

# Ciphers Secure Against Related-Key Attacks

Stefan Lucks

University of Mannheim, Germany

<http://th.informatik.uni-mannheim.de/people/lucks/index.html>

**Abstract.** In a related-key attack, the adversary is allowed to transform the secret key and request encryptions of plaintexts under the transformed key. This paper studies the security of PRF- and PRP-constructions against related-key attacks.

For adversaries who can only transform a part of the key, we propose a construction and prove its security, assuming a conventionally secure block cipher is given. By the terms of concrete security, this is an improvement over a recent result by Bellare and Kohno [2]. Further, based on some technical observations, we present two novel constructions for related-key secure PRFs, and we prove their security under number-theoretical infeasibility assumptions.

**Keywords:** related-key attacks, provable security, pseudorandom functions, block ciphers, concrete security

## 1 Introduction

In a related-key scenario, the adversary can partially control the key. It remains secret to the adversary (i.e., she can't *read* it), but she can *choose key transformations*, *modify* the key accordingly, and request *encryptions under the modified keys*. The well-known DES complementation property can be viewed as a vulnerability against a related-key DES-distinguisher.

*One motivation* to study related-key attacks is to evaluate the security of secret-key cryptosystems, namely the security of block ciphers and their “key schedules”, see Knudsen [11] and Biham [3]. Kelsey, Schneier and Wagner [9, 10] presented related-key attacks against several block ciphers, including three-key triple-DES. Today, related-key attacks are a well established tool to evaluate the security of block ciphers, e.g. in the context of the AES [4, 5, 7]. *Another motivation* is the existence of cryptographic schemes, whose security depends on the related-key security of some underlying primitive. Two examples are tweakable block ciphers by Liskov, Rivest and Wagner [13] and RMAC by Jaulmes, Joux and Valette [8]. Knudsen and Kohno [12] pointed out that the triple-DES based variant of RMAC (which had been proposed for standardisation [6]) can be attacked by exploiting the related-key insecurity of triple-DES.

Recently, Bellare and Kohno [1, 2] investigated related-key attacks from a theoretical point of view. They presented an approach to formally handle the notion of related-key attacks. As it turned out, the security of a scheme against related-key attacks greatly depends on the adversary’s capabilities, namely on the set of key transformations available to her.

### 1.1 Focus of this Paper and Overview

In the current paper, we follow the approach from [1, 2], presenting some improved *possibility results*, i.e., constructions for block ciphers (PRPs) and pseudo-random function generators (PRFs), which are provably secure against related-key (RK) adversaries. We first concentrate on “partially-transforming” adversaries, where a part of the secret key is unaffected and remains constant. Then we deal with stronger “ $T^+$ -transforming” adversaries, where the adversary can add a known (or rather chosen) difference to the secret key. See Section 1.2 for the exact definitions. Finally, we provide a short summary. Our main results are:

- For some applications of RK-secure PRFs or PRPs, it would suffice to use a cipher being secure against *partially-transforming* RK adversaries [2]. Section 2 introduces a new construction for secure PRPs provably secure against partially-transforming adversaries. A similar construction and a proof of security can be found in [2]<sup>1</sup>. The concrete complexity (i.e., the upper bound on the adversary’s advantage) shown in [2] turns out to be rather weak, though. The construction in Section 2 allows to prove a better bound.
- In Section 3, we explore equivalent constructions for related-key secure PRFs, and we consider the composition of conventionally secure and related-key secure PRFs. Our observations may be useful as a tool for finding PRFs provably secure against more general related-key adversaries, instead of only partially-transforming ones.
- Section 4 describes two new PRF-constructions. Based on certain number-theoretical assumptions, we prove the security of these constructions against  $T^+$ -transforming adversaries. This is a step towards solving a challenge posed in [1, 2]. To the best of the author’s knowledge, these constructions are the only PRFs so far with a standard-model proof of security against group-induced transformations. Note though, that the assumptions we make are new and not well-studied.
- Sections 5 and 6 conclude the paper with a remark on using a hash function as a tool to ensure related-key security, and with a summary.

### 1.2 Notation and Definitions

We write PRF for a Pseudo-Random Function generator and PRP for a Pseudo-Random Permutation generator (= block cipher). Let  $K$ ,  $D$  and  $R$  be finite

<sup>1</sup> . . . , but it was not included in the Eurocrypt version [1] of that paper.

sets. We write  $\text{Perm}(D)$  for the *set of permutations* over  $D$ . I.e.,  $p : D \rightarrow D$  is in  $\text{Perm}(D)$  if and only if  $p^{-1} : D \rightarrow D$  exists with  $p^{-1}(p(d)) = d$  for all  $d \in D$ . We view a function  $F : K \times D \rightarrow R$  as a *family of functions*  $F(k, \cdot) = F_k(\cdot)$  indexed by  $k \in K$ . If additionally  $D = R$  and  $F_k \in \text{Perm}(D)$  for all  $k \in K$ , then  $F$  is a *family of permutations*, also called a *block cipher*. We write  $F_k^{-1}$  for the inverse of  $F_k$ , i.e., for the decryption function.  $\text{Perm}(K, D)$  denotes the *set of all block ciphers*  $E : K \times D \rightarrow D$ . Below,  $E : K \times D \rightarrow D$  denotes a block cipher encryption function and  $E^{-1}$  denotes its inverse.

