

Semester Thesis in Cryptography

Adaptive Security of Compositions

Patrick Pletscher*
ETH Zurich

June 30, 2005

Supervised by:
Krzysztof Pietrzak, Prof. Ueli Maurer

*Email: pat@student.ethz.ch

In a recent paper by Pietrzak [Pie05] it was shown that the sequential composition of two non-adaptively secure pseudo-random functions is in general not guaranteed to be secure against adaptive distinguishers. In this paper we will extend this results and construct a non-adaptively secure pseudo-random function, such that the sequential composition of any number of such functions can be distinguished with only two adaptive queries.

Contents

1	Introduction and Preliminaries	4
1.1	Introduction	4
1.2	Notations and definitions	4
1.2.1	(In)distinguishability and pseudo-random functions	4
1.2.2	Sequential composition	5
1.2.3	Parallel composition	5
1.2.4	The Decisional Diffie-Hellman (DDH) assumption	6
1.3	The Pohlig-Hellman exponentiation cipher	6
1.4	Known results	7
1.4.1	Information theoretic results	7
1.4.2	Black-Box composition does not imply adaptive security	7
1.4.3	Counterexamples for compositions of two functions	7
2	Sequential Composition	9
2.1	Definition of the function for the counterexample	9
2.2	Adaptive Distinguishability of the Sequential Composition	9
2.3	Non-Adaptive Indistinguishability of \mathbf{F}	10
2.3.1	At least as hard as Decisional Diffie-Hellman	10
2.3.2	The hybrid argument	12

1 Introduction and Preliminaries

1.1 Introduction

It is a belief shared by many cryptographers that the composition of pseudo-random systems enhances the security. A specific question in this area is, whether the composition of non-adaptively secure pseudo-random functions is secure against adaptive adversaries. Interestingly this is true in the information theoretic setting [MP04], but until recently it was unknown if the same holds as well for computational indistinguishability. [Pie05] shows that this is wrong by giving two counterexamples, one for the sequential and one for the parallel composition, however this counterexamples are restricted to the composition of only two functions. In this paper we extend the counterexample for the sequential composition to arbitrary many functions. The computational assumption we use for the counterexample is closely related to the Decisional Diffie-Hellman assumption, however our assumption might be even weaker.

Sadly we were unable to reuse the ideas gained through the counterexample for the sequential composition as well for the parallel composition. So the generalization for the parallel composition remains an open problem.

1.2 Notations and definitions

1.2.1 (In)distinguishability and pseudo-random functions

Definition (Distinguisher). A *distinguisher* A is an efficient oracle algorithm which at the end of the computation outputs a decision bit. A distinguisher is *non-adaptive* if he generates all his oracle queries before reading any input and *adaptive* if the distinguisher can choose the $i + 1$ 'th query after seeing the output of the i 'th query.

Definition (Distinguisher's Advantage). The advantage of an algorithm A for distinguishing S_0 from S_1 is

$$\mathbf{Adv} = |\Pr[A^{S_0} = 1] - \Pr[A^{S_1} = 1]|$$

Definition (Pseudo-random function). A function $R : \mathcal{K}_n \times \mathcal{X}_n \rightarrow \mathcal{Y}_n$, where n is a security parameter in \mathbb{N} , is pseudo-random if

1. $R(k, x)$ where $k \in \mathcal{K}_n$, $x \in \mathcal{X}_n$ can be computed by a polynomial-time algorithm.
2. For a randomly chosen k , $R_k(\cdot) := R(k, \cdot)$ can not be distinguished from a function $\mathcal{R} : \mathcal{X}_n \rightarrow \mathcal{Y}_n$ which is selected uniformly at random from all functions from \mathcal{X}_n to

\mathcal{Y}_n . Formally, for any $c > 0$ and any efficient A there $\exists n_0$ such that for $\forall n \geq n_0$

$$\max_{\text{efficient } A} \left| \Pr[A^{\mathcal{R}_k(\cdot)} = 1] - \Pr[A^{\mathcal{R}(\cdot)} = 1] \right| \leq \frac{1}{n^c},$$

We denote the distinguishing advantage of any distinguisher A which runs in time t and makes at most q queries for \mathbf{R} from \mathcal{R} by

$$\mathbf{Adv}_{\mathbf{R}}(q, t) := \max_A \left| \Pr[A^{\mathcal{R}_k(\cdot)} = 1] - \Pr[A^{\mathcal{R}(\cdot)} = 1] \right|$$

1.2.2 Sequential composition

We write $S := F_1 \triangleright \dots \triangleright F_n$ to denote the sequential composition of n functions F . An illustration of the sequential composition of n functions F is given in Figure 1.1.

