Second Preimages for Iterated Hash Functions Based on a b-Block Bypass [⋆]

Mario Lamberger, Norbert Pramstaller, and Vincent Rijmen

Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology, Austria
Norbert.Pramstaller@iaik.tugraz.at

Abstract. In this article, we present a second preimage attack on a double block-length hash proposal presented at FSE 2006. If the hash function is instantiated with DESX as underlying block cipher, we are able to construct second preimages deterministically. Nevertheless, this second preimage attack does not render the hash scheme insecure. For the hash scheme, we only show that it should not be instantiated with DESX but AES should rather be used. However, we use the instantiation of this hash scheme with DESX to introduce a new property of iterated hash functions, namely a so-called *b-block bypass*. We will show that if an iterated hash function possesses a b-block bypass, then this implies that second preimages can be constructed. Additionally, the attacker has more degrees of freedom for constructing the second preimage.

Keywords: iterated hash functions, double block-length hash functions, differential cryptanalysis, second preimage

1 Introduction

A cryptographic hash function maps a binary string of arbitrary length to a fixed length binary string, called hash value. A cryptographic hash function H has to be secure against the following attacks:

- **Collision attack:** Find two messages m and $m^* \neq m$ such that $H(m) = H(m^*)$
- **Preimage attack:** For a given hash value h, find a message m such that $H(m) = h$
- **Second preimage attack:** For a given message m, find a second message $m^* \neq m$ such that $H(m) = H(m^*)$

Based on the birthday paradox the expected complexity for a collision attack is about $2^{n/2}$ hash computations, where n is the size of the hash value. For a preimage attack and a second preimage attack the complexity is about 2^n hash computations. If for a given hash function H, collisions and (second) preimages

[⋆] The work in this paper has been supported by the Austrian Science Fund (FWF), project P18138.

can be found with a complexity less than $2^{n/2}$ and 2^n, respectively, the hash function is considered to be broken.

An alternative to dedicated hash functions such as MD5 and SHA-1 are hash functions that are based on block ciphers. In [13], Preneel analyzed possible constructions of hash functions using block ciphers in different modes of operation. Out of 64 schemes, 12 are considered to be secure. Black *et al.* have proven the security of these schemes in the ideal cipher model in [3]. A drawback of block cipher based hash functions is the limited output length. More precisely, by using a block cipher with an output length of for instance 128 bits, the complexity to find a collision is 'only' about 2^{64} hash computations. It is clear that with the increase of available computational power such an output size does not give a satisfying security margin (especially, if a hash function should be secure for the next decades). A possible and common approach to overcome the limitation of the security due to the output size is to use two or more block ciphers and concatenate their outputs. Such schemes are referred to as "double block-length hash functions" if the output of two block ciphers is concatenated. A recent example is the proposal of Hirose [5] at FSE 2006. He proposes a double block-length hash function and proves the security in the ideal cipher model. For the remainder of this article, we will refer to this hash function as "DBLH".

In general, the approach to prove the security of a cryptographic scheme in the ideal cipher model (black box model) has attracted a lot of attention and is of common use. Nevertheless, also doubts about the implications of such security proofs do exist. In other words, it is an open question what one can expect from a scheme that is provably secure in the ideal cipher model once the scheme is instantiated with a concrete block cipher. A very recent paper covering security proofs and the related doubts was published by Black at FSE 2006 [2].

The main contribution of this article is as follows. Firstly, we define a new property of an iterated hash function, namely a b-block bypass. We will show that if an iterated hash function possesses a b-block bypass then second preimages can be constructed. Secondly, we show that for the double block-length hash function DBLH instantiated with DESX as underlying block cipher, we can construct a b-block bypass (depending on the configuration $b = 2, 3$) by exploiting differential cryptanalysis. This leads to a powerful second preimage attack on the hash scheme DBLH with DESX.

