Tatra Mt. Math. Publ. **33** (2006), 15–29

Tatra Mountains
Mathematical Publications

RANDOM ORACLE MODEL AND ANALYSIS OF PRIMITIVES OF A CRYPTOGRAPHIC SCHEME

Lenka Fibíková

ABSTRACT. Using CT cipher as an example we show how the Random oracle model can be used to discover weak primitives of a cipher without the need for full cryptanalysis, and how to make changes to them in order to get a provable secure scheme.

1. Introduction

The concept of the random oracle model (ROM), first formalized in [1], provides a nice approach to argue about security of a particular cryptographic scheme. Here we want to show that the ROM is also useful to discover weak primitives of a scheme without the need for a full cryptanalysis, and it enables one to select the "best" primitive for a given scheme.

We illustrate our approach using the encryption scheme from [2] as an example. Namely, using the ROM we show that the hash functions used there are weak and even the enhancement of the scheme, proposed in [6], does not compensate for the weakness. Subsequently we investigate other hash functions and their use in the scheme, and show how one can select the best one with respect to security of the overall scheme.

2. ROM and proof of security

Our security model is based on two general ideas: The first one is the ROM ([1]), which enables one to simplify and decompose the proof of security by replacing the primitives of a cryptographic function with perfect random functions, which simplifies the proof significantly. The other one is the distinguishing approach ([6]). It can be viewed as a certain game in which an attacker has access

2000 Mathematics Subject Classification: 94A60.
Keywords: cryptoanalysis, ROM, distinguisher, provable security, unbalanced Feistel network, cryptographic primitives, hash function.

The work was done during the PhD study at Institute of Experimental Mathematics, University Duisburg-Essen.

to an oracle that implements either the cryptographic function or a perfect random function. Querying the oracle limited number of times and using its answers and infinite computation power, the attacker has to decide which function the oracle implements, and output 1 ("accept") if it is the function, or 0 ("reject") if it is a perfect random function. Since the goal of the attacker is to distinguish two functions, it is called a *distinguisher*. Different types of distinguishers may be defined, depending on the type of attack they perform (e.g., chosen-plaintext attack, differential cryptanalysis, etc.).

The success of a distinguisher D to distinguish between a cryptographic function F and a random function F^* is determined by the probability of answering "accept" when the oracle implements F (a correct answer)—p_0, and probability of answering "accept" when the oracle implements F^* (an incorrect answer) p_1. The overall ability of a distinguisher D implementing an attack ATK and querying the oracle up to d times to distinguish two functions is measured by *advantage* defined as $\mathrm{Adv}_D^{\mathrm{ATK}(d)}(F_1, F_2) = |p_0 - p_1|$. The advantage of the best distinguisher implementing an attack ATK and querying the oracle up to d times is denoted by $\mathrm{Adv}^{\mathrm{ATK}(d)}(F) = \max\limits_D \{\mathrm{Adv}_D^d(F)\}$.

These two approaches enable one to prove security of a composed scheme $F = \Omega[F_1, \ldots, F_r]$ built on primitives $F_1, \ldots, F_r$ in the following four steps:

1. *Definition of a class of attacks the cipher should be secure against.*

In our analysis we will consider two types of attacks—the chosen plaintext attacks and the adaptive chosen plaintext–ciphertext attacks.

2. *Evaluation of advantage of the scheme in the random oracle model.*

The primitives of the schemes are substituted by perfect random functions, and advantage of $F' = \Omega[F_1^*, \ldots, F_r^*]$ is calculated.

3. *Evaluation of advantage of the individual cryptographic primitives of the scheme.*

The advantage of individual functions $F_1, \ldots, F_r$ is calculated separately. In a more complex scheme some of them may be decomposed in the similar way as the main scheme; then the advantage may be calculated recursively.

4. *Proof of how the advantage of the underlying primitives propagates to the scheme.*

In this step results of Steps 2 and 3 are combined into a final advantage. For any d-limited distinguisher between F and a perfect random function the advantage is upper-bounded by the sum of advantage of the scheme in the ROM (Step 2) and advantages of its primitives (Step 3) as follows ([3])

$$\mathrm{Adv}^{\mathrm{ATK}(d)}(F) \le \mathrm{Adv}^{\mathrm{ATK}(d)}(F') + \sum_{i=1}^{r} \mathrm{Adv}^{\mathrm{ATK}\,(a_i d)}(F_i).$$

Note that this model of security is very strong. If we can prove that a cryptographic function is secure in this model, its output looks perfectly random. On the other hand, if we prove that a function is not secure in this model, it only means that an attacker has a high probability to distinguish its output from a perfectly random output. This fact does not say anything about how secure the function is against weaker attacks which are more practical for an attacker, for example finding an encryption key for a particular ciphertext, or being able to encrypt/decrypt another message.

