Composition Does Not Imply Adaptive Security

Krzysztof Pietrzak[⋆]

ETH Zürich, Department of Computer Science, CH-8092 Zürich Switzerland
pietrzak@inf.ethz.ch

Abstract. We study the question whether the sequential or parallel composition of two functions, each indistinguishable from a random function by non-adaptive distinguishers is secure against adaptive distinguishers. The sequential composition of $\mathsf{F}(.)$ and $\mathsf{G}(.)$ is the function $\mathsf{G}(\mathsf{F}(.))$, the parallel composition is $\mathsf{F}(.) \star \mathsf{G}(.)$ where $\star$ is some group operation. It has been shown that composition indeed gives adaptive security in the information theoretic setting, but unfortunately the proof does not translate into the more interesting computational case.
In this work we show that in the computational setting composition does not imply adaptive security: If there is a prime order cyclic group where the decisional Diffie-Hellman assumption holds, then there are functions F and G which are indistinguishable by non-adaptive polynomially time-bounded adversaries, but whose parallel composition can be completely broken (i.e. we recover the key) with only three adaptive queries. We give a similar result for sequential composition. Interestingly, we need a standard assumption from the asymmetric (aka. public-key) world to prove a negative result for symmetric (aka. private-key) systems.

1 Sequential and Parallel Composition

We continue to investigate the question whether composition of (pseudo) random functions yields a function whose security is in some sense superior to any of it's components. The two most natural ways to compose functions is to either apply them sequentially or in parallel. For two function F and G we denote by $\mathsf{G} \circ \mathsf{F}$ the *sequential composition*: $\mathsf{G} \circ \mathsf{F}(x) \stackrel{\text{def}}{=} \mathsf{G}(\mathsf{F}(x))$. And by $\mathsf{F} \star \mathsf{G}$ the *parallel composition*: $\mathsf{F} \star \mathsf{G}(x) \stackrel{\text{def}}{=} \mathsf{F}(x) \star \mathsf{G}(x)$ where $\star$ is some group operation defined on the range of F and G.

In the information theoretic model one considers computationally unbounded adversaries and only bounds the number of queries they are allowed to make. In this model Vaudenay [9] shows that if a permutation F cannot be distinguished from random with advantage more than ϵ by any adaptive (resp. non-adaptive)[1] distinguisher making q queries, then the sequential composition of k such permutations has security $2^{k-1}\epsilon^k$ against adaptive (resp. non-adaptive) distinguishers.

[⋆] Supported by the Swiss National Science Foundation, project No. 200020-103847/1.

[1] Adaptive means that the distinguisher can choose the $(i+1)$'th query after seeing the output to the i'th query. A non-adaptive distinguisher must decide which q queries he wants to make beforehand.

The same holds for parallel composition where F can be a function and doesn't have to be a permutation. For the computational case, where one considers polynomially time-bounded adversaries a similar amplification result was proven by Luby and Rackoff [3].[2] So if we have a function with some security against adaptive (resp. non-adaptive) distinguishers we can amplify this security for *the same* class of distinguishers in both models.

Another question is whether we always get adaptive security by the composition of non-adaptively secure functions. This is in fact true in the information theoretic model: Maurer and Pietrzak [4] show that if F and G both have security ϵ against *non-adaptive* distinguishers, then $\mathsf{F} \star \mathsf{G}$ has security $2\epsilon(1 + \ln \epsilon^{-1})$ against *adaptive* distinguishers (the same holds for $\mathsf{G} \circ \mathsf{F}$ if F and G are permutations). But no such result is known for the computational case. In fact, Myers [6] showed that there is an oracle relative to which non-adaptively secure permutations exist, but their sequential composition is not adaptively secure. This means that if it was indeed true that composition would always imply adaptive security, no relativizing proof for that does exist. As only very few non-relativizing proofs are known (not only in cryptography, but in complexity theory in general), Myers argues that this might be the reason for the lack of formal evidence that composition increases security even though this belief is shared by many cryptographers (including myself until recently).

Here we show that composition does not imply adaptive security in general if there is a group where the decisional Diffie-Hellman assumption holds. We will construct functions F and G which are indistinguishable by non-adaptive (polynomial time) distinguishers if the DDH assumption holds. But where a simple adaptive strategy exists to get the whole key out of $\mathsf{F} \star \mathsf{G}$ with only three adaptive queries. We then construct F and G such that the same holds for $\mathsf{G} \circ \mathsf{F}$.

1.1 Notation and Definitions

EFFICIENT/NEGLIGIBLE/INDISTINGUISHABLE. We denote by $\kappa \in \mathbb{N}$ our security parameter. An efficient algorithm is an algorithm whose running time is polynomial in κ. A function $\mu : \mathbb{N} \rightarrow [0, 1]$ is negligible if for any $c > 0$ there is an n_0 such that $\mu(n) \leq 1/n^c$ for all $n \geq n_0$. Two families of distributions (indexed with κ) are indistinguishable if any efficient algorithm has negligible advantage (over a random guess) in distinguishing those distributions.