The remainder of this article is structured as follows. In Section 2, we give notation and definitions. We introduce the definition of a b-block bypass for an iterated hash function and show the implications for second preimage attacks in Section 3. In Section 4, we describe the double block-length hash proposal of Hirose and give the definition of the block cipher DESX. Furthermore, we show different configurations of how the hash scheme can be instantiated with DESX as underlying block cipher. Section 5 is dedicated to the construction of a b-block bypass for different configurations of the hash scheme DBLH with DESX. In Section 6, we discuss our results and present conclusions in Section 7.

2 Preliminaries

2.1 Notation

For the concatenation of two variables, we write $a||b$. Addition modulo 2 (XOR) is denoted by $a \oplus b$. The bit length of variable a is denoted by $|a|$. We stick to the convention of [1] to denote a difference by $u' = u \oplus u^*$. Furthermore, we write $DES_k(x)$ for the encryption of the input x with DES [10] under the key k.

2.2 Iterated Hash Functions

Let H be an iterated hash function with compression function f. Then, H is a mapping $H : \{0,1\}^* \to \{0,1\}^n$ and $f : \{0,1\}^l \times \{0,1\}^n \to \{0,1\}^n$. The hash value computation can be described as follows:

$$\begin{aligned} h_0 &= IV \\ h_i &= f(m_i, h_{i-1}) \quad i = 1, \ldots, t \\ H(h_0; m) &= h_t \ , \end{aligned}$$

where h_i is an n-bit chaining variable and IV is a predefined n-bit initial value. After MD strengthening, *i.e.* fixing the IV and applying an unambiguous padding method including the binary representation of the message length (cf. [9]), the message m consists of t blocks, *i.e.* $m = m_1, \ldots, m_t$, where each block consists of l bits.

3 Iterated Hash Functions Possessing a b-Block Bypass

In this section, we introduce a new property of iterated hash functions and show which implications it has. For the remainder of this article, we assume without loss of generality that we have message lengths that are a multiple of the block length. Furthermore, we assume that the blocks required for MD strengthening have been removed.

Definition 1. *(b-Block Bypass) Let H be an iterated hash function. We say H possesses a* b-block bypass *if for any b-block message $m = m_1, \ldots, m_b$ there exists a b-block message $m^* = m_1^*, \ldots, m_b^* \neq m$ such that for any initial value h_0 the following holds:*

$$\begin{aligned} &H(h_0; m_1, \ldots, m_i) \neq H(h_0; m_1^*, \ldots, m_i^*) \quad \textit{for}\ i = 1, \ldots, b-1 \\ &H(h_0; m_1, \ldots, m_b) = H(h_0; m_1^*, \ldots, m_b^*) \end{aligned} \qquad (1)$$

An example of a 2-block bypass for a message $m = m_1, \ldots, m_t$ with $t \geq 2$ is shown in Figure 1.

Fact 1 *It follows directly from Definition 1 that if an iterated hash function possesses a b-block bypass then it is possible to construct a second preimage m^* for any given message $m = m_1, \ldots, m_t \neq m^*$ with $t \geq b$.*

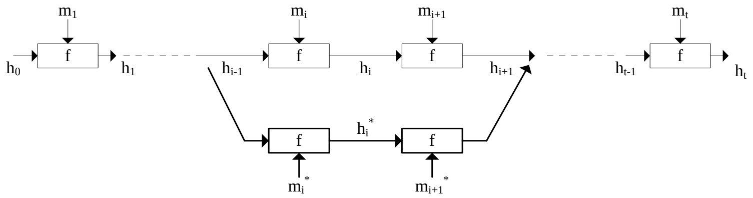


Fig. 1. An iterated hash function with a 2-block bypass for messages $m = m_1, \ldots, m_t$, with $t \geq 2$

Lemma 1. *Let H be an iterated hash function that possesses a b-block bypass. Then, for every message $m = m_1, \ldots, m_t$ with $t \geq b \geq 1$, we can construct*

$$\sum_{j=1}^{\lfloor t/b \rfloor} \binom{t - j(b-1)}{j} \tag{2}$$

distinct second preimages.