Recall the advantage of an adversary in a (conventional) chosen plaintext attack (cf. e.g. [14]): Given  $E$  and an adversary  $A(\langle \text{CP-oracle} \rangle)$  with access to a chosen plaintext oracle, the PRP-advantage of  $A$  when attacking  $E$  is the unsigned difference for  $A$  to distinguish the *real case* from a *random case*:

$$\text{Adv}_E^{\text{prp}}(A) = \left| \Pr[k \in_{\text{R}} K : A(E_k(\cdot)) = 1] - \Pr[g \in_{\text{R}} \text{Perm}(D) : A(g(\cdot)) = 1] \right|.$$

Let  $k \in_{\text{R}} K$  be a secret key. A *related-key oracle*  $E_{\text{rk}(\cdot, k)}(\cdot)$  is an oracle with two inputs, a key transformation  $t : K \rightarrow K$ , and an element  $d \in D$ . Given a query  $(t, d)$ , the related-key oracle responds  $E_{t(k)}(d)$  in the real case, which is to be distinguished from a random case.

**Definition 1 (Security of a PRP under RK attacks).**

Let the block cipher  $E$  and the set of transformations  $T$  be given. The adversary  $A(\langle \text{RK-oracle} \rangle)$  with access to a related-key oracle is a  $T$ -transforming adversary,<sup>2</sup> if she is allowed to choose queries  $(t, d) \in T \times D$  as oracle queries. The PRP-RK-advantage of a  $T$ -transforming adversary  $A$  when attacking  $E$  is

$$\text{Adv}_{T,E}^{\text{prp-rk}} = \left| \Pr[k \in_{\text{R}} K : A(E_{\text{rk}(\cdot, k)}(\cdot)) = 1] - \Pr[k \in_{\text{R}} K, G \in_{\text{R}} \text{Perm}(K, D) : A(G_{\text{rk}(\cdot, k)}(\cdot)) = 1] \right|.$$

Here, the *real case* is the experiment “randomly choose  $k \in K$  and, on a query  $(t, d)$ , respond the value  $E_{t(k)}(d)$ ”. The *random case* is: “Randomly choose  $k \in K$  and  $G \in \text{Perm}(K, D)$ , i.e., a family of  $|K|$  independent random permutations. Respond  $G_{t(k)}(d)$  to oracle queries  $(t, d)$ .” The *attack game* for  $A$  means to distinguish the real from the random case.

Similarly, we define the **security of a PRF under RK attacks**.

We concentrate on the following types of key transformations:

<sup>2</sup> [1, 2] call this a “ $T$ -restricted adversary”. This could be misleading, since RK adversaries appear to be *enhanced* and not restricted, in comparison to conventional adversaries.

**Group-induced transformations:** Let  $(K, \diamond)$  be a group. We define

$$T^\diamond := \{f : K \rightarrow K \mid \exists \delta \in K : f(k) = k \diamond \delta\}.$$

In Section 4, we focus on  $T^+$ , where “+” denotes addition mod  $|K|$ .

**Partial transformations:** Set  $K = K_1 \times K_2$  for non-empty sets  $K_1$  and  $K_2$ .

$T$  is a set of partial transformations, if  $T$  can be rewritten as

$$T = \{t \mid \exists t' \in T' : t(k_1, k_2) = (k_1, t'(k_2))\},$$

where  $T'$  is a set of functions  $K_2 \rightarrow K_2$ .

**Collision free sets of transformations:**  $T$  is *collision-free*, if, for all  $k, k' \in K$ , there exists at most one  $t \in T$  with  $t(k) = k'$ . This is relevant in the context of protocol design, such the previously mentioned RMAC and tweakable block ciphers. Sets of group-induced transformations are collision free. Sets of partial transformations can be collision-free.