THE DDH ASSUMPTION. The DDH assumption for a prime order cyclic group $\mathcal{G} = \mathcal{G}(\kappa)$ 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

[2] Unlike in Vaudenay's information theoretic result, where k, the number of components in the cascade, can be arbitrary (in particular any function of n), the computational amplification proven in [3] requires k to be a constant and independent of the security parameter. Myers [5] proves a stronger amplification for PRFs (which unlike [3] allows to turn a weak PRF into a strong one) for a construction which is basically parallel composition with some extra random values XOR-ed to the inputs.

algorithm A running in time t for the DDH problem as

$$\mathbf{Adv}_{DDH}(t) \stackrel{\text{def}}{=} \max_A |\mathsf{Pr}_{x,y}[A(g^x, g^y, g^{xy}) \to 1] - \mathsf{Pr}_{a,b,c}[A(g^a, g^b, g^c) \to 1]|$$

For example the DDH assumption is believed to be true for the following groups: Let Q be a prime such that $Q-1 = rP$ for some large prime P (say $\log(P) \geq \kappa$). Let h be a generator of $\mathbb{Z}_Q^*$, then $g = h^r$ is a generator of the subgroup $\mathcal{G} \stackrel{\text{def}}{=} \langle g \rangle$ of order P. In $\mathcal{G}$ any $a \neq 1$ is a generator, here 1 denotes the identity element.
The El-Gamal Cryptosystem. Let $\mathcal{G}, g, P$ be like above. The El-Gamal public-key cryptosystem [2] over $\mathcal{G}$ with generator g is defined as follows: The private-key is a random $x \in \mathbb{Z}_P$, and the public key is g^x. To encrypt $m \in \mathcal{G}$ with the public key g^x we choose $r \in \mathbb{Z}_P$ at random and compute the ciphertext in $\mathcal{G}^2$ as

$$\mathsf{Enc}_{g^x}(m, r) = (mg^{xr}, g^r)$$

The decryption of a ciphertext (a, b) with secret key x goes as

$$\mathsf{Dec}_x(a, b) = a/b^x$$

This scheme has some nice properties we will use. It is multiplicative homomorphic: Given an encryption $(mg^{xr}, g^r) = \mathsf{Enc}_{g^x}(m, r)$ of m we can compute an encryption of ℓm as $(\ell m g^{xr}, g^r) = \mathsf{Enc}_{g^x}(\ell m, r)$ even without knowing m or even the public key g^x. In particular given an encryption $(mg^{xr}, g^r) = \mathsf{Enc}_{g^x}(m, r)$ of a known message m we can compute $(g^{xr}, g^r) = \mathsf{Enc}_{g^x}(1, r) = \mathsf{Enc}_{g^x}(1, r)$, an encryption of 1, without even knowing the public key g^x. And further we can rerandomise this encryption by exponentiating with some random r' as $(g^{xrr'}, g^{rr'}) = \mathsf{Enc}_{g^x}(1, rr')$.[3]
Distinguisher. By *distinguisher* we denote 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 queries before reading any inputs.

A function $\mathsf{R} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is pseudorandom if for a random key $k \in \mathcal{K}$ the function $\mathsf{R}_k(.) \stackrel{\text{def}}{=} \mathsf{R}(k, .)$ is indistinguishable from a random function $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$. We denote the distinguishing advantage for R form $\mathcal{R}$ of any distinguisher which runs in time t and makes at most q queries by

$$\mathbf{Adv}_{\mathsf{R}}(q, t) \stackrel{\text{def}}{=} \max_A |\mathsf{Pr}_k[A^{\mathsf{R}_k(.)} \to 1] - \mathsf{Pr}[A^{\mathcal{R}(.)} \to 1]|$$

We write $\mathbf{Adv}_{\mathsf{R}}^{non-adaptive}(q, t)$ if the maximum is only taken over all non-adaptive distinguishers.

[3] This is not the standard way of randomising El-Gamal encryptions, where one multiplies (mg^{xr}, g^r) with $(g^{xr'}, g^{r'})$ for a random r' to get $(mg^{x(r+r')}, g^{r+r'}) = \mathsf{Enc}_{g^x}(m, r + r')$. This randomisation works for any encrypted message (not just for $m = 1$), but it requires knowledge of the public key and because of that is not useful for our purpose.