Proof. Based on Fact 1, we know that we can construct a second preimage for every message with block-length $\geq b$. From Definition 1 it follows immediately that it doesn't matter which b consecutive blocks of the message m are taken to construct a second preimage m^*. This implies that we have at least $t - b + 1$ second preimages for the message m (see Example 1).
On the other hand, if $\lfloor t/b \rfloor \geq 2$, we can apply the fact from Definition 1 not only for one b-block sub-message of m but for j sub-messages, where j can be in the range of $1, \ldots, \lfloor t/b \rfloor$. An illustration of this fact is also shown in Example 1. The problem of counting all these possible second preimages of m boils down to counting the number of possibilities of putting $t - jb$ indistinguishable balls into $j + 1$ distinguishable urns. This number is known to be

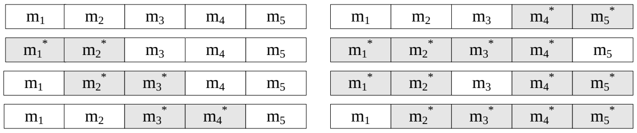
$$\binom{t - j(b-1)}{j},$$

cf. [4, page 38, Eq. (5.2)]. Summing over all $j = 1, \ldots, \lfloor t/b \rfloor$ proves (2). □

Example 1. Assume, we have an iterated hash function with a 2-block bypass. Thus, for every two blocks m_i, m_{i+1} of a message $m = m_1, \ldots, m_5$ there exist m_i^*, m_{i+1}^* with property (1). From formula (2), we know that we can construct 7 distinct second preimages for m. All these possibilities are depicted in Figure 2.

4 Instantiating DBLH with DESX

In this section, we describe the double block-length hash proposal of Hirose [5]. Then, we describe the block cipher DESX [7,8] and show different configurations of the hash scheme instantiated with DESX as underlying block cipher.



m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4^*	m_5^*
m_1^*	m_2^*	m_3	m_4	m_5	m_1^*	m_2^*	m_3^*	m_4^*	m_5
m_1	m_2^*	m_3^*	m_4	m_5	m_1^*	m_2^*	m_3	m_4^*	m_5^*
m_1	m_2	m_3^*	m_4^*	m_5	m_1	m_2^*	m_3^*	m_4^*	m_5^*

Fig. 2. For an iterated hash function that possesses a 2-block bypass, we can construct for any 5-block message $m = m_1, \ldots, m_5$ seven distinct second preimages. The dashed rectangles show which blocks of the original message m have been modified to construct the second preimage.

4.1 The Double Block-Length Hash Proposal DBLH

Shoichi Hirose proposed a double block-length hash function at FSE 2006 [5]. It is an iterated, block cipher based hash function. The compression function is defined as follows:

$$\begin{aligned} g_i &= e_{h_{i-1}\|m_i}(g_{i-1}) \oplus g_{i-1} \\ h_i &= e_{h_{i-1}\|m_i}(g_{i-1} \oplus c) \oplus g_{i-1} \oplus c, \end{aligned} \tag{3}$$

where c is an arbitrary constant ($c \neq 0$), and e_k ($k = h_{i-1}\|m_i$) is an arbitrary block cipher. The two blocks h_0, g_0 are two initial values. After t message blocks have been processed, the final hash value is the concatenation $h_t\|g_t$. As it can be seen in (3), the key length of the underlying block cipher e_k has to be greater than the block length. This is due to the fact that $|k| = |h_{i-1}| + |m_i|$, where $|h_{i-1}|$ is the block length of the cipher. In [5], Hirose proved the security of DBLH in the ideal cipher model. The security proof omits any reduction showing that if an adversary breaks the scheme he can distinguish the underlying block cipher from random.

4.2 DESX and the General FX-Construction

The block cipher DESX [8] was proposed by Rivest to protect DES against exhaustive key search attacks. Kilian and Rogaway proved the security of the DESX construction in [7,8].