3. CT scheme

The encryption scheme from [2] (we will further call it *CT scheme*) is a fully scalable cipher based on the unbalanced Feistel network dividing input into two parts: a small left part of m bits and a big right part of nm $(n > 1)$ bits. In general, there are two unbalanced Feistel schemes, yielding to the *source heavy* (see Figure 1A), or *target heavy* scheme (see Figure 1B) [5]. The CT cipher combines both these approaches (see Figure 1C).

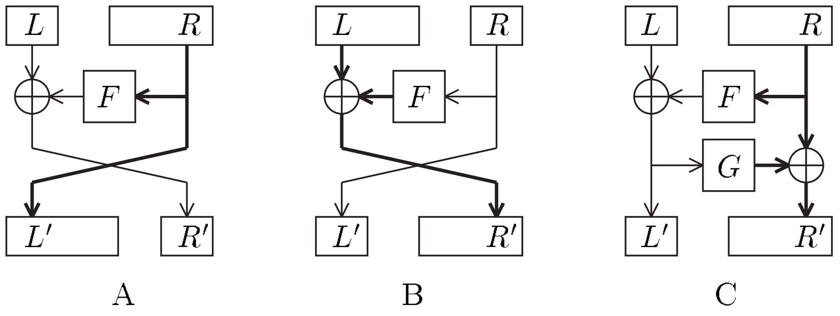
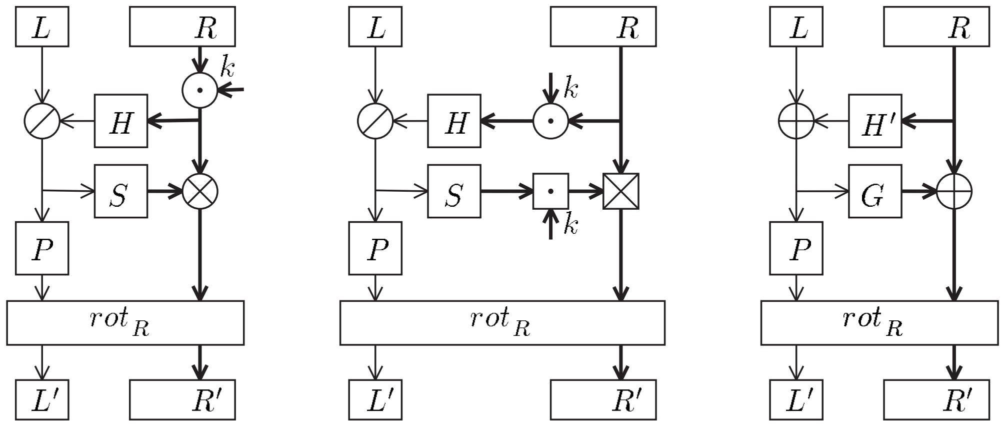


FIGURE 1. Unbalanced Feistel network and the CT cipher.

Although the CT cipher is based on the mixed unbalanced Feistel scheme, it makes some modifications to it (see Figure 2A: It changes the XOR operation to any invertible operation on binary strings ($\odot$, $\oslash$, and $\otimes$), uses unkeyed round functions with addition of a round key to the right part before each (double)-round, adds a new permutation to the left part, and applies string rotation after each (double)-round. In general, the function H may be any hash function, S any random sequence generator, and P any random permutation ([2]).

Since the function H does not depend on the key (it is not a real primitive of the scheme), we are not able to use the ROM directly. For this reason we make some changes to the scheme which, however, do not weaken its security. First, the addition of the key k ($\odot$) may be attached to both functions H, and S: $H'(X) = H(X \odot k)$, $G(X) = k \boxdot S(X)$ (it means that both sub-rounds



A B C

FIGURE 2. CT and its simplification.

of a double-round have the same key), and the $\otimes$ is substituted by another operation $\boxtimes$, where $\boxdot$ and $\boxtimes$ are such that $a \boxtimes (b \boxdot c) = (a \odot b) \otimes c$ for all entries (see Figure 2B). Even if such a pair of functions does not exist, one may use $\boxdot = \odot$ and $\boxtimes = \otimes$. In that case the modified scheme is not equivalent to the original scheme, but we buy flexibility in choice of the functions H' and G, which actually introduce randomness into the scheme.

For compatibility with the unbalanced Feistel scheme we will first consider only XOR for $\oslash$ and $\boxtimes$. However, in general $\oslash$ and $\boxtimes$ may be any randomness-preserving operation, because, as mentioned before, functions H' and G introduce the randomness, and the operations $\oslash$ and $\boxtimes$ just transfer the randomness to the data blocks. Thus we may do this substitution without loss of generality.

Using these substitutions, one gets the simplified form of CT, which actually is a mixed unbalanced Feistel scheme with two additional operations—the permutation P, and the final rotation rot_R (see Figure 2C). Note that the simplification is only in its form. In fact, functions H' and G may be any key-dependent functions with the appropriate input and output sizes, thus the scheme is more general and easier to optimize.