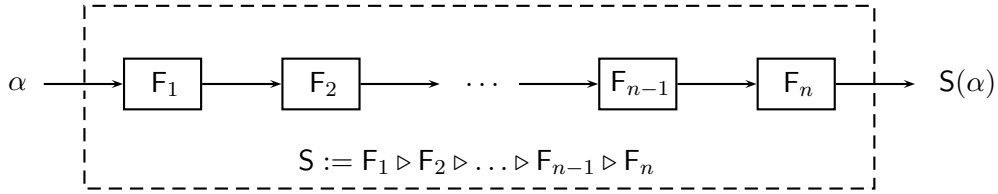


Figure 1.1: Sequential composition of n functions

1.2.3 Parallel composition

We write $P := F_1 \star \dots \star F_n$ to denote the parallel composition of n functions F . Here \star is the group over the range of F . To simplify things we assume that this group is abelian, like this we can ignore questions regarding the order of computations. An illustration of the parallel composition of n functions F is given in Figure 1.2.

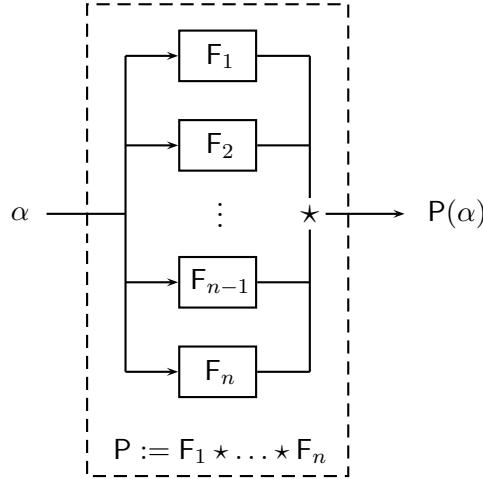


Figure 1.2: Parallel composition of n functions

1.2.4 The Decisional Diffie-Hellman (DDH) assumption

The DDH assumption for a prime order cyclic group \mathcal{G} states that for a generator g of \mathcal{G} and random x, y the triplet (g^x, g^y, g^{xy}) is indistinguishable from random. We denote the maximal advantage of any algorithm A running in time t for the DDH problem as

$$\text{Adv}_{DDH}(t) := \max_A \left| \Pr_{x,y}[A(g^x, g^y, g^{xy}) = 1] - \Pr_{a,b,c}[A(g^a, g^b, g^c) = 1] \right|$$

1.3 The Pohlig-Hellman exponentiation cipher

In our counterexample we will use the Pohlig-Hellman encryption scheme [HP84]. In this section we shortly summarize this rarely used encryption scheme and stress the important property we will use.

First we choose a prime p along with a secret key e , $1 \leq e \leq p-2$, from which a second key d , $1 \leq d \leq p-2$ is computed, such that $ed \equiv 1 \pmod{p-1}$. A message $m \in \mathbb{Z}_p^*$ is encrypted by exponentiating it with the secret key e : $c \equiv m^e \pmod{p}$. The plaintext can then be recovered by exponentiating the ciphertext with the multiplicative inverse of e : $c^d \equiv m^{ed} \equiv m \pmod{p}$.