For the description of DESX, we follow the notation of [8] except that we denote concatenation by '$\|$'. DESX is defined as follows:

$$DESX_{k\|k_1\|k_2}(x) = DES_k(x \oplus k_1) \oplus k_2 \ , \tag{4}$$

where $|k| = 56$, $|k_1| = 64$, and $|k_2| = 64$.

The more general construction is referred to as FX [7,8], where F can be any (k, n) block cipher with block length $|n|$ and key length $|k|$. The FX construction is defined as follows:

$$FX_{k\|k_1\|k_2}(x) = F_k(x \oplus k_1) \oplus k_2 \ , \tag{5}$$

where $|k_1| = |k_2| = |n|$.

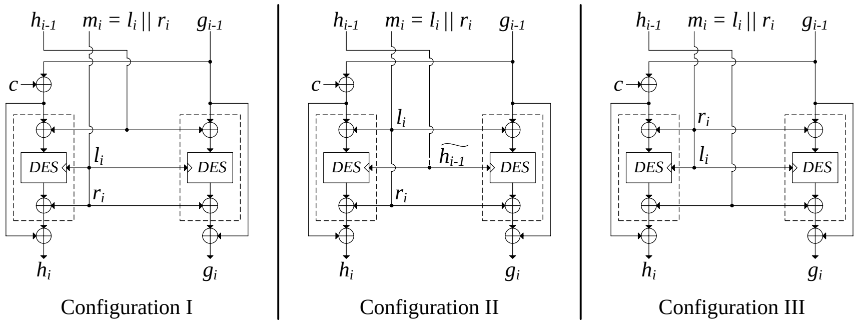


Fig. 3. Three possible configurations of DBLH with DESX as underlying block cipher. The hatch denotes the key input of DES.

4.3 DBLH with DESX

For DBLH with underlying block cipher $DESX_{k\|k_1\|k_2}(x)$, we can construct the following three configurations (see Figure 3), where $m_i = l_i\|r_i$.

$$\begin{aligned}
&\text{Configuration I:}\\
&k\|k_1\|k_2 = l_i\|h_{i-1}\|r_i,\ \text{where } |l_i| = 56, |h_{i-1}| = |r_i| = 64\\[1em]
&\text{Configuration II:}\\
&k\|k_1\|k_2 = \widetilde{h_{i-1}}\|l_i\|r_i,\ \text{where } |\widetilde{h_{i-1}}| = 56, |l_i| = |r_i| = 64\\[1em]
&\text{Configuration III:}\\
&k\|k_1\|k_2 = l_i\|r_i\|h_{i-1},\ \text{where } |l_i| = 56, |r_i| = |h_{i-1}| = 64
\end{aligned} \tag{6}$$

For each configuration, we can interchange l_i and r_i. However, without loss of generality, we take the configurations defined in (6) for the further analysis. For Configuration II, we have to truncate the 64-bit chaining variable h_{i-1} to 56 bits denoted by $\widetilde{h_{i-1}}$, since we need a 56-bit key k. Which bits are truncated does not have any impact on the analysis.

For the sake of simplicity, we will write DX to denote the instantiation of DBLH with DESX as underlying block cipher. If we speak of a specific configuration, we append the number of the configuration. For instance for DBLH with DESX in Configuration II, we write DX-II.

5 The Second Preimage Attack on DX

In this section, we present our second preimage attack on DX. For each of the configurations defined in (6) it is possible to construct second preimages since we can construct a b-block bypass as in Definition 1.

5.1 Second Preimages for DX-I Based on a 2-Block Bypass

We can construct second preimages for DX-I based on the following theorem.

Theorem 1. *The iterated hash function DX-I possesses a 2-block bypass, since for every two block message $m = m_1, m_2$ the following message m^* satisfies the conditions of Definition 1:*

$$m^* = m_1 \oplus (0\|u'), m_2 \oplus (0\|u') \ , \tag{7}$$

where $m_i = l_i\|r_i$, $|l_i| = 56$, $|r_i| = 64$, u' any value with $|u'| = 64$, and 0 is the 56-bit all-zero binary string.