Security of the unbalanced Feistel in the ROM was discussed in several papers (e.g., [4]), and the results can be easily adopted to the CT scheme. Using perfect random primitives we get:

- 1-round CT scheme is secure against the chosen plaintext attacks ($\mathrm{Adv}^{\mathrm{CPA}(d)}(\mathrm{CT}[H^*, S^*, P^*]) \le d^2/2^m$);
- 1-round CT scheme is not secure against the adaptive chosen plaintext attacks;
- 2-round CT scheme is secure against the adaptive chosen plaintext–ciphertext attacks

$$(\mathrm{Adv}^{\mathrm{ACPCA}(d)}(\mathrm{CT}[H_1^*, S_1^*, P_1^*, H_2^*, S_2^*, P_2^*]) \leq d^2/2^m);$$

where m is the size of the left part in bits.

The cipher is built on three primitive functions: a hash function H, an S-box S and a permutation P. However, the functions S and P are generated by a pseudorandom generator and considered to be unpredictable (i.e., $\mathrm{Adv}^d(S) = \mathrm{Adv}^d(P) = 0$ for any d), thus they do not decrease security of the scheme. We will therefore concentrate on the influence of the hash function on the overall security of the scheme: First, we prove that the hash functions proposed in [2] are weak. We do not show how to exploit the weaknesses of the primitive to break the cipher, which is not possible in our model, but we show how a sequence of improvements in the hash function may result in a construction of a provable secure cipher.

4. Hash function H

To analyze the hash function H, we make use of the ROM approach—by constructing a distinguisher for a particular primitive we do not only show that the primitive is weak, but from the construction we also get some insight on which property of the primitive is "responsible" for the weakness in the sense that the distinguisher makes use of this property to distinguish between the primitive and a perfect random function.

4.1 Weak structure

First, consider the hierarchical structure H_1—one of the schemes proposed in [2]—depicted in Figure 3.

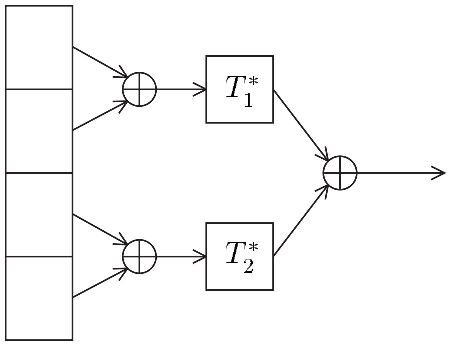


FIGURE 3. Hierarchical structure H_1.

Since the functions T_i are executed only after the XOR functions, any two inputs containing two blocks leading into the same execution of T_i with equal

XOR cause equal outputs of the hash function. Thus we can build the following distinguisher.

Distinguisher 1 (D_1): 2-limited CPA distinguisher for H_1

---

1. Create $X_1 = (x_1, x_2, x_3, \ldots, x_n)$ at random.
2. Create $X_2 = (x_1 \oplus c, x_2 \oplus c, x_3, \ldots, x_n)$ for a non-zero constant c.
3. Query the oracle with X_1 and X_2, and get Y_1 and Y_2, where Y_i is either $H_1(X_i)$ or $F^*(X_i)$.
   If $Y_i = H_1(X_i)$, then XORing the first two blocks in both of the plaintext messages eliminates the constant c ($x_1 \oplus x_2 = x_1 \oplus c \oplus x_2 \oplus c$), and all T_is are executed with the same arguments for both X_1 and X_2. Therefore $Y_1 = Y_2$.
4. If $Y_1 = Y_2$, then output "accept".
5. Output "reject".

When the oracle implements H_1, the distinguisher D_1 always answers correctly, i.e., $p_0 = 1$. When the oracle implements a perfect random function, the probability that two random strings of length m are equal is $p_1 = \frac{1}{2^m}$. Therefore the advantage of this 2-limited distinguisher is

$$\mathrm{Adv}_{D_1}^{\mathrm{CPA}(2)} = |p_0 - p_1| = 1 - \frac{1}{2^m}.$$

The advantage may be increased by adding further chosen plaintexts in similar way as X_2, so that $\mathrm{Adv}^{\mathrm{CPA}(d)} \geq 1 - \frac{1}{2^{m(d-1)}}$. Thus the hierarchical structure H_1 is not secure against the chosen plaintext attacks.

In the original design of CT a round key is added to the right part of the plaintext, and only then it is hashed (Figure 2a). If the key is XORed with the plaintext, the changes in the second message are eliminated as described in the distinguisher D_1. Authors of [2] suggested to use modular addition operation to combine round keys with messages in order to make an attack more difficult. However, we can show that there is no improvement on the XOR function in the ROM.

