ'), where $\tau \neq \tau'$ and $H(\omega^*, \tau') = c_2$ holds (i.e. (Ψ^*, τ') is a valid encapsulation). The probability of this event and $\neg Ask2$ is upperbounded by $q_D/HashLen$.

[6] This affects the adversary's oracle queries as well as the generation of the challenge encapsulation.

As the challenge in Game 2 is independent from everything the adversary knows, we conclude

$$\Pr[S_2] = \frac{1}{2}. \tag{6}$$

To compare Game 1 and Game 2, we use the following Lemma from [Sho04]:

Lemma 3 (Difference Lemma). *Let A, B, F be events defined in some probability distribution, and suppose that $A \wedge \neg F \iff B \wedge \neg F$. Then $|\Pr[A] - \Pr[B]| \leq \Pr[F]$.*

From this lemma, we conclude

$$|\Pr[S_1]-\Pr[S_2]| \leq \Pr(Ask2)+q_K/KLen+q_D/(n+HashLen)+q_D/HashLen. \tag{7}$$

Game 3: This game is the same as above, except that instead of having access to the "real" decapsulation oracle, the adversary's queries are responded using the following simulation: If a query Ψ, τ is given, then the following steps are performed: Parse Ψ to (c_1, c_2). Search HList, if there is an entry $((x, \tau), hash)$ with $|x|_2 \leq rLen$ and $f_{pq}(x) = c_1$. If yes and if $hash = c_2$, then return KDF(x) (simulated as above). Otherwise return $\perp$.
Let E_3 be the event that the adversary invokes the decapsulation oracle on a valid query $(c_1, c_2), \tau$ without $f_{pq}^{-1}(c_1)$ having been asked to H. Obviously, we have $\Pr[E_3] \leq q_D/HashLen$. If $\neg E_3$ is true, then Game 2 and Game 3 are perfectly indistinguishable from the adversary's point of view. Define $Ask3$ as the event that the adversary ever queries H or KDF on ω^*. Again using the Difference Lemma, we have

$$|\Pr[Ask_2] - \Pr[Ask_3]| \leq q_D/HashLen. \tag{8}$$

It remains to bound $\Pr[Ask3]$. For that reason, we describe an adversary $\mathcal{A}_{Fact}$ against the p^2q factorization problem with advantage $\epsilon' \geq \Pr[Ask3]$. Adversary $\mathcal{A}_{Fact}$ proceeds at follows: On input n, $\mathcal{A}_{Fact}$ chooses $r \in \mathbb{Z}_n$ at random and computes $y^* = r^n \bmod n$. Then $\mathcal{A}_{Fact}$ lets the adversary $\mathcal{A}_T$ run Game 3 on the public key n. In Step 4, a randomly generated bit-string of length $KLen$ is passed to $\mathcal{A}_T$. When $\mathcal{A}_T$ invokes the encapsulation oracle in Step 5, the challenge $\Psi^* := (y^*, c_2)$ with a randomly generated bit-string c_2 of length $HashLen$ is responded to $\mathcal{A}_T$. The oracle queries are responded by $\mathcal{A}_{Fact}$ exactly as in Game 3 except the following modification of the H oracle:

H: Given a query (x, τ), the following steps are performed: If $((x, \tau), hash)$ is in HList, then return $hash$. Otherwise compute $\gcd(r - x, n)$. If $\gcd(r - x, n)$ is a non-trivial factor of n, return this factor and halt. Randomly generate $hash'$, a bit-string of length $HashLen$, add $((x, \tau), hash')$ to HList, and return $hash'$.

It is obvious that $\mathcal{A}_{Fact}$ perfectly simulates the attack environment of $\mathcal{A}_T$ in Game 5 and finds a non-trivial factor of n with probability $\epsilon' \geq \Pr[Ask3](1-1/p)$ (The attack may fail if $r < pq$ holds, which occurs with probability $1/p$. Also note the events $r < pq$ and $Ask3$ are independent, because the distributions of $\{x^n \bmod n | n \in \mathbb{Z}_n^\times\}$ and $\{x^n \bmod n | n \in \mathbb{Z}_{pq}^\times\}$ are identical.). Thus we conclude

$$\Pr[Ask3] \le \frac{\epsilon'}{1 - 1/p}. \tag{9}$$

Finally, from the equations (4), (5), (6), (7), (8), and (9) the assertion follows.

Remark 4. The gcd-trick exploited in the proof above is due to Fujisaki [Fuj01]. The benefit here is that f_{pq} evaluations are replaced by cheaper gcd-computations.