Proof. Assume, we have the following 2-block messages m, m^*, where:

$$\begin{aligned} m &= m_1, m_2 = (l_1\|r_1), (l_2\|r_2) \\ m^* &= m_1^*, m_2^* = m_1 \oplus (0\|u'), m_2 \oplus (0\|u') = (l_1^*\|r_1^*), (l_2^*\|r_2^*) \\ l_1^* &= l_1 \oplus 0 = l_1, \quad r_1^* = r_1 \oplus u' \\ l_2^* &= l_2 \oplus 0 = l_2, \quad r_2^* = r_2 \oplus u' \end{aligned}$$

After one iteration, we have

$$\begin{aligned} g_1 &= g_0 \oplus DES_{l_1}(g_0 \oplus h_0) \oplus r_1 \\ g_1^* &= g_0 \oplus DES_{l_1}(g_0 \oplus h_0) \oplus r_1 \oplus u' = g_1 \oplus u' \ , \text{ and} \\ h_1 &= g_0 \oplus c \oplus DES_{l_1}(g_0 \oplus c \oplus h_0) \oplus r_1 \\ h_1^* &= g_0 \oplus c \oplus DES_{l_1}(g_0 \oplus c \oplus h_0) \oplus r_1 \oplus u' = h_1 \oplus u' \ . \end{aligned}$$

The outputs after two iterations are

$$\begin{aligned} g_2 &= g_1 \oplus DES_{l_2}(g_1 \oplus h_1) \oplus r_2 \\ g_2^* &= g_1 \oplus u' \oplus DES_{l_2}(g_1 \oplus u' \oplus h_1 \oplus u') \oplus r_2 \oplus u' \\ &= g_1 \oplus DES_{l_2}(g_1 \oplus h_1) \oplus r_2 = g_2 \ , \text{ and} \\ h_2 &= g_1 \oplus c \oplus DES_{l_2}(g_1 \oplus c \oplus h_1) \oplus r_2 \\ h_2^* &= g_1 \oplus u' \oplus c \oplus DES_{l_2}(g_1 \oplus u' \oplus c \oplus h_1 \oplus u') \oplus r_2 \oplus u' \\ &= g_1 \oplus c \oplus DES_{l_2}(g_1 \oplus c \oplus h_1) \oplus r_2 = h_2 \ . \end{aligned}$$

Hence, $g_2' = g_2 \oplus g_2^* = 0$ and $h_2' = h_2 \oplus h_2^* = 0$. Since the difference of the chaining variables $g_0' = h_0' = 0$, we have constructed a 2-block bypass for DX-I. □

Corollary 1 *For the iterated hash function DX-I and an arbitrary message $m = m_1, \ldots, m_t$ with $t \geq 2$, we can find at least*

$$\sum_{j=1}^{\lfloor t/2 \rfloor} \binom{t-j}{j}$$

second preimages.

Proof. This is an immediate consequence of Lemma 1 with $b = 2$ and Theorem 1. □

5.2 Second Preimages for DX-II, DX-III Based on a 3-Block Bypass

For Configuration II and Configuration III of DX, we can also construct second preimages. Different to the attack in Section 5.1, we will exploit the fact that we can construct a 3-block bypass.

Theorem 2. *The iterated hash function DX-II possesses a 3-block bypass, since for every 3-block message $m = m_1, m_2, m_3$ the following message m^* satisfies the conditions of Definition 1:*

$$m^* = m_1 \oplus (0 \| u'), m_2 \oplus (v' \| w'), m_3 \oplus (z' \| z') \ , \tag{8}$$

where $m_i = l_i \| r_i$, $|l_i| = |r_i| = 64$, u', v' any value with $|u'| = |v'| = 64$, and 0 is the 64-bit all-zero binary string. The difference $w' = u' \oplus t'$, where t' is the output difference of the left DES instance in iteration 2, namely

$$t' = \Big[DES_{\widetilde{h_1}}(g_1 \oplus c \oplus l_2) \Big] \oplus \Big[DES_{\widetilde{h_1 \oplus u'}}(g_1 \oplus u' \oplus c \oplus l_2 \oplus v') \Big] \, . \tag{9}$$