Consider the following distinguisher between H_1 and a perfect cipher:

Distinguisher 2 (D_2): 2-limited CPA distinguisher for H_1

---

1. Create $X_1 = \big((00\ldots0)_2, (00\ldots0)_2, x_3, \ldots, x_n\big)$ at random.
2. Create $X_2 = \big((10\ldots0)_2, (10\ldots0)_2, x_3, \ldots, x_n\big)$.
3. Query the oracle with X_1 and X_2, and get Y_1 and Y_2, where Y_i is either $H_1(X_i)$ or $F^*(X_i)$.

If $Y_i = H_1(X_i)$, and the added key is $k = (k_1, k_2, \ldots, k_n)$, then for the first plaintext the first XOR is $(0 + k_1) \oplus (0 + k_2)$. For the second plaintext we have $(2^{m-1} + k_1) \oplus (2^{m-1} + k_2)$. The only bits changed in $2^{m-1} + k_i$ ($i \in \{1, 2\}$) are the most significant ones, and since one-bit addition is actually XOR function, they are equal for both plaintexts. Therefore $k_1 \oplus k_2 = (2^{m-1} + k_1) \oplus (2^{m-1} + k_2)$, and $Y_1 = Y_2$.

4. If $Y_1 = Y_2$, then output "accept".
5. Output "reject".

When the oracle implements H_1, the distinguisher D_2 always answers correctly, i.e., $p_0 = 1$. When the oracle implements a perfect random function, the probability that the distinguisher accepts is $p_1 = \frac{1}{2^m}$. Therefore $\mathrm{Adv}^{\mathrm{CPA}(2)}(H_1) \geq \mathrm{Adv}_{D_2}^{\mathrm{CPA}(2)}(H_1) = |p_0 - p_1| = 1 - \frac{1}{2^m}$.

4.2 Weak underlying primitive

Now we take a look at the second structure of the hash function suggested in [2]. It is the scheme depicted in Figure 4, using an unkeyed function f for T. Here we can apply the following attack:

Distinguisher 3 (D_3): 2-limited CPA distinguisher for H_2

---

1. Create $X_1 = (x_1, x_2, x_3, \ldots, x_n)$ at random.
2. Create $X_2 = (x_1 \oplus a, x_2 \oplus c, x_3, \ldots, x_n)$ for any two constants a and c such that $f(x_1 \oplus a) = f(x_1) \oplus c$, and at least one of them is nonzero.
3. Query the oracle with X_1 and X_2, and get Y_1 and Y_2, where Y_i is either $H_2(X_i)$ or $F^*(X_i)$.

   If $Y_i = H_2(X_i)$, then the calculation for the message X_1 follows this calculation sequence:

$$
\begin{aligned}
r_1 &= f(x_1)\,, \\
r_i &= f(r_{i-1} \oplus x_i) \qquad \text{for all } i = 2, \ldots, n\,.
\end{aligned}
$$

   The second calculation sequence starts with:

$$
\begin{aligned}
r_1' &= f(x_1 \oplus a) = f(x_1) \oplus c = r_1 \oplus c\,, \\
r_2' &= f(r_1 \oplus c \oplus x_2 \oplus c) = f(r_1 \oplus x_2) = r_2\,,
\end{aligned}
$$

   and thus in all further steps they both follow the same computation sequence. Therefore $Y_1 = Y_2$.
4. If $Y_1 = Y_2$, then output "accept".
5. Output "reject".

When the oracle implements H_2, the distinguisher D_3 always answers correctly, and therefore the advantage of this 2-limited distinguisher is again

$$\mathrm{Adv}_{D_3}^{\mathrm{CPA}(2)} = |p_0 - p_1| = 1 - \frac{1}{2^m}.$$

The advantage may be increased by adding further chosen plaintexts in a similar way as X_2, so that $\mathrm{Adv}^{\mathrm{CPA}(d)} \geq 1 - \frac{1}{2^{m(d-1)}}$.

In order to increase security of the hash function, authors of [2] suggested a double-scheme consisting of two executions of the hash function H_2—one as described above, and one with a shifted message—and finally XORing their outputs, i.e., $H_2'(x_1, \dots, x_n) = H_2(x_1, \dots, x_n) \oplus H_2(x_n, x_1, \dots, x_{n-1})$. However, when the messages are constructed as in the distinguisher D_3, the difference is eliminated when the modified blocks are on any place in the message provided that they follow each other as described. Thus the double structure does not bring any improvement against this distinguishing attack.

5. ROM and selection of the best candidate

Here we show how careful analysis of various candidates in the spirit of the ROM approach and subsequent modifications allows us to construct hash schemes with small advantage, and to choose the best one.

5.1 Serial structure

Consider the same structure as H_2, but with perfect random function T^* as the primitive (let us denote it H_3).