1.2 Some Intuition

For the counterexamples to the conjecture that parallel (resp. sequential) composition does imply adaptive security we will define functions F and G whose output looks random with high probability for any fixed sequence of queries. But if we can query $\mathsf{F} \star \mathsf{G}$ (reps. $\mathsf{G} \circ \mathsf{F}$) adaptively we can somehow "convince" F and G of the fact that they are queried adaptively. We then define F, G such that they output their key when they are convinced. We achieve this by letting F and G "communicate" using a semantically secure public-key cryptosystem. The El-Gamal cryptosystem has all the additional features we will need.

The Parallel Composition Counterexample. We will now sketch our counterexample of two non-adaptively secure functions F, G where $\mathsf{F} \star \mathsf{G}$ can be broken with three adaptive queries. The full proof is given in Section 2. Let R be any adaptively secure pseudorandom function. The keyspace of F is a (El-Gamal) public/secret-key pair $(pk_\mathsf{F}, sk_\mathsf{F})$ and a key k_F for R (G's key is $(pk_\mathsf{G}, sk_\mathsf{G}), k_\mathsf{G}$). The first thing F/G do on any input is to run it through $\mathsf{R}_{k_\mathsf{F}}/\mathsf{R}_{k_\mathsf{G}}$ to produce some pseudorandomness.

We define F and G such that on one particular input α they output their public keys. So if we query $\mathsf{F} \star \mathsf{G}$ with α we get $pk_\mathsf{F} pk_\mathsf{G}$.

$$\alpha \to \left\{ \begin{matrix} \overset{\mathsf{F}}{\to} pk_\mathsf{F} \\ \overset{\mathsf{G}}{\to} pk_\mathsf{G} \end{matrix} \right\} \to pk_\mathsf{F} pk_\mathsf{G}$$

We further define F and G such that for some fixed β on all inputs of the form (u, β) F computes $pk = u/pk_\mathsf{F}$ and then outputs the encryption (using the randomness generated by R) of some fixed value γ under pk. G does the same thing. So if we now feed the output from the first query back into $\mathsf{F} \star \mathsf{G}$ we get

$$(pk_\mathsf{F} pk_\mathsf{G}, \beta) \to \left\{ \begin{matrix} \overset{\mathsf{F}}{\to} \mathsf{Enc}_{pk_\mathsf{G}}(\gamma, r) \\ \overset{\mathsf{G}}{\to} \mathsf{Enc}_{pk_\mathsf{F}}(\gamma, r') \end{matrix} \right\} \to \mathsf{Enc}_{pk_\mathsf{G}}(\gamma, r)\mathsf{Enc}_{pk_\mathsf{F}}(\gamma, r')$$

And finally on general input (u, v) we define F as follows: First F divides v by the output it would have produced on input (u, β). If this value is an encryption of γ under pk_F, F is "convinced" that it is in an adaptive setting and outputs his key, otherwise F just outputs some pseudorandom stuff. G does the same thing. Let's see what happens if our third query consists of the outputs from the two first queries we made, i.e. $(pk_\mathsf{F} pk_\mathsf{G}, \mathsf{Enc}_{pk_\mathsf{G}}(\gamma, r)\mathsf{Enc}_{pk_\mathsf{F}}(\gamma, r'))$. Here F checks if the value computed as

$$\frac{\mathsf{Enc}_{pk_\mathsf{G}}(\gamma, r)\mathsf{Enc}_{pk_\mathsf{F}}(\gamma, r')}{\mathsf{Enc}_{pk_\mathsf{G}}(\gamma, r) \leftarrow \mathsf{F}(pk_\mathsf{F} pk_\mathsf{G}, \beta)} = \mathsf{Enc}_{pk_\mathsf{F}}(\gamma, r')$$

is an encryption of γ, as is the case here F outputs its key, and so will G. To prove the non-adaptive security of F and G, we first observe that for a fixed input the above check will fail almost certainly. So we must only care about the α query, which gives the random pk_F and queries of the form (u, β) where

we get $\mathsf{Enc}_{u/pk_{\mathsf{F}}}(\gamma, r)$. We show that if (given pk_{F}) we could distinguish such $\mathsf{Enc}_{u/pk_{\mathsf{F}}}(\gamma, r)$ from random, then DDH cannot be hard in $\mathcal{G}$.
The Sequential Composition Counterexample. We will now sketch our counterexample of two non-adaptively secure functions F, G where $\mathsf{G} \circ \mathsf{F}$ can be broken with three adaptive queries. The full proof is given in Section 3. As in the previous section, let R be an adaptively secure pseudorandom function. Again, F's key is $(pk_{\mathsf{F}}, sk_{\mathsf{F}}), k_{\mathsf{F}}$ and G's key is $(pk_{\mathsf{G}}, sk_{\mathsf{G}}), k_{\mathsf{G}}$.