## 2 Secure PRPs and Partial Transformations

Set  $K^* := \{0, 1\}^{m+n}$  and consider a set  $T$  of partial transformations:

$$T \subseteq \{t \in \{K^* \rightarrow K^*\} \mid \exists \tau : \{0, 1\}^n \rightarrow \{0, 1\}^n : t(x, y) = (x, \tau(y))\}. \quad (1)$$

Let  $E : \{0, 1\}^m \times \{0, 1\}^n \rightarrow \{0, 1\}^n$  be a block cipher and consider

$$E^0 : \{0, 1\}^{m+n} \times \{0, 1\}^n \rightarrow \{0, 1\}^n, \quad E_{(X,Y)}^0(M) = E_X(M).$$

The adversary has no control over the key  $X$  in use. So if  $E$  is conventionally secure, shouldn't  $E^0$  be secure against  $T$ -transforming adversaries? Consider the following adversary: Choose transformations  $\sigma, \tau : \{0, 1\}^n \rightarrow \{0, 1\}^n$  with  $\sigma(Y) \neq \tau(Y)$  being likely for random  $Y$ . Ask for the encryptions of a random plaintext  $M$  under  $(X, \sigma(Y))$  and  $(X, \tau(Y))$ . In the *real case* (encryption using  $E_0$ ), you get the same answer both times. In the *random case*, if  $\sigma(Y) \neq \tau(Y)$  then  $M$  is encrypted under two independent random permutations – and the two answers are probably different. By comparing the two answers, the adversary can win her attack game.

So we need a different construction. Assume  $E$  (as above) being conventionally secure and consider

$$E' : \{0, 1\}^{m+n} \times \{0, 1\}^n \rightarrow \{0, 1\}^n, \quad E'_{(X,Y)}(M) = E_X(Y \oplus E_X(M)).$$

**Theorem 1 (Security of  $E'$  [2]).** *Let  $K^*$ ,  $T$ ,  $E$ , and  $E'$  be as above. Let  $A'$  be a  $T$ -transforming adversary. We limit the oracle-queries  $(t_i, x_{i,j})$  made by  $A'$  as follows:  $r$  is the number of different transformations  $t_i$  and  $q$  is the highest number of different queries  $(t_i, x_{i,j})$  for any transformation  $t_i$ . (Formally  $r$  and  $q$  are defined as  $r = |\{t_i \in T \mid \exists \text{ query } (t_i, \cdot)\}|$  and  $q = \max_{t_i} |\{x_{i,j} \in \{0, 1\}^n \mid \exists \text{ query } (t_i, x_{i,j})\}|$ .<sup>3</sup>)*

<sup>3</sup> Hence, the actual number of oracle queries  $A$  makes is between  $(r + q - 1)$  and  $rq$ .

For any such RK-adversary  $A'$  attacking  $E'$ , we can construct a chosen plaintext adversary  $A$  attacking  $E$  with

$$\text{Adv}_E^{\text{PP}}(A) \geq \text{Adv}_{T,E'}^{\text{PRP-rk}}(A') - \frac{16r^2q^2 + rq'(q' - 1)}{2^{n+1}},$$

where  $q' = q * \max_{k,k' \in \{0,1\}^{m+n}} |\{ \text{transformations } t \in T \text{ with } t(k) = k' \}|$ , and  $A$  needs the same running time as  $A'$ .

Theorem 1 describes the concrete security of  $E'$ , depending on the security of  $E$ . As usual with concrete security analysis, Theorem 1 should provide a *practically relevant security assurance* for security architects. Intuitively: *The difference between  $\text{Adv}_E^{\text{PP}}(A)$  and  $\text{Adv}_{T,E'}^{\text{PRP-rk}}(A')$  is low (or rather negligible)*. Unfortunately, this only holds for large  $n$ . E.g., with  $E=\text{AES}$  and thus  $n = 128$ , the difference may exceed  $16r^2q^2/2^{n+1}$ , even if  $T$  is collision-free. Assume the AES to be practically secure against chosen plaintext attacks. This means that the advantage of any “reasonable-time” adversary  $A$  against  $E$  is  $\text{Adv}_E^{\text{PP}}(A) = \epsilon \approx 0$ . Allow for  $r = q = 2^{31}$ . Since  $n = 128$ , a “reasonable-time” RK-adversary  $A'$  can exist, who distinguishes  $E'$  from random with  $\text{Adv}_{T,E'}^{\text{PRP-rk}}(A') > 1/2 + \epsilon$ . The number of oracle queries made by  $A'$  can be as low as  $p + q - 1 < 2^{32}$ . Therefore,  $E'$  can be insecure in practice, in spite of Theorem 1 and the (assumed) security of  $E=\text{AES}$ . We don’t claim  $A'$  exists – but we would like to prove its *nonexistence*.