In [7] it was shown that advantage of this scheme is

$$\mathrm{Adv}^{\mathrm{ACPA}(d)}(H_3) \leq \frac{(dn)^2}{2^{m+1}}.$$

Advantage of this scheme can be improved by using different random functions $T_1^*, \dots, T_n^*$ for each m-bit block of the input (let us denote the scheme H_4):

THEOREM 5.1. *Let H_4 be a function from $\{0,1\}^{nm}$ to $\{0,1\}^m$ defined for any $X = (x_1, \dots, x_n) \in \{0,1\}^{nm}$ as*

$$H_4(x_1, \dots, x_n) = T_n^*\big(T_{n-1}^*(\dots T_1^*(x_1) \oplus x_2 \dots) \oplus x_n\big),$$

where T_i^*s are independent perfect random functions from $\{0,1\}^m$ to $\{0,1\}^m$ (see Figure 4). Then for any integer d,

$$\mathrm{Adv}^{\mathrm{ACPA}(d)}(H_4) \leq \frac{nd^2}{2^{m+1}}.$$

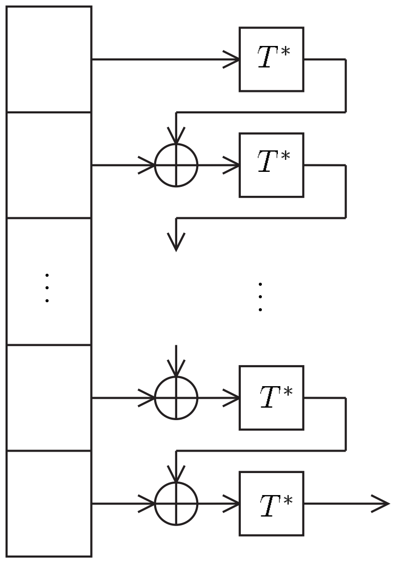


FIGURE 4. Serial structure H_2–H_4.

P r o o f. Assume that the attacker has d plaintext/ciphertext pairs (X_i, Y_i) at his disposal, such that $X_i = (x_{i,1}, \ldots, x_{i,n})$, $Y_i = (y_{i,1}, \ldots, y_{i,n})$, and $1 \le i \le d$. Without loss of generality we may assume that all X_is are pairwise different. For each input X_i, the computation of the function H_4^* follows these steps:

1. $y_{i,1} = T_1^*(x_{i,1})$,
2. $y_{i,a} = T_a^*(y_{i,a-1} \oplus x_{i,a})$, for all $1 < a \le n$,

and $Y_i = y_{i,n}$

Inputs to the individual executions of T^* can be collected into a $d \times n$ matrix I, so that $I_{i,a} = y_{i,a-1} \oplus x_{i,a}$, and $y_{i,0} = 0$, for all i. Thus the rows contain the computational sequences, and the column inputs to the same instance of the function T^*.

If inputs into the last instance of T^*, i.e., $I_{i,n}$, are all pairwise distinct, then all outputs Y_i are perfect random, since T^* is perfect random and the attacker cannot distinguish them from random outputs.

In the following we look at the probability that $I_{i,n}$ are all distinct.

Let `ColFree` be an event that for all pairs of inputs $I_{i,a}$ and $I_{j,b}$ such that $1 \le i, j \le d$, and $1 \le a, b < n$, if $I_{i,a} \ne I_{j,b}$, then $y_{i,a} \ne y_{j,b}$, i.e., that T^* returns different outputs for different inputs. The probability that there is a collision is

$$\Pr[\,\overline{\texttt{ColFree}}\,] \le (n-1)\binom{d}{2}\frac{1}{2^m}.$$

When `ColFree` occurs,

$$\Pr[I_{i,n} = I_{j,n}] = \Pr[y_{i,n-1} \oplus x_{i,n} = y_{j,n-1} \oplus x_{j,n}]$$
$$= \begin{cases} \Pr[y_{i,n-1} = y_{j,n-1}] \overset{\texttt{ColFree}}{=} \Pr[I_{i,n-1} = I_{j,n-1}] & \text{if } x_{i,n} = x_{j,n}\,, \\ \Pr[y_{i,n-1} = y_{j,n-1} \oplus x_{i,n} \oplus x_{j,n}] & \text{if } x_{i,n} \neq x_{j,n}\,. \end{cases}$$

Let r be the smallest integer such that $x_{i,n-r} \neq x_{j,n-r}$. Note that $r \geq 1$, because $x_i \neq x_j$. Then

$$\begin{aligned} \Pr[I_{i,n} = I_{j,n}] &= \Pr[I_{i,n-1} = I_{j,n-1}] = \cdots = \Pr[I_{i,n-r} = I_{j,n-r}] \\ &= \Pr[y_{i,n-r-1} = y_{j,n-r-1} \oplus x_{i,n-r} \oplus x_{j,n-r}] \\ &= \begin{cases} 0 & \text{if } I_{i,n-r-1} = I_{j,n-r-1} \\ \frac{1}{2^m} & \text{if } I_{i,n-r-1} \neq I_{j,n-r-1} \end{cases} \leq \frac{1}{2^m}\,, \end{aligned}$$

and

$$\begin{aligned} \mathrm{Adv}^{\mathrm{ACPA}(d)}(H_4) &= 1 - \Pr[\forall\, i,j\colon I_{i,n} \neq I_{j,n}] = \Pr[\exists\, i,j\colon I_{i,n} = I_{j,n}] \\ &\leq \Pr\big[\exists\, i,j\colon I_{i,n} = I_{j,n} \wedge \texttt{ColFree}\big] + \Pr\big[\,\overline{\texttt{ColFree}}\,\big] \\ &\leq \binom{d}{2}\frac{1}{2^m} + \binom{d}{2}\frac{n-1}{2^m} \\ &\leq \frac{nd^2}{2^{m+1}}\,. \end{aligned}$$