We define F such that it outputs his public key on some special input α. G first checks if the input is an encryption of 1 (using sk_{G}): if this is the case G is "convinced" and outputs his key. Otherwise the output is simply an encryption of the input. If the first query we make to $\mathsf{G} \circ \mathsf{F}$ is α then

$$\alpha \xrightarrow{\mathsf{F}} pk_{\mathsf{F}} \xrightarrow{\mathsf{G}} \mathsf{Enc}_{pk_{\mathsf{G}}}(pk_{\mathsf{F}}, r)$$

except in the unlikely case where by chance pk_{F} happens to be an encryption of 1 under sk_{G}.

F on inputs $u \neq \alpha$ "treats" u as if it was $\mathsf{Enc}_{pk}(pk_{\mathsf{F}}, r)$, i.e. an encryption of his public key pk_{F} under some key pk. Now (as described earlier in this section) F computes $\mathsf{Enc}_{pk}(1, rr')$, an encryption of 1 with some fresh randomness rr'. If we now feed back the output of the first query into $\mathsf{G} \circ \mathsf{F}$

$$\mathsf{Enc}_{pk_{\mathsf{G}}}(pk_{\mathsf{F}}, r) \xrightarrow{\mathsf{F}} \mathsf{Enc}_{pk_{\mathsf{G}}}(1, rr') \xrightarrow{\mathsf{G}} \mathsf{G}\text{'s key } (sk_{\mathsf{G}}, k_{\mathsf{G}})$$

and we get G's key. With the third query, which we will not sketch here we then can get F's key as well. Again the non-adaptive indistinguishability of F and G can be shown under the DDH assumption.

2 Parallel Composition does Not Imply Adaptive Security

In this section we prove that there are two functions F and G, both $\mathcal{K} \times \mathcal{G}^3 \to \mathcal{G}^3$ ($\mathcal{K}$ denotes the keyspace) which are indistinguishable from a random function $\mathcal{G}^3 \to \mathcal{G}^3$ by any *non-adaptive* distinguisher if the DDH-assumption is true in $\mathcal{G}$. But the parallel composition $\mathsf{F} \star \mathsf{G}$ can be completely broken (i.e. we recover the keys of F and G) with only 3 *adaptive* queries.

The systems F and G are almost identically defined, we first define F and then make a small change to get G. Let $\mathsf{R} : \mathcal{K}_{\mathsf{R}} \times \mathcal{G}^3 \to \mathbb{Z}_P^3$ be any pseudorandom function with keyspace $\mathcal{K}_{\mathsf{R}}$. The keyspace of F and G is $\mathcal{K}_{\mathsf{R}} \times \mathbb{Z}_P$.

There is one annoying technicality we must consider; Because we do not only want to distinguish $\mathsf{F} \star \mathsf{G}$ from a random function in the adaptive case, but recover the keys of F and G, we must somehow encode the keys $(\mathcal{K}_{\mathsf{R}} \times \mathbb{Z}_P)^2$ into the range $\mathcal{G}^3$ of $\mathsf{F} \star \mathsf{G}$. For simplicity we will simply assume that this is possible, i.e. there are two mappings $\phi_1, \phi_2 : \mathcal{K}_{\mathsf{R}} \times \mathbb{Z}_P \to \mathcal{G}^3$ such that from $\phi_1(k_1, x_1)\phi_2(k_2, x_2)$ we can recover k_1, k_2, x_1, x_2.[4]

[4] One could also easily solve this problem without this assumption by simply extending the range of F, G and R with a term $\{0,1\}^{2\ell}$ for an ℓ such that $\mathbb{Z}_P \times \mathcal{K}_{\mathsf{R}}$ can be encoded

F with key $(x \in \mathbb{Z}_P, k_\mathsf{F} \in \mathcal{K}_\mathsf{R})$ on input (u,v,w) first computes some pseudorandom values.

$$(r_1, r_2, r_3) \leftarrow \mathsf{R}_{k_\mathsf{F}}(u,v,w) \tag{1}$$

Now the output is computed as (we set the values α, β and γ as described in Section 1.2 to $\alpha = (1,1,1), \beta = (1,1)$ and $\gamma = 1$)

$$\begin{aligned}
\mathsf{F}(1,1,1) &\rightarrow (g^x, g^{r_2}, g^{r_3}) \\
\mathsf{F}(u \neq 1, 1, 1) &\rightarrow ((u/g^x)^{r_1}, g^{r_1}, g^{r_3}) \\
\mathsf{F}(u \neq 1, v \neq 1, w \neq 1) &\rightarrow (a,b,c) \quad \text{where} \\
&\quad (d,e,f) \leftarrow \mathsf{F}(u,1,1) && (2)\\
&\quad \text{if } (v/d) = (w/e)^x \text{ then} && (3)\\
&\quad\quad (a,b,c) = \phi_1(k_\mathsf{F}, x) && (4)\\
&\quad \text{otherwise } (a,b,c) = (g^{r_1}, g^{r_2}, g^{r_3}) \\
\mathsf{F}(\text{ all other cases }) &\rightarrow (g^{r_1}, g^{r_2}, g^{r_3})
\end{aligned}$$