Thus, it is practically interesting to find an improved bound, either for construction  $E'$ , or an alternative construction. Below, we consider

$$E'' : \{0, 1\}^{2n} \times \{0, 1\}^n \rightarrow \{0, 1\}^n, \quad E''_{(X,Y)}(M) = E_{E_X(Y)}(M),$$

where  $E : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$  is conventionally secure, as above. Thus, we have  $K^* = \{0, 1\}^{2n}$ , and  $T$  is a set of partial transformations, as before (Eq. 1). For simplicity, we additionally require  $T$  to be *collision-free*.

**Theorem 2 (Security of  $E''$ ).** *Let  $K^* = \{0, 1\}^{2n}$ ,  $T$  a collision-free set of partial transformations.  $A''$  is a  $T$ -transforming adversary for  $E''$ . Count the transformations in  $A''$ -queries by  $r = |\{t_i \in T \mid \exists \text{ query } (t_i, \cdot)\}|$ . Then a chosen plaintext adversary  $A$  for  $E$  exists, making no more oracle queries than  $A$ , with the same running time as  $A''$  and the advantage*

$$\text{Adv}_E^{\text{PP}}(A) \geq \frac{\text{Adv}_{T,E''}^{\text{PRP-rk}}(A'')}{r + 1}.$$

*Proof.* Assume the *nonexistence* of an adversary  $A$  for  $E$  with the advantage  $a \geq \text{Adv}_{T,E''}^{\text{PRP-rk}}(A'')/(r + 1)$ .

Observe that the oracle queries  $(t_i, d_{i,j})$  from  $A''$  can be viewed as accessing  $r$  different oracles, each implementing a permutation. So in the real case,  $A''$  is querying the  $r$ -tuple

$$P = (E_{E_X(t_1(Y))}, \dots, E_{E_X(t_r(Y))})$$

of permutations over  $\{0,1\}^n$ . An oracle query  $(t_i, d_{i,j})$  is equivalent to asking the  $i$ -th permutation  $p_i = E_{E_X(t_i(Y))}$  for  $p_i(d_{i,j})$ .<sup>4</sup>

Due to the collision-freeness of  $T$ , we have  $t_i(Y) \neq t_j(Y)$  for  $t_i \neq t_j$ , thus  $E_X(t_i(Y)) \neq E_X(t_j(Y))$ . Hence, the  $r$  permutations in  $P$  are defined by  $r$  different keys  $E_X(t_1(Y)), \dots, E_X(t_r(Y))$ . But in the random case, the tuple of permutations can actually be viewed as  $r$  independent random permutations  $E_i^*$  over  $\{0,1\}^n$ . We write this tuple as

$$P_r = (E_1^*, \dots, E_{r-1}^*, E_r^*).$$

The attack game of  $A''$  is equivalent to distinguishing the  $r$ -tuple  $P$  of permutations from  $P_r$ . In doing so, the advantage of  $A''$  is  $\text{Adv}_{T, E''}^{\text{prp-rk}}(A'')$ .

There are other ways to respond to oracle queries  $(t_i, d_{i,j})$ , different from both the real and the random case. Let  $E^*$  be a random permutation, and replace  $E_{E_X(t_i(Y))}(M)$  by  $E_{E^*(t_i(Y))}(M)$ . This way, we get  $r$  independent random values  $Z_i = E^*(t_i(Y))$ , and a new  $r$ -tuple of permutations

$$P_0 = (E_{Z_1}, \dots, E_{Z_r}).$$

Distinguishing  $P_0$  from  $P$  is equivalent to distinguishing  $E_X$  from  $E^*$ , which is exactly the attack game for  $A$ . From the assumption on  $A$ , we conclude that  $A''$  can only distinguish  $P$  from  $P_0$  with an advantage  $< a$ .

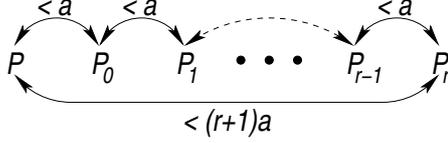
What is the advantage of  $A''$  in distinguishing  $P_0$  from  $P_r$ ? Consider the  $r$ -tuples

$$\begin{aligned} P_1 &= (E_{Z_1}, \dots, E_{Z_{r-2}}, E_{Z_{r-1}}, E_r^*), \\ P_2 &= (E_{Z_1}, \dots, E_{Z_{r-2}}, E_{r-1}^*, E_r^*), \\ &\vdots \\ P_r &= (E_1^*, \dots, E_{r-2}^*, E_{r-1}^*, E_r^*). \end{aligned}$$

If, for any  $i \in \{1, \dots, r\}$ ,  $A''$  could distinguish  $P_{i-1}$  from  $P_i$  with an advantage  $a$ , then  $A''$  could as well distinguish  $E_{Z_i}$  from  $E_i^*$  in the same running time. Since  $Z_i$  is just a random value, and  $E_i^*$  a random function, independent from the other values and functions here, distinguishing  $E_{Z_i}$  from  $E_i^*$  is (again) equivalent to winning the attack game for  $A$ . Thus, the advantage of  $A''$  to distinguish  $P_{i-1}$  from  $P_i$  must be less than  $a$ .