This encryption scheme has a nice property we are interested in: It is a commutative encryption. This means that if we encrypt a message with several keys we always get the same cipher-text no matter in which order the keys are used. For example if both, Alice and Bob (with secret keys e_A resp. e_B) should encrypt a message m , it is unimportant in which order this encryption happens.

$$(m^{e_A})^{e_B} \equiv_p m^{e_A e_B} \equiv_p (m^{e_B})^{e_A}$$

1.4 Known results

1.4.1 Information theoretic results

Maurer and Pietrzak examine in [MP04] the question whether the composition of random systems enhances the security of its components. They indeed prove that the sequential and parallel composition of two random functions, each secure against non-adaptive distinguishers, is secure against adaptive adversaries. In the following we state the two important theorems of [MP04].

We denote the advantages of the best adaptive distinguisher and the best non-adaptive distinguisher (each computationally unbounded) making k queries by Δ_k and δ_k respectively. \mathbf{R} and \mathbf{P} denote a uniform random function and permutation respectively.

Theorem 1.1. *For random functions \mathbf{E}, \mathbf{F} and \star the parallel composition*

$$\delta_k(\mathbf{E}, \mathbf{R}) \leq \epsilon \quad \wedge \quad \delta_k(\mathbf{F}, \mathbf{R}) \leq \epsilon \quad \implies \quad \Delta_k(\mathbf{F} \star \mathbf{E}, \mathbf{R}) \leq 2\epsilon \left(1 + \ln \frac{1}{\epsilon}\right).$$

Theorem 1.2. *For two random permutations \mathbf{E}, \mathbf{F} and \triangleright the sequential composition*

$$\delta_k(\mathbf{E}, \mathbf{P}) \leq \epsilon \quad \wedge \quad \delta_k(\mathbf{F}, \mathbf{P}) \leq \epsilon \quad \implies \quad \Delta_k(\mathbf{E} \triangleright \mathbf{F}, \mathbf{P}) \leq 2\epsilon \left(1 + \ln \frac{1}{\epsilon}\right).$$

It is important to notice that this result only holds in the information theoretic sense. However, what happens if we only consider computationally bounded adversaries? Can we prove the same as well for computational indistinguishability?

1.4.2 Black-Box composition does not imply adaptive security

In [Mye04] Myers studies the question whether one can prove any adaptive security properties of black-box compositions. He defines an oracle O that implements adaptively secure pseudo-random permutations together with a second oracle R , which is used to weaken the security of O . R is defined in such a way that it only helps adversaries which can query the composition of permutations implemented by O adaptively, but is of no help for non-adaptive adversaries.

Like this Myers shows that there is an oracle relative to which there exist non-adaptively secure permutations such that every composition of polynomially many such permutations can be distinguished adaptively from random. This shows that if composition would imply adaptive security, then no black-box reduction proof for this could exist. He argues that this might be the reason for the lack of formal evidence that composition increases security.

1.4.3 Counterexamples for compositions of two functions

In [Pie05] it is shown that the parallel and sequential composition of non-adaptively secure pseudo-random functions is in general not secure against adaptive distinguishers. The proof is done by giving a counterexample which uses an El-Gamal public-key

cryptosystem whose security is based on the Decisional Diffie-Hellman assumption. The counterexample we will give in chapter 2 is greatly inspired by the counterexamples used in [Pie05], although we will use a Pohlig-Hellman encryption instead of the El-Gamal encryption.

The counterexample presented in our paper has two advantages and one big disadvantage over the ones in [Pie05]: First our counterexample is conceptually simpler, all the involved functions are defined in the same way and secondly it can be used for any number of functions, not only two. The big drawback is however, as already mentioned, that we were only able to use the idea for the sequential composition, but not for the parallel.

2 Sequential Composition

In this chapter we define a pseudo-random function F which is secure against non-adaptive distinguishers, if an assumption at least as weak as the Decisional Diffie-Hellman assumption holds. We will then prove that the sequential composition of any $n \in \mathbb{N}$ such functions F is not secure against adaptive distinguishers. An adaptive adversary is able to distinguish the output of the sequential composition from random with only two adaptive queries.

So we show that the sequential composition of n pseudo-random functions, which are non-adaptively secure, does not imply adaptive security.