G with key (y, k_G) is defined similarly, but with (x, k_F) replaced with (y, k_G) and (4) replaced with

$$(a,b,c) = \phi_2(k_\mathsf{G}, y)$$

2.1 Breaking $\mathsf{F} \star \mathsf{G}$ with 3 Adaptive Queries

We will now describe how to get the key out of $\mathsf{F}\star\mathsf{G}$ with 3 adaptive queries. The attack below is successful with probability almost 1. It only fails if by chance P divides one of the random values which appear in the exponent of g below. Below we denote with $r_{(i,j)}$ the pseudorandom value r_i computed by F in step (1) on the j'th input. We define $s_{(i,j)} \overset{\text{def}}{=} g^{r_{(i,j)}}$. Similarly the r', s' are defined for G. We will use s and s' for uninteresting terms whose only raison d'être is to pad the output to the right length. The first query we make is $(1,1,1)$

$$(1,1,1) \rightarrow \left\{ \begin{array}{l} \xrightarrow{\mathsf{F}} (g^x, s_{(2,1)}, s_{(3,1)}) \\ \xrightarrow{\mathsf{G}} (g^y, s'_{(2,1)}, s'_{(3,1)}) \end{array} \right\} \rightarrow \{g^{x+y}, s_{(2,1)}s'_{(2,1)}, s_{(3,1)}s'_{(3,1)})$$

Four our second query we use the first value from the above output.

$$\begin{gathered}(g^{x+y}, 1, 1) \rightarrow \\ \left\{ \begin{array}{l} \xrightarrow{\mathsf{F}} (g^{yr_{(1,2)}}, g^{r_{(1,2)}}, s_{(3,2)}) \\ \xrightarrow{\mathsf{G}} (g^{xr'_{(1,2)}}, g^{r'_{(1,2)}}, s'_{(3,2)}) \end{array} \right\} \rightarrow (g^{yr_{(1,2)}+xr'_{(1,2)}}, g^{r_{(1,2)}+r'_{(1,2)}}, s_{(3,2)}s'_{(3,2)})\end{gathered}$$

with ℓ bits (the group operation on this term is bitwise XOR). If F or G must output their key, they encode it into this term (F into the first, and G into the second half). In all other cases this term is simply filled with a pseudorandom value generated by R.

Our last query is a combination of the two outputs we have seen.

$$(g^{x+y}, g^{yr_{(1,2)}+xr'_{(1,2)}}, g^{r_{(1,2)}+r'_{(1,2)}})) \rightarrow \left\{ \begin{array}{l} \xrightarrow{\mathsf{F}} \phi_1(k_{\mathsf{F}}, x) \\ \xrightarrow{\mathsf{G}} \phi_2(k_{\mathsf{G}}, y) \end{array} \right\} \rightarrow \phi_1(k_{\mathsf{F}}, x)\phi_2(k_{\mathsf{G}}, y)$$

Thus we learn the whole key! Let's see what happened in the last query. F on this input first by (2) simulated itself on the input $(g^{x+y}, 1, 1)$, which was exactly the input in the second query.

$$(g^{yr_{(1,2)}}, g^{r_{(1,2)}}, s_{(3,2)}) \leftarrow \mathsf{F}(g^{x+y}, 1, 1)$$

Next by (3) F checked whether

$$g^{yr_{(1,2)}+xr'_{(1,2)}}/g^{yr_{(1,2)}} = (g^{r_{(1,2)}+r'_{(1,2)}}/g^{r_{(1,2)}})^x$$

and as this is true, F proceeds with (4) and outputs its key $\phi_1(k_{\mathsf{F}}, x)$. Similarly G outputs its key $\phi_2(k_{\mathsf{G}}, y)$.

2.2 Non-Adaptive Indistinguishability of F and G

We will prove that

$$\mathbf{Adv}_{\mathsf{F}}^{non-adaptive}(q, t) \leq \mathbf{Adv}_{\mathsf{R}}(q, t') + \frac{2q}{P} + q\mathbf{Adv}_{DDH}(t') \qquad (5)$$

Where $t' = t + poly(\log P, q)$ for some polynomial $poly$ which accounts for the overhead implied by the reduction we make. The same bound holds for G. Below we will treat $\mathsf{R}_{k_{\mathsf{F}}}$ as if it was a truly random function, the $\mathbf{Adv}_{\mathsf{R}}(q, t')$ term in (5) does account for this inaccuracy.

Assume that the non-adaptive distinguisher A chooses to make q queries (u_i, v_i, w_i) for $i = 1, \ldots, q$. We must only consider inputs of the form $(u, 1, 1)$ and $(u \neq 1, v \neq 1, w \neq 1)$ as in all other cases the output is simply computed by $\mathsf{R}_{k_{\mathsf{F}}}$ and thus is random.