Finally, we put things together:  $A''$  can only distinguish  $P$  from  $P_0$  with an advantage less than  $a$ ,  $A''$  can only distinguish  $P_0$  from  $P_1$  with an advantage less than  $a$ ,  $\dots$ ,  $A''$  can only distinguish  $P_{r-1}$  from  $P_r$  with an advantage less than  $a$ . Consequently, the advantage for  $A$  in distinguishing  $P$  from  $P_r$  must be strictly smaller than  $(r+1)a$ . See the picture below.

<sup>4</sup> This does not restrict the order in which  $A$  makes her oracle queries. After making an oracle query  $(t_i, \cdot)$ , and having seen the answer,  $A''$  may of course freely choose some queries  $(t_{i'}, \cdot)$  for arbitrary values  $i' \in \{1, \dots, i\}$ .



By the definition of  $a$ , we know that  $A''$  can distinguish  $P$  from  $P_r$  with the advantage  $(r + 1)a$ . This contradicts the assumption on  $A$ .  $\square$

Theorem 2 implies that if  $E$  is practically secure and  $r$  is not overwhelmingly large, then  $E''$  is secure, too. As above, consider  $E$ =AES (with a key size of 128 bit) and assume the AES to be practically secure against chosen plaintext attacks. This means that the advantage of any “reasonable-time” adversary  $A$  against  $E$  is  $\text{Adv}_E^{\text{ptp}}(A) = \epsilon \approx 0$ . Restrict  $A''$  to less than  $2^{32}$  oracle queries, thus  $r < 2^{32}$ . In this case, attacking  $E''$  can be at most  $2^{32}$ -times better (i.e. lead to an advantage  $2^{32}$ -times as large), compared to an attack on the AES in the same running time.

We argue that the bound in Theorem 2 is sharp, and hence our result is close to optimal: The attack scenario on  $E''$  allows the adversary to see encryptions under  $r$  different keys. Consider an exhaustive key-search attack against  $E''$ , trying to find any of the  $2^r$  keys and compare it with an exhaustive key-search attack against  $E$ . The chances of successfully attacking  $E''$  are  $2^r$ -times better than the chances of successfully attacking  $E$ .

### 3 Equivalence and Composition of PRF Constructions

In this section, we make some technical observations. While rather simple, these observations may nevertheless be useful both for understanding the phenomenon of related-key security, and for designing ciphers provably secure against related-key attacks.

Let  $F$  be a function  $F : K \times D \rightarrow R$  (which equivalently is a family of functions  $D \rightarrow R$ ). For  $F$ , we consider a set of transformations  $T \subseteq \{K \rightarrow K\}$ . Let  $D = D' \times D''$  (where even  $|D'| = 1$  or  $|D''| = 1$  is allowed). We can rewrite  $F$  as a function  $F'$  with

$$F' : \overbrace{(K \times D')}^{K'} \times D'' \rightarrow R.$$

Equivalently,  $F'$  is a family of functions  $D'' \rightarrow R$ . We consider the set  $T'$  of transformations:

$$T' = \{t' : K' \rightarrow K' \mid \exists t \in T, d' \in D' : t'(k, x) = (t(k), d')\}.$$

**Theorem 3 (Equivalence of  $F$  and  $F'$ ).**

1. Let  $A$  be a  $T$ -transforming RK adversary for  $F$ . A  $T'$ -transforming RK adversary  $A'$  for  $F'$  exists, with the same running time, the same number of oracle queries and the same advantage.
2. Let  $A'$  be a  $T'$ -transforming RK adversary for  $F'$ . A  $T$ -transforming RK adversary  $A$  for  $F$  exists, with the same running time, the same number of oracle queries and the same advantage.

*Proof.* Consider claim 1 and the  $T$ -transforming RK adversary  $A$  for  $F$ .  $A$ 's queries are of the form  $(t, (d', d'')) \in T \times (D' \times D'')$ . Our  $T'$ -transforming adversary  $A'$  for  $F'$  is identical to  $T$ , except that each query  $(t, (d', d''))$  is replaced by the equivalent query  $((t, d'), d'') \in K' \times D''$ . Thus,  $T'$  makes exactly the same number of oracle queries, needs the same running time and wins the attack game with the same advantage as  $T$ . Proving claim 2 is similar.  $\square$