2.1 Definition of the function for the counterexample

For the counterexample we use a cyclic group \mathcal{G} , in which the Decisional Diffie-Hellman problem is assumed to be difficult, i.e. we work in a group with order p , where p is prime. g denotes a generator of this group. Furthermore we define $\mathcal{G}_S := \mathcal{G} - \{1\}$.

$F : \mathbb{Z}_p^* \times \mathcal{K}_R \times \mathcal{G}_S^4 \rightarrow \mathcal{G}_S^4$ with key $(x \in \mathbb{Z}_p^*, k \in \mathcal{K}_R)$ on input (s, t, u, v) first computes some pseudo-random values with the help of a pseudo-random function $R : \mathcal{K}_R \times \mathcal{G}_S^4 \rightarrow (\mathbb{Z}_p^*)^2$.

$$(r_1, r_2) \leftarrow R_k(s, t, u, v)$$

Then the output is computed as

$$F(s, t, u, v) \rightarrow (s^{xr_1}, t^{r_1}, u^{x^{-1}r_2}, v^{r_2}) \quad (2.1)$$

where x^{-1} denotes the multiplicative inverse of x in \mathbb{Z}_p^* , i.e. $xx^{-1} \equiv 1 \pmod{p}$.

Remark. We exclude the neutral element of \mathcal{G} in the domain of F , as exponentiating the neutral element with any value in \mathbb{Z}_p^* would always yield to a 1 as output, which would make F distinguishable from random with one query. As the generated randomness and the secret keys are different from 0 modulo p , we won't get the neutral element when exponentiating with those values. Thus the output excludes as well the neutral element.

2.2 Adaptive Distinguishability of the Sequential Composition

We now consider the cascade $F_1 \triangleright \dots \triangleright F_n$. First we need to introduce some abbreviations we will use in this section. With (x_i, k_i) we denote the key of the i 'th function F_i in the sequential composition. $r_{F_i}^{(j,l)}$ is used to denote the l 'th output of the pseudo-random

function R_{k_i} in the j 'th query. Furthermore $r_S^{(j,l)}$ is used to denote the product,

$$r_S^{(j,l)} := r_{F_1}^{(j,l)} \cdot \dots \cdot r_{F_n}^{(j,l)}.$$

We now show that we are able to distinguish the sequential composition of n functions F with only two queries from random. We use (g, g, g, g) as the first input to S , for any $g \in \mathcal{G}_S$. We will then get an output of the following form

$$(g^{x_1 \dots x_n \cdot r_S^{(1,1)}}, g^{r_S^{(1,1)}}, g^{x_1^{-1} \dots x_n^{-1} \cdot r_S^{(1,2)}}, g^{r_S^{(1,2)}}),$$

If we then interchange the first two output arguments by the third and fourth argument we get the quadruple

$$(g^{x_1^{-1} \dots x_n^{-1} \cdot r_S^{(1,2)}}, g^{r_S^{(1,2)}}, g^{x_1 \dots x_n \cdot r_S^{(1,1)}}, g^{r_S^{(1,1)}}).$$

By using this as our next query, all the x_1, \dots, x_n resp. $x_1^{-1}, \dots, x_n^{-1}$ in the exponent will cancel out. So after the second query we get the output

$$(g^{r_S^{(1,2)} r_S^{(2,1)}}, g^{r_S^{(1,2)} r_S^{(2,1)}}, g^{r_S^{(1,1)} r_S^{(2,2)}}, g^{r_S^{(1,1)} r_S^{(2,2)}}).$$

We are of course able to distinguish this quadruple from random, as e.g. the second argument is always the same as the first. We succeed with a very high probability, as a random function fulfills this test only with a very small probability.

2.3 Non-Adaptive Indistinguishability of F

We will prove that

$$\mathbf{Adv}_F^{\text{non-adaptive}}(q, t) \leq \mathbf{Adv}_R(q, t') + q \mathbf{Adv}_{DDH}(t') \quad (2.2)$$

Where $t' = t + \text{poly}(\log p, q)$ for some polynomial poly which accounts for the overhead implied by the reduction we make. Below we will treat R_{k_F} as if it was a truly random function, the $\mathbf{Adv}_R(q, t')$ term in (2.2) does account for this inaccuracy.