If we make a $(u \neq 1, v \neq 1, w \neq 1)$ query the output is also computed by R_{k_F}, except when $(v/d) = (w/e)^x$ for random d, e, x (here and below we say an element is random if its distribution is uniform over his domain. So here e, f are uniform over $\mathcal{G}$ and x over $\mathbb{Z}_P$). Now

$$\mathsf{Pr}_{d,e,x}[(v/d) = (w/e)^x] \leq 2P^{-1}$$

holds. To see this first note that we have $\mathsf{Pr}[w/e = 1] = \mathsf{Pr}[e = w] = P^{-1}$. Now as in $\mathcal{G}$ any element except 1 is a generator, conditioned on $w \neq e$ the $(w/e)^x$ is random and thus equal to v/d with probability P^{-1}.

So probability that for any of the $t \leq q$ queries of the form $(u \neq 1, v \neq 1, w \neq 1)$ will satisfy $(v/e) = (w/f)^x$ is at most $2tP^{-1}$, the $2qP^{-1}$ term in (5) does account for this probability.

Now we must only consider the case where all q queries of the from $(u_i, 1, 1)$. We make a deal with A: he will only make queries where $u_i \neq 1$ for all $1 \leq i \leq q$

but for this we allow A a $(q+1)$'th query which must be $(1,1,1)$, clearly this can only help A.

Moreover we assume that A knows the discrete logarithm z_i of all his u_i's (i.e. $g^{z_i} = u_i$). Of course this can not be guaranteed, but not knowing them can only decrease A's advantage in the analysis below. So the output A gets on his query $(g^{z_1},1,1),\ldots,(g^{z_q},1,1),(1,1,1)$ is

$$(g^{(z_1-x)r_1}, g^{r_1}, *), \ldots, (g^{(z_q-x)r_q}, g^{r_q}, *), (g^x, *, *) \tag{6}$$

where the $*$'s denote random values which are independent of all the other terms. Now distinguishing (6) from random is equivalent to distinguishing

$$(g^x, g^{r_1 x}, g^{r_1}), \ldots, (g^x, g^{r_q x}, g^{r_q}) \tag{7}$$

for random $x, r_1, \ldots, r_q$ from

$$(a, b_1, c_1), \ldots, (a, b_q, c_q) \tag{8}$$

where $a, b_1, \ldots, b_q, c_1, \ldots, c_q$ are random. To see this consider the (randomised) mapping τ (here $*$ are random values)

$$\tau[(\alpha,\beta_1,\gamma_1),\ldots,(\alpha,\beta_q,\gamma_q)] \rightarrow [(\gamma_1^{z_1}\beta_1^{-1},\gamma_1,*),\ldots,(\gamma_q^{z_q}\beta_q^{-1},\gamma_q,*),(\alpha,*,*)]$$

We get the distribution (6) if we apply τ to (7) and the uniform distribution over $(\mathcal{G}^3)^q$ if we apply τ to (8).

So assume A could distinguish (7) from (8) with probability ϵ, then we can construct an algorithm A' which can distinguish (g^x, g^{xr}, g^r) from a random (a,b,c) with advantage ϵ/q using a hybrid argument. The distribution of the i'th hybrid is

$$(g^x, g^{r_1 x}, g^{r_q}), \ldots, (g^x, g^{r_i x}, g^{r_i}), (g^x, b_{i+1}, c_{i+1}), \ldots, (g^x, b_q, c_q)$$

for random $x, r_1, \ldots, r_q, b_{i+1}, \ldots, b_q, c_{i+1}, \ldots, c_q$. Our A' on input (α,β,γ) chooses a random $i, 1 \leq i \leq q$ and generates the distribution

$$(\alpha, \alpha^{r_1}, g^{r_1}), \ldots, (\alpha, \alpha^{r_{i-1}}, g^{r_{i-1}}), (\alpha,\beta,\gamma), (\alpha, b_{i+1}, c_{i+1}), \ldots, (\alpha, b_q, c_q)$$

whose distribution is equal to the i'th hybrid if (α,β,γ) was generated as (g^x, g^{xr}, g^r) for random x, r, and equal to the $i-1$'th hybrid if (α,β,γ) are three random values.

3 Sequential Composition does Not Imply Adaptive Security

We will define two functions F and G, both $\mathcal{K} \times \mathcal{G}^3 \rightarrow \mathcal{G}^3$ which are indistinguishable from a random function $\mathcal{G}^3 \rightarrow \mathcal{G}^3$ by any *non-adaptive* distinguisher if the DDH-assumption is true in $\mathcal{G}$. But the sequential composition $\mathsf{G} \circ \mathsf{F}$ can be completely broken (i.e. we recover the keys of F and G) with only 3 *adaptive* queries. Unlike in the previous section, here F and G are defined somewhat differently.