In the context of Theorem 3, we even allowed  $|D''| = 1$ . In this case,  $F' : K' \times D'' \rightarrow R$  can, of course, be rewritten as  $F' : K' \rightarrow R'$ . This apparently trivial case is worth investigating. By means of some function  $F'' : R' \times D \rightarrow R$ , we define a composed function

$$F : K' \times D \rightarrow R, \quad F_k(d) = F''_{F'(k)}(d).$$

**Theorem 4 (Security of composed function  $F$ ).** *Let  $A$  be a  $T$ -transforming adversary for  $F$ . We can construct a  $T$ -transforming adversary  $A'$  for  $F'$ , and a chosen ciphertext adversary  $A''$  for  $F''$ , such that neither the running time of  $A'$ , nor the running time of  $A''$  exceed the running time of  $A$ , and the following condition holds:*

$$\text{Adv}_{T,F}^{\text{prf-rk}}(A) \leq \text{Adv}_{T,F'}^{\text{prf-rk}}(A') + \text{Adv}_{F''}^{\text{prf}}(A''). \quad (2)$$

Neither  $A'$  nor  $A''$  makes more oracle queries than  $A$ .

*Proof.* Let  $k \in K'$  be a random key, unknown to the adversary  $A$ .  $A$  distinguishes between the events Real and Random

- Real: All responses to oracle queries  $(\delta, d) \in K' \times D$  are generated as

$$F''_{F'(k+\delta)}(d).$$

- Random: Let  $F^* : K' \times D \rightarrow R$  be a random function. All responses to oracle queries  $(\delta, d) \in K' \times D$  are generated as  $F^*(\delta, d)$ .

We introduce a third event,  $K'$ -Random:

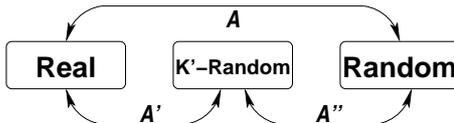
- $K'$ -Random: Let  $F^{**} : K' \rightarrow R'$  be a random function. All responses to oracle queries  $(\delta, d) \in K' \times D$  are generated as

$$F''_{F^{**}(\delta)}(d).$$

Distinguishing Real from  $K'$ -Random means to distinguish  $K'$  from a random function. This is exactly the task  $A'$  is supposed to do. Thus, we can turn  $A$  into  $A'$  without increasing either running time or number of queries.

Similarly, we observe that if we can distinguish  $K'$ -Random from Random, we can mount a chosen plaintext attack against  $F''$  and thus turn  $A$  into  $A''$ , again with the same running time and number of queries.

What about Condition 2? See the picture below.



With  $A'$  and  $A''$  as described above, condition 2 holds. □

In short, Theorem 4 implies that if  $F'$  is practically secure against  $T$ -transforming adversaries and  $F''$  is practically secure against chosen ciphertext adversaries, then  $F$  must be practically secure against  $T$ -transforming adversaries. This provides us with a *tool* for finding RK-secure PRFs, or proving their existence under reasonable assumptions:

Let  $T$ ,  $K'$ ,  $D$  and  $R$  be given. We are searching for

$$F : K' \times D \rightarrow R,$$

practically secure w.r.t.  $T$ -transforming adversaries. It is sufficient to choose an appropriate  $R'$  and a conventionally secure PRF  $F'' : R' \times D \rightarrow R$ , and then search for a function

$$F' : K' \rightarrow R',$$

practically secure w.r.t.  $T$ -transforming adversaries.

The idea is that finding  $F'$  may be less difficult than finding  $F$  directly. In the next section, we concentrate on finding appropriate functions  $F'$ .

## 4 PRFs and Group-Induced Key Transformations

In this section, we describe two PRFs and prove their security against  $T^+$ -transforming adversaries under certain assumptions from algorithmic number-theory. This is a step towards solving a “challenging problem” posed by Bellare and Kohno [1, 2]. Note though, that our assumptions are non-standard and have not much been studied much, so far. It remains an open problem, to describe some PRFs or PRPs and reduce their security against  $T^+$ -transforming adversaries to some cryptographic standard assumption, such as Decisional Diffie-Hellman, Quadratic Residuousity, or others.

#### 4.1 The RSA-based PRF $F'_{\text{RSA}}$

Let  $N$  be the product of two large random primes. We define the function

$$F'_{\text{RSA}} : \mathbb{Z}_N \rightarrow \mathbb{Z}_N, \quad F'_{\text{RSA}}(k) := k^N \bmod N.$$

To evaluate the security of  $F'_{\text{RSA}}$ , we define an appropriate problem:

**Definition 2.** *Let  $N$  be the product of two large random primes. Let  $R$  be a random value in  $\mathbb{Z}_N$ . Define*

$$f(x) = (x + R)^N \bmod N. \tag{3}$$

**Interactive Dependent RSA Problem (IDRP):** *Distinguish  $f$  from a random function  $\mathbb{Z}_N \rightarrow \mathbb{Z}_N$ . The distinguisher is given  $N$  and oracle access to the function (but neither  $R$  nor the factors of  $N$ ).*