2.3.1 At least as hard as Decisional Diffie-Hellman

We will first prove that distinguishing one output of the function F from random, is at least as difficult as the Decisional Diffie-Hellman problem. We assume that the adversary can use an oracle before querying, which tells him the discrete logarithms of his inputs. So if he inputs (s, t, u, v) to the oracle, he gets (z_1, z_2, z_3, z_4) , which satisfy $(g^{z_1}, g^{z_2}, g^{z_3}, g^{z_4}) = (s, t, u, v)$. This additional information can only help the adversary, as he can always ignore it.

On input (s, t, u, v) the adversary gets the output of the function F :

$$(s^{xr_1}, t^{r_1}, u^{x^{-1}r_2}, v^{r_2}),$$

which is equivalent to

$$(g^{z_1 x r_1}, g^{z_2 r_1}, g^{z_3 x^{-1} r_2}, g^{z_4 r_2}). \quad (2.3)$$

As the mapping ϕ , defined by

$$\phi : (s, t, u, v) \rightarrow (s^{z_1^{-1}}, t^{z_2^{-1}}, u^{z_3^{-1}}, v^{z_4^{-1}})$$

and its inverse

$$\phi^{-1} : (s, t, u, v) \rightarrow (s^{z_1}, t^{z_2}, u^{z_3}, v^{z_4}),$$

are efficiently computable for the adversary, we can focus on the problem of distinguishing the output of applying ϕ to (2.3), namely

$$(g^{x r_1}, g^{r_1}, g^{x^{-1} r_2}, g^{r_2})$$

for random x, r_1, r_2 from random output. We then notice that the problem where we have $(x r_1, r_1, x^{-1} r_2, r_2)$ in the exponents is equivalent to the problem where we have $(a, b, c, a c b^{-1})$ in the exponent. So the adversary's task is now to distinguish

$$(g^a, g^b, g^c, g^d) \quad (2.4)$$

which satisfy $d = a c b^{-1}$ from the case where a, b, c, d are random and independent.

To shortly repeat what we have done: we have reformulated our original problem to distinguish the output of (2.1) from random to the problem of distinguishing (2.4) where $d = a c b^{-1}$ from the case where d is random and independent. We will now show that solving (2.4) is at least as hard as the Decisional Diffie-Hellman problem.

We now assume that there exists a distinguisher D which can distinguish

$$(g^a, g^b, g^c, g^{a c b^{-1}}),$$

for truly random a, b, c , from

$$(g^a, g^b, g^c, g^d)$$

for random a, b, c, d with advantage ϵ . But with the help of the distinguisher D , we could as well solve the Decisional Diffie-Hellman problem by creating a distinguisher D' which uses D and has the same advantage ϵ .

On input $(u, v, w) = (g^\alpha, g^\beta, g^\gamma)$ D' would create a random value r and compute its inverse r^{-1} . D' would then use the distinguisher D with the input

$$(u, g^r, v, w^{r^{-1}}) = (g^\alpha, g^r, g^\beta, g^{\gamma r^{-1}})$$

to decide the Decisional Diffie-Hellman problem. If γ satisfies $\gamma = \alpha\beta$ then $\gamma r^{-1} = \alpha\beta r^{-1}$ is as well satisfied. If the exponent γ is random, then $(g^\alpha, g^r, g^\beta, g^{\gamma r^{-1}})$ is a random quadruple.

2.3.2 The hybrid argument

So we have now proven that distinguishing one output of F from random is at least as difficult, as solving the Decisional Diffie-Hellman problem. We now use the hybrid argument to give an upper bound for the non-adaptive distinguishing probability for q queries.