Let $\mathsf{R} : \mathcal{K}_{\mathsf{R}} \times \mathcal{G}^3 \to \mathbb{Z}_P^3$ be any adaptively secure pseudorandom function. The keyspace of F and G is $\mathcal{K}_{\mathsf{R}} \times \mathbb{Z}_P$. Let $\phi : \mathcal{K}_{\mathsf{R}} \times \mathbb{Z}_P \to \mathcal{G}^2$ be some encoding of the keyspace of G into a subset of the range of $\mathsf{G} \circ \mathsf{F}$.

F with key $(x \in \mathbb{Z}_P, k_{\mathsf{F}} \in \mathcal{K}_{\mathsf{R}})$ on input (u, v, w) first computes the pseudorandom values

$$(r_1, r_2, r_3) \leftarrow \mathsf{R}_{k_{\mathsf{F}}}(u, v, w)$$

Then the output is computed as (we set the value α as described in Section 1.2 to $(1,1,1)$)

$$\begin{aligned}
\mathsf{F}(1,1,1) &\to (g^x, g^{r_2}, g^{r_3})\\
\mathsf{F}(u \neq 1, v \neq 1, w) &\to \text{ if } u = g^x \text{ then } (v, \phi(k_{\mathsf{F}}, x))\\
&\qquad \text{else } ((u/g^x)^{r_1}, v^{r_1}, g^{r3})\\
\mathsf{F}(\text{ all other cases }) &\to (g^{r_1}, g^{r_2}, g^{r_3})
\end{aligned}$$

G with key $(y \in \mathbb{Z}_P, k_{\mathsf{G}} \in \mathcal{K}_{\mathsf{R}})$ on input (u, v, w) first computes the pseudorandom values

$$(r_1, r_2, r_3) \leftarrow \mathsf{R}_{k_{\mathsf{G}}}(u, v, w)$$

Then the output is computed as

$$\begin{aligned}
\mathsf{G}(u \neq 1, v \neq 1, w) &\to \text{ if } u = g^y \text{ then } (u, v, w)\\
&\qquad \text{elseif } u = v^y \text{ then } (\phi(k_{\mathsf{G}}, y), 1)\\
&\qquad \text{else } (u g^{y r_1}, g^{r_1}, g^{r_3})\\
\mathsf{G}(\text{ all other cases }) &\to (g^{r_1}, g^{r_2}, g^{r_3})
\end{aligned}$$

3.1 Breaking $\mathsf{G} \circ \mathsf{F}$ with 3 Adaptive Queries

We will now describe how to get the key out of $\mathsf{G} \circ \mathsf{F}$ with three adaptive queries. The attack below is successful with probability almost 1. It only fails if by chance P divides one of the random values which appears in the exponent of g below. Let r, r', s, s' be like in the previous section. The first query we make is $(1,1,1)$

$$(1,1,1) \xrightarrow{\mathsf{F}} (g^x, s_{(2,1)}, s_{(3,1)}) \xrightarrow{\mathsf{G}} (g^x g^{y r'_{(1,1)}}, g^{r'_{(1,1)}}, s'_{(3,1)})$$

For the next query we use the first two terms of this output

$$(g^x g^{y r'_{(1,1)}}, g^{r'_{(1,1)}}, 1) \xrightarrow{\mathsf{F}} (g^{y r'_{(1,1)} r_{(1,2)}}, g^{r'_{(1,1)} r_{(1,2)}}) \xrightarrow{\mathsf{G}} (\phi(k_{\mathsf{G}}, y), 1)$$

And we get G's key. Now with the y we just learned and the first output we can compute $g^x = g^x g^{y r'_{(1,1)}} / (g^{r'_{(1,1)}})^y$ and get F's key with the query

$$(g^x, g^y, 1) \xrightarrow{\mathsf{F}} (g^y, \phi(k_{\mathsf{F}}, x)) \xrightarrow{\mathsf{G}} (g^y, \phi(k_{\mathsf{F}}, x))$$

3.2 Non-Adaptive Indistinguishability of F and G

The security of F and G can be reduced to the indistinguishability of R and the hardness of the DDH problem in $\mathcal{G}$ as in section 2.2, i.e.

$$\mathbf{Adv}_{\mathsf{F}}^{non-adaptive}(q,t) \leq \mathbf{Adv}_{\mathsf{R}}(q,t') + \frac{2q}{P} + q\mathbf{Adv}_{DDH}(t') \qquad (9)$$

And the same holds for G. Again we will treat $\mathsf{R}_{k_\mathsf{F}}$ as if it was a truly random function, the $\mathbf{Adv}_{\mathsf{R}(q,t')}$ term in (9) does account for that.