□

5.2 Parallel structure

We have already shown that message blocks in the hierarchical structure H_1 should not be XORed before execution of the functions T_i. The simplest way how to avoid this is to apply the functions on the message blocks first, and then simply XOR the results together, thus getting the scheme H_5 depicted in Figure 5.

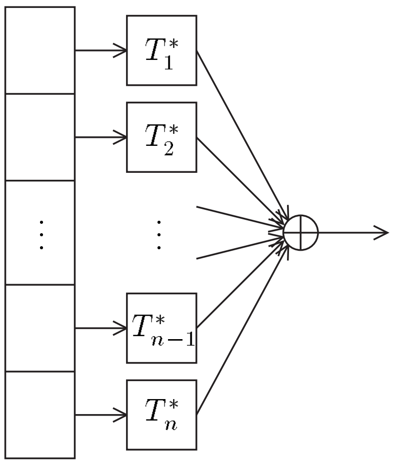


FIGURE 5. Parallel structure H_5.

This scheme is indistinguishable from a perfect random one as long as an attacker cannot obtain more then three plaintext/ciphertext pairs:

THEOREM 5.2. *Let H_5 be a function from $\{0,1\}^{nm}$ to $\{0,1\}^m$ defined for any $X = (x_1, \dots, x_n) \in \{0,1\}^{nm}$ as*

$$H_5(x_1, \dots, x_n) = \bigoplus_{a=1}^{n} T_a^*(x_a)\,,$$

where T_i^* are independent perfect random functions from $\{0,1\}^{nm}$ to $\{0,1\}^m$ (see Figure 6). Then for all $d \le 3$,

$$\mathrm{Adv}^{\mathrm{ATK}(d)}(H_5) = 0\,.$$

P r o o f . Without loss of generality assume that $X_1, X_2, \dots, X_d$ are all different. Let X_i consist of n blocks, i.e., $X_i = x_{i,1}, \dots x_{i,n}$. Let $z_{i,k} = T_k^*(x_{i,k})$.

If $d = 1$:

$$\begin{aligned}
\Pr\big[H_5(x_1) = y_1\big] &= \sum_{\substack{z_{1,1},\dots,z_{1,n}\\ \bigoplus_{a=1}^{n} z_{1,a}=y_1}} \Pr\left[\bigwedge_{a=1}^{n} T_a^*(x_{1,a}) = z_{1,a}\right] \\
&= \sum_{\substack{z_{1,1},\dots,z_{1,n}\\ \bigoplus_{a=1}^{n} z_{1,a}=y_1}} \prod_{a=1}^{n} \Pr\big[T_a^*(x_{1,a}) = z_{1,a}\big] \\
&= 2^{m(n-1)} \cdot \frac{1}{2^{mn}} = \frac{1}{2^m} = \Pr\big[F^*(x_1) = y_1\big]\,.
\end{aligned}$$

If $d = 2, 3$:

Assume that for ath block there are $k_a \in \{0, \dots, d\}$ different values among $x_{1,a}, x_{2,a}, \dots, x_{d,a}$. Since all T_a s are independent, for any $z_{1,a}, z_{2,a}, \dots, z_{d,a}$

$$\Pr\big[\forall i \le d\colon T_a^*(x_{i,a}) = z_{i,a}\big] = \begin{cases} 0\,, & \exists\, i,j\colon x_{i,a} = x_{j,a} \wedge z_{i,a} \ne z_{j,a}\,,\\ \dfrac{1}{2^{mk_a}}\,, & \text{otherwise}\,. \end{cases}$$

Therefore

$$\Pr\left[\bigwedge_{i=1}^{d} H_4^*(x_i)=y_i\right] = \sum_{\substack{i=2,3:\ z_{i,1},\dots,z_{i,n} \\ \bigoplus_{a=1}^{n} z_{i,a}=y_i}} \Pr\left[\bigwedge_{i=1}^{d}\bigwedge_{a=1}^{n} T_a^*(x_{i,a})=z_{i,a}\right]$$

$$= \sum_{\substack{i=2,3:\ z_{i,1},\dots,z_{i,n} \\ i:\ \bigoplus_{a=1}^{n} z_{i,a}=y_i \\ a(d-k_a\text{equations}):\ z_{i,a}=z_{j,a}}} \prod_{a=1}^{n} \frac{1}{2^{mk_a}}$$

$$= 2^{m\left((n-1)\cdot d-\sum_{a=1}^{n}(d-k_a)\right)} \cdot 2^{-m\sum_{a=1}^{n}k_a} = \frac{1}{2^{md}}\,. \qquad \square$$