**Interactive Dependent RSA Assumption:** *The IDRP is infeasible.*

Some remarks on the IDRP:

1. We can make the above scheme and the IDRP more “RSA-like” by choosing any large RSA-exponent  $e$  (that means,  $e$  and  $\varphi(N)$  have no common divisors) and rewriting Equation 3 by  $f(x) = (x + R)^e \bmod N$ . If  $e$  is small, however, this variant of the IDRP is feasible [15].
2. The IDRP can be seen as a generalisation of Pointcheval’s Dependent-RSA problem [15]: For independent random  $x, y \in \mathbb{Z}_N$ , distinguish the random pair  $(x, y)$  from the pair  $(x^e \bmod N, (x + 1)^e \bmod N)$ . Given an efficient algorithm to solve the Dependent-RSA problem, we could efficiently solve the IDRP.

**Theorem 5 (Security of  $F'_{\text{RSA}}$ ).** *Under the Interactive Dependent RSA Assumption, no efficient  $T^+$ -transforming adversary with significant advantage for  $F'_{\text{RSA}}$  can exist.*

*Proof.* The proof is quite straightforward. Assume  $F'_{\text{RSA}}$  to be insecure. Then an efficient  $T^+$ -transforming adversary  $A$  for  $F'_{\text{RSA}}$  wins the following attack game with significant advantage:

- choose  $\delta \in \mathbb{Z}_N$ , define a key transformation  $t_\delta(k) = k + \delta \bmod N$ ,
- ask for  $F'_{\text{RSA}}(t_\delta(k)) = F'_{\text{RSA}}(k + \delta) = (k + \delta)^N \bmod N$  (with  $k$  unknown),
- and, after repeating the above two steps a couple of times, distinguish the results from the outputs of a random function.

This attack game is equivalent to solving the IDRP. □

## 4.2 The Diffie-Hellman based PRF $F'_{\text{DH}}$

In [1, 2], Bellare and Kohno consider two PRF-constructions which are provably secure against chosen plaintext adversaries under the Decisional Diffie-Hellman assumption. Both turn out to be insecure against additive-transforming adversaries. How can we define a Diffie-Hellman based PRF, with plausible hope for security against additive-transforming adversaries?

Let  $P$ ,  $P_2$ ,  $P_3$  and  $P_4$  be primes,  $P = 2P_2 + 1$ ,  $P_2 = 2P_3 + 1$ ,  $P_3 = 2P_4 + 1$ . Let  $g$  be an element of order  $P_2$  in  $\mathbb{Z}_P^*$ . Let  $g_2$  be an element of order  $P_3$  in  $\mathbb{Z}_{P_2}^*$ . Let  $g_3$  be an element of order  $P_4$  in  $\mathbb{Z}_{P_3}^*$ . As before, the key transformations are additions (below, we will formally define the set  $T^+$  in this context). We consider the following functions:

- The function  $F'_1(k) = g^k \bmod P$  is weak, since  $g^{k+\delta} = g^k * g^\delta \bmod P$ . Thus, given  $g^k = F'_1(k+0)$  and  $\delta$ , we can compare a response from the RK oracle with  $F'_1(k+0) = g^k * g^\delta \bmod P$ . This is used in [1, 2] for straightforward RK attacks certain Diffie-Hellman based PRFs.
- Similarly, the function  $F'_2(k) = g^{(g_2^k)} \bmod P$  is also weak, since  $g^{(g_2^{k+\delta})} = g^{(g_2^k)(g_2^\delta)} = (g^{(g_2^k)})^{(g_2^\delta)} \bmod P$ .
- The function

$$F'_{\text{DH}}(k) = g^{(g_2^{(g_3^k)})} \bmod P$$

looks like a promising candidate.

For  $F'_{\text{DH}}(k)$ , the set of keys is  $\mathbb{Z}_{P_4}$ . Consequently, our set  $T^+$  of key transformations is defined by the addition modulo  $P_4$ .

**Definition 3.** Let  $P$ ,  $P_4$ ,  $g$ ,  $g_2$ , and  $g_3$  be defined as above. Let  $r$  be a random value in  $\mathbb{Z}_{P_4}^*$ . Define

$$f(x) = g^{(g_2^{(g_3^{x+r})})} \bmod P.$$

Define  $R = \{z \in \mathbb{Z}_P \mid \exists k \in \mathbb{Z}_{P_4} : z = F'_{\text{DH}}(k)\}$ .

**Diffie-Hellman Random Function Assumption (DHRFA):** It is infeasible, to distinguish  $f$  from a random function  $\mathbb{Z}_{P_4} \rightarrow R$ .