A query to G of the form (u,v,w) where $u=1$ or $v=1$ will just produce a random output. If the i'th query is of the form $(u_i \neq 1, v_i \neq 1, w_i)$ we get as output $(u_i g^{yr_i}, g^{r_i}, *)$ (again $*$ stands for a random value which is independent of all other terms) unless $u_i = v_i^y$ or $u_i = g^y$ for some i, the probability of each such event is exactly P^{-1} as y is random. With the union bound over all $i, 1 \leq i \leq q$ we get an upper bound $2qP^{-1}$ for the probability that any such event happens.

Thus we can assume that the queries are all of the form $(u_i \neq 1, v_i \neq 1, w_i)$ for $i = 1, \ldots, q$ and the output on the i'th query is $(u_i g^{yr_i}, g^{r_i}, *)$ for some random r_i. The distinguisher must now distinguish those $(u_i g^{yr_i}, g^{r_i})$ from sequence of random pairs (b_i, c_i) for $i = 1, \ldots, q$. Or equivalently (as the u_i's are known values) he must distinguish the sequence (g^{yr_i}, g^{r_i}) from random. We are generous and give g^y to the distinguisher. Now we can state that problem as distinguishing a sequence (g^y, g^{yr_i}, g^{r_i}) from (g^y, b_i, c_i) for $i = 1, \ldots, q$, those are exactly the sequences (7) and (8) for which we already proved that they cannot be distinguished with advantage more than $q\mathbf{Adv}_{DDH}(t')$.

Similarly the non-adaptive security of F can be reduced to the task of distinguishing $(u_i^{r_i} g^{-xr_i}, v_i^{r_i})$ for $i = 1, \ldots, q$ from random given g^x. We can assume that the adversary knows $s_i = \log_{v_i}(g), t_i = \log_{v_i}(u_i)$. Then he can map those tuples to $(g^x, g^{xr_i}, g^{r_i}) = (g^x, (u_i^{r_i} g^{-xr_i}/(v_i^{r_i})^{t_i})^{-1}, (v_i^{r_i})^{s_i})$. So again this is equivalent to distinguish the distribution (7) from (8).

4 Conclusions and Further Work

We showed that the sequential or parallel composition of pseudorandom functions with non-adaptive security is not adaptively secure in general if the DDH assumption is true in any group. Some interesting remaining question we're currently looking at are the following:

– Can we prove the same thing with pseudorandom *permutations*, ideally for efficiently invertible ones. This would show that cascading non-adaptively secure block-ciphers will not give adaptive security in general.
– We only gave counterexamples for the composition of two functions. How does this scale to the composition of $n > 2$ functions? This question has been partially answered in [8], where a non-adaptively secure PRF is constructed (under an assumption which is implied by the DDH-assumption) such that the *sequential* composition of *any* number of this PRFs can be distinguished with 2 adaptive queries with high probability. Unfortunately the approach used there seems not to generalise to parallel composition.

– Can we give this result under weaker assumptions or even unconditionally? Of course we may always assume that one-way functions exist (they imply pseudorandomness and vice-versa) as otherwise there's nothing to prove. We give a negative result in this direction in [7]. There we show that if a non-adaptively secure PRF exists where the sequential composition can be distinguished with two queries (as constructed in this paper[5]), then a secure key-agreement protocol exists. Thus any construction of such PRFs must either assume or unconditionally prove the existence of key-agreement (the DDH-assumption we use is know to imply key agreement [1]).
– The domain and the range for our counterexamples is a product of subgroups of $\mathbb{Z}_Q^*$. This is not what one usually does, can we adapt this such that the range and domain are $\{0,1\}^\ell$, ideally with standard bitwise XOR as group operation for parallel composition.

References

1. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.
2. Taher El-Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.
3. Michael Luby and Charles Rackoff. Pseudo-random permutation generators and cryptographic composition. In *Proc, 18th ACM Symposium on the Theory of Computing (STOC)*, pages 356–363, 1986.
4. Ueli Maurer and Krzysztof Pietrzak. Composition of random systems: When two weak make one strong. In *Theory of Cryptograpy — TCC '04*, volume 2951 of *Lecture Notes in Computer Science*, pages 410–427, 2004.
5. Steven Myers. Efficient amplification of the security of weak pseudo-random function generators. *Journal of Cryptology*, 16(1):1–24, 2003.
6. 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.
7. Krzysztof Pietrzak. Exploring minicrypt, 2005. Manuscript.
8. Patrick Pletscher. Adaptive security of composition, 2005. Semester Thesis. Advisors K. Pietrzak and U. Maurer.
9. Serge Vaudenay. Decorrelation: A theory for block cipher security. *Journal of Cryptology*, 16(4):249–286, 2003.

[5] Actually we make three queries, but the third query is only needed to get the key of the second component, we can distinguish already after the second.