However, when the attacker may choose plaintext messages, it is possible to distinguish this scheme using the following distinguisher:

Distinguisher 4 (D_4): 4-limited CPA distinguisher for H_1

1. Create $X_1=(s,u,x_3,\dots,x_n)$ at random.
2. Create $X_2=(s,v,x_3,\dots,x_n)$ so that $u\neq v$.
3. Create $X_3=(t,u,x_3,\dots,x_n)$ so that $s\neq t$.
4. Create $X_4=(t,v,x_3,\dots,x_n)$.
5. Query the oracle with X_1, X_2, X_3 and X_4, and get Y_1, Y_2, Y_3 and Y_4, where Y_i is either $H_5(X_i)$ or $F^*(X_i)$.
   If $Y_i=H_5(X_i)$, then

   $$\begin{aligned} T_1^*(s)\oplus T_2^*(u)\oplus T_3^*(x_3)\cdots\oplus T_n^*(x_n)&=Y_1\,,\\ T_1^*(s)\oplus T_2^*(v)\oplus T_3^*(x_3)\cdots\oplus T_n^*(x_n)&=Y_2\,,\\ T_1^*(t)\oplus T_2^*(u)\oplus T_3^*(x_3)\cdots\oplus T_n^*(x_n)&=Y_3\,,\\ T_1^*(t)\oplus T_2^*(v)\oplus T_3^*(x_3)\cdots\oplus T_n^*(x_n)&=Y_4\,.\end{aligned}$$

   Therefore $0=Y_1\oplus Y_2\oplus Y_3\oplus Y_4$.
6. If $Y_1\oplus Y_2\oplus Y_3\oplus Y_4=0$ then output "accept".
7. Output "reject".

When the oracle implements H_5, the distinguisher D_4 always answers correctly, i.e., $p_0=1$. When the oracle implements a perfect random function, probability that XOR of four random strings of length m gives 0 is $p_1=\frac{1}{2^m}$. Therefore advantage of this 4-limited distinguisher is

$$\mathrm{Adv}_{D_4}^{\mathrm{CPA}(4)} = |p_0-p_1| = 1-\frac{1}{2^m}\,.$$

The advantage may be increased by adding further chosen plaintexts in a similar way as X_3 and X_4, so that $\mathrm{Adv}^{\mathrm{CPA}(2d)} \geq 1 - \frac{1}{2^{m(d-1)}}$.

This scheme is therefore not secure, when the attacker can obtain more then three plaintext/ciphertext pairs. However, by adding another random function to the output of the scheme (see Figure 6), we can achieve small advantage even for bigger values of d, while retaining the zero advantage for $d \leq 3$.

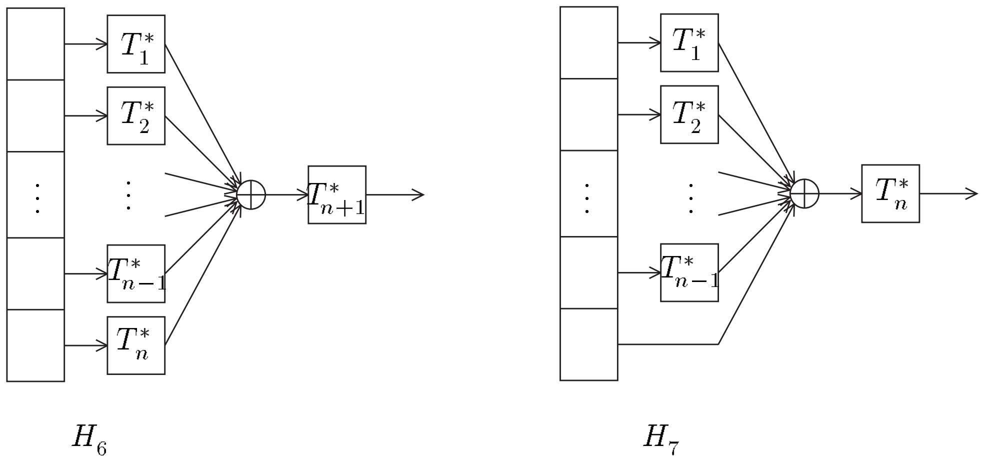


FIGURE 6. Parallel structures H_6 and H_7.

This scheme is less efficient due to the additional random function. However, one of the first-level functions may be omitted (see Figure 6 without increasing the upper bound of its advantage. Since the proofs of security for both schemes H_6 and H_7 are very similar, we only present here the proof for the later one.