We assume that there exists an algorithm A , which can distinguish

$$(g^{xr^{(1,1)}}, g^{r^{(1,1)}}, g^{x^{-1}r^{(1,2)}}, g^{r^{(1,2)}}), \dots, (g^{xr^{(q,1)}}, g^{r^{(q,1)}}, g^{x^{-1}r^{(q,2)}}, g^{r^{(q,2)}}) \quad (2.5)$$

for random $x, r^{(i,j)}$, from

$$(g^{a_1}, g^{b_1}, g^{c_1}, g^{d_1}), \dots, (g^{a_q}, g^{b_q}, g^{c_q}, g^{d_q}) \quad (2.6)$$

for random a_i, b_i, c_i, d_i . If A can distinguish (2.5) from (2.6) with probability ϵ we could construct an algorithm A' , which is able to distinguish $(g^{xr_1}, g^{r_1}, g^{x^{-1}r_2}, g^{r_2})$ from (g^a, g^b, g^c, g^d) for random x, r_1, r_2, a, b, c and d with advantage ϵ/q using the hybrid argument. The distribution of the i 'th hybrid is

$$(g^{xr^{(1,1)}}, g^{r^{(1,1)}}, g^{x^{-1}r^{(1,2)}}, g^{r^{(1,2)}}), \dots, (g^{xr^{(i,1)}}, g^{r^{(i,1)}}, g^{x^{-1}r^{(i,2)}}, g^{r^{(i,2)}}), \\ (g^{a_{i+1}}, g^{b_{i+1}}, g^{c_{i+1}}, g^{d_{i+1}}), \dots, (g^{a_q}, g^{b_q}, g^{c_q}, g^{d_q})$$

for random $x, r^{(1,1)}, \dots, r^{(i,1)}, r^{(1,2)}, \dots, r^{(i,2)}, a_{i+1}, \dots, a_q, b_{i+1}, \dots, b_q, c_{i+1}, \dots, c_q$ and d_{i+1}, \dots, d_q . Our A' on input $(\alpha, \beta, \gamma, \delta)$ chooses a random i , $1 \leq i \leq q$ and generates the distribution

$$(g^{xr^{(1,1)}}, g^{r^{(1,1)}}, g^{x^{-1}r^{(1,2)}}, g^{r^{(1,2)}}), \dots, (g^{xr^{(i-1,1)}}, g^{r^{(i-1,1)}}, g^{x^{-1}r^{(i-1,2)}}, g^{r^{(i-1,2)}}), \\ (\alpha, \beta, \gamma, \delta), (g^{a_{i+1}}, g^{b_{i+1}}, g^{c_{i+1}}, g^{d_{i+1}}), \dots, (g^{a_q}, g^{b_q}, g^{c_q}, g^{d_q})$$

whose distribution is equal to the $i - 1$ 'th hybrid if $(\alpha, \beta, \gamma, \delta)$ are three random values, and equal to the i 'th hybrid if $(\alpha, \beta, \gamma, \delta)$ was generated as $(g^{xr_1}, g^{r_1}, g^{x^{-1}r_2}, g^{r_2})$.

Acknowledgments

I would like to thank Krzysztof Pietrzak for his enormous support while writing this semester thesis, especially his ideas concerning both the definition of the counterexample and the reduction to the DDH problem, proved to be very important. He as well contributed many corrections which helped a lot to improve the earlier versions of this thesis. I as well would like to thank Prof. Maurer for introducing me to the fascinating topic of cryptography and in a broader sense for teaching formal concepts in way, such that they are a great joy to learn.

Bibliography

- [HP84] Martin Hellman and Steve Pohlig. Exponentiation cryptographic apparatus and method, 1984. US Patent 4,424,414 (Expired).
- [MP04] Ueli Maurer and Krzysztof Pietrzak. Composition of random systems: When two weak make one strong. In *Theory of Cryptography — TCC '04*, volume 2951 of *Lecture Notes in Computer Science*, pages 410–427, 2004.
- [Mye04] Steven Myers. Black-box composition does not imply adaptive security. In *Advances in Cryptology — EUROCRYPT 04*, volume 3027 of *Lecture Notes in Computer Science*, pages 189–206, 2004.
- [Pie05] Krzysztof Pietrzak. Composition does not imply adaptive security. In *Advances in Cryptology — CRYPTO '05 (to appear)*, Lecture Notes in Computer Science, 2005.