**Theorem 6 (Security of  $F'_{\text{DH}}$ ).** Under the DHRFA, there exists no efficient  $T^+$ -transforming adversary for  $F'_{\text{DH}}$  with significant advantage.

The *proof* of Theorem 6 is similar to the proof of Theorem 5 and omitted here.

## 5 Using a Hash Function

As Ross Anderson pointed out at the FSE workshop in Delhi, a common engineering technique to ensure related-key security is to combine a block cipher  $E$  with a hash function  $H$ , defining a new block cipher

$$E_X^H(M) = E_{H(X)}(M).$$

This is a reasonable construction. In fact, for many types of related-key adversaries – including those considered in the current paper – it is straightforward to prove the security of  $E^H$  in the *random oracle model*, assuming  $E$  to be conventionally secure.

This approach has the following drawbacks, however:

- For implementing  $E^H$ , we need to implement two cryptographic primitives  $E$  and  $H$ .
- The security of  $E^H$  depends on the security of  $E$  *and* on the security of  $H$ . If, e.g.,  $E$  is conventionally secure but  $H$  fails to meet its security requirements,  $E^H$  can be insecure.
- A random oracle proof of security for  $E^H$  does reveal the security requirements for  $H$ .

On the other hand, it may be possible to prove the security of  $E^H$  against certain kinds of related-key adversaries in the *standard model*, making some nonstandard assumptions on  $H$ .

## 6 Summary

This paper presented new constructions for related-key secure PRFs.

For one construction, a tight security bound against partially-transforming adversaries has been shown, improving the concrete complexity of previous constructions. The proof assumes some block cipher to be secure in the conventional sense (i.e., without related keys).

Two other constructions are shown secure against more general adversaries, however under certain non-standard number-theoretical assumptions.

## References

1. M. Bellare, T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs RKA-PRFs and applications. E. Biham, editor, *Eurocrypt 2003*, Springer Lecture Notes in Computer Science # 2654, pp. 491–506.

2. M. Bellare, T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs RKA-PRFs and applications. March 18, 2003. Full version of [1]. <http://www.cs.ucsd.edu/users/tkohno/papers/RKA/> (URL checked: Jan. 14, 2004).
3. E. Biham. New types of cryptanalytic attacks using related keys. T. Helleseht, editor, *Eurocrypt 93*, Springer Lecture Notes in Computer Science # 765, pp. 398–409.
4. J. Daemen, V. Rijmen. *AES proposal: Rijndael*.
5. J. Daemen, V. Rijmen. *The design of Rijndael*, Springer-Verlag, 2002.
6. Morris Dworkin. DRAFT Recommendation for block cipher modes of operation: the RMAC authentication mode. *NIST Special Publication 800-38b*. October 18, 2002.
7. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, D. Whiting. Improved cryptanalysis of Rijndael. B. Schneier, editor, *Fast Software Encryption 2000*, Springer Lecture Notes in Computer Science # 1978, pp. 213–230.
8. E. Jaulmes, A. Joux, F. Valette. On the security of randomized CBC-MAC beyond the birthday paradox limit: A new construction. J. Daemen, V. Rijmen, editors, *Fast Software Encryption 2002*, Springer Lecture Notes in Computer Science.
9. J. Kelsey, B. Schneier, D. Wagner. Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES. N. Kobitz, editor, *Crypto '96*, Springer Lecture Notes in Computer Science # 1109, pp. 237–251.
10. J. Kelsey, B. Schneier, D. Wagner. Related-key cryptanalysis of 3-Way, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. Y. Han, T. Okamoto, S. Quing, editors, *Information and Communications Security '97*, Springer Lecture Notes in Computer Science # 1334, pp. 233–246.
11. L. Knudsen. Cryptanalysis of LOKI91. J. Seberry, Y. Zheng, editors, *Auscrypt '92*, Springer Lecture Notes in Computer Science # 718, pp. 196–208.
12. L. Knudsen, T. Kohno. Analysis of RMAC. T. Johansson, editor, *Fast Software Encryption 2003*, Springer Lecture Notes in Computer Science # 2887, pp. 182–191.
13. M. Liskov, R. Rivest, D. Wagner. Tweakable block ciphers. M. Yung, editor, *Crypto '02*, Springer Lecture Notes in Computer Science # 2422, pp. 31–46.
14. M. Naor, O. Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited. *J. of Cryptology*, vol 12, 1999, pp. 29-66.
15. D. Pointcheval. New public key cryptosystems based on the dependent-RSA problems. Jacques Stern, editor, *Eurocrypt 1999*, Springer Lecture Notes in Computer Science # 1592, pp. 239-254.