THEOREM 5.3. *Let H_7 be a function from $\{0,1\}^{nm}$ to $\{0,1\}^m$ defined for any $X = (x_1, \dots, x_n) \in \{0,1\}^{nm}$ as*

$$H_7(x_1, \dots, x_n) = T_n^* \left(x_n \oplus \bigoplus_{a=1}^{n-1} T_a^*(x_a) \right),$$

where $T_1^*, \dots T_n^*$ are independent perfect random functions from $\{0,1\}^m$ to $\{0,1\}^m$. Then for any integer d,

$$\mathrm{Adv}^{\mathrm{ATK}(d)}(H_7) \leq \frac{d^2}{2^{m+1}}.$$

P r o o f . If all inputs to T_n^* are pairwise distinct, then the outputs are perfect random, since the function T_n^* is perfect random.

Assume that all inputs are pairwise distinct. Consider two cases: First, when in a pair of plaintexts X_i, X_j there are all blocks up to the last one equal, i.e., when for all $1 \leq a < n$, $x_{i,a} = x_{j,a}$. Then $x_{i,n} \neq x_{j,n}$. Let $A := \bigoplus_{a=1}^{n-1} T_a^*(x_{i,a}) = \bigoplus_{a=1}^{n-1} T_a^*(x_{j,a})$.

$$\Pr[Z_i = Z_j] = \Pr[x_{i,n} \oplus A = x_{j,n} \oplus A] = 0\,.$$

If there is $r < n$ such that $x_{i,r} \neq x_{j,r}$, then

$$\begin{aligned}\Pr[Z_i = Z_j] &= \Pr\Biggl[T_r^*(x_{i,r}) = T_r^*(x_{j,r}) \oplus x_{i,n} \oplus x_{j,n} \oplus \bigoplus_{\substack{1\le a<n\\ a\neq r}}^{n} T_a^*(x_{i,a}) \oplus \bigoplus_{\substack{1\le a<n\\ a\neq r}} T_a^*(x_{j,a})\Biggr]\\ &= \frac{1}{2^m}\,.\end{aligned}$$

Hence

$$\begin{aligned}\mathrm{Adv}^{\mathrm{ATK}(d)}(H_6) &= 1 - \Pr[\forall\, i,j\colon Z_i \neq Z_j] = \Pr[\exists\, i,j\colon Z_i = Z_j]\\ &\le \sum_{1\le i<j\le d} \Pr[Z_i = Z_j] \le \binom{d}{2}\frac{1}{2^m} \le \frac{d^2}{2^{m+1}}\,.\end{aligned}$$

□

6. Conclusion

Using CT scheme as an example we have shown that the ROM can be used to investigate not only security of a cryptographic scheme, but also to gain some insight into the "security contributions" of its particular constituents (primitives). By this way it allows one to understand where the weak point that needs to be strengthened lies. In the same spirit the approach also allows one to assess security properties of different candidates for a primitive and to select the best one.

We have applied this approach to the CT scheme to show that the hash function H_2 recommended in [2] is weak, although we have not shown how its weakness can be exploited. Also, we have used the approach to carefully analyze a sequence of improvements resulting in the construction of another hash scheme H_7, which is provably secure—it is perfect random when the attacker can obtain up to 3 plaintext/ciphertext pairs, and for attacks of larger sizes the probability of distinguishing it from a perfect random function is still small if the size of the attack $d \ll 2^{\frac{nm}{2}}$.

REFERENCES

[1] BELLARE, M.—ROGAWAY, P.: *Random oracles are practical: A paradigm for designing efficient protocols*, in: Proc. of the 1th ACM Conference on Computer and Communications Security, 1993, pp. 62–73.

[2] ČANDA, V.—TRAN VAN TRUNG: *Scalable block ciphers based on Feistel-like structure*, TATRACRYPT '01, Tatra Mt. Math. Publ. **25** (2002), 39–66.

[3] FIBÍKOVÁ, L.: *Provably Secure Scalable Block Ciphers*, PhD thesis, Institute of Experimental Mathematics, University Duisburg-Essen, 2003.

[4] LUBY, M.—RACKOFF, CH.: *How to construct pseudorandom permutations from pseudo--random functions*, Proceedings of CRYPTO '85, Springer-Verlag, Berlin, 1986, pp. 447.

[5] SCHNEIER, B.—KELSEY, J.: *Unbalanced Feistel networks and block–cipher design*, in: Lecture Notes Comput. Sci., Vol. 1039, Springer-Verlag, Berlin, 1996, pp. 121–144.

[6] VAUDENAY, S.: *Provable security for block ciphers by decorrelation*, in: STACS, Vol. 1373, 1998, pp. 249–275.

[7] VAUDENAY, S.: *On provable security for conventional cryptography*, in: Information Security and Cryptology ICISC '99, Springer-Verlag, Berlin, 1999, pp. 1–16.

Received September 24, 2004

DaimlerChrysler Financial Services AD
Information Security
098/Y623–ITF/ITO
Leitzstraße 54
G–70469 Stuttgart
GERMANY

E-mail: lenka.fibikova@daimlerchrysler.com