Once w' is determined, the difference z' can be computed as follows:

$$\begin{aligned} z' = \Big[& DES_{\widetilde{h_1}}(g_1 \oplus l_2) \oplus r_2 \oplus g_1 \Big] \oplus \\ & \Big[DES_{\widetilde{h_1 \oplus u'}}(g_1 \oplus u' \oplus l_2 \oplus v') \oplus r_2 \oplus w' \oplus g_1 \oplus u' \Big] \end{aligned} \tag{10}$$

Proof. We show that for the 3-block messages m and m^*, where

$$\begin{aligned} m &= m_1, m_2, m_3 = (l_1 \| r_1), (l_2 \| r_2), (l_3 \| r_3) \\ m^* &= m_1 \oplus (0 \| u'), m_2 \oplus (v' \| w'), m_3 \oplus (z' \| z') = (l_1^* \| r_1^*), (l_2^* \| r_2^*), (l_3^* \| r_3^*) \\ l_1^* &= l_1 \oplus 0, \quad r_1^* = r_1 \oplus u' \\ l_2^* &= l_2 \oplus v', \quad r_2^* = r_2 \oplus w' \\ l_3^* &= l_3 \oplus z', \quad r_3^* = r_3 \oplus z' \ , \end{aligned}$$

the output difference equals zero after three iterations. As described in Section 4.3, the chaining variable h_i is truncated since we need a 56-bit key. This is denoted by $\widetilde{h_i}$. After one iteration, we have

$$\begin{aligned} g_1 &= g_0 \oplus DES_{\widetilde{h_0}}(g_0 \oplus l_1) \oplus r_1 \\ g_1^* &= g_0 \oplus DES_{\widetilde{h_0}}(g_0 \oplus l_1) \oplus r_1 \oplus u' = g_1 \oplus u' \\ h_1 &= g_0 \oplus c \oplus DES_{\widetilde{h_0}}(g_0 \oplus c \oplus l_1) \oplus r_1 \\ h_1^* &= g_0 \oplus c \oplus DES_{\widetilde{h_0}}(g_0 \oplus c \oplus l_1) \oplus r_1 \oplus u' = h_1 \oplus u' \ . \end{aligned}$$

After two iterations, chaining variable h_2 is computed as follows

$$\begin{aligned} h_2 &= g_1 \oplus c \oplus DES_{\widetilde{h_1}}(g_1 \oplus c \oplus l_2) \oplus r_2 \\ h_2^* &= g_1 \oplus u' \oplus c \oplus DES_{\widetilde{h_1 \oplus u'}}(g_1 \oplus u' \oplus c \oplus l_2 \oplus v') \oplus r_2 \oplus w' \ . \end{aligned}$$

With $w' = u' \oplus t'$ and t' as defined in (9), we get

$$\begin{aligned} h_2^* &= g_1 \oplus u' \oplus c \oplus DES_{\widetilde{h_1 \oplus u'}}(g_1 \oplus u' \oplus c \oplus l_2 \oplus v') \oplus r_2 \oplus u' \\ &\oplus \underbrace{DES_{\widetilde{h_1}}(g_1 \oplus c \oplus l_2) \oplus DES_{\widetilde{h_1 \oplus u'}}(g_1 \oplus u' \oplus c \oplus l_2 \oplus v')}_{t'} \\ &= g_1 \oplus u' \oplus c \oplus r_2 \oplus u' \oplus DES_{\widetilde{h_1}}(g_1 \oplus c \oplus l_2) \\ &= h_2 \ . \end{aligned}$$

The difference in chaining variable g_2 after two iterations is

$$g_2^* = g_2 \oplus z' \ ,$$

where z' is defined in (10). After three iterations, we get

$$\begin{aligned} g_3 &= g_2 \oplus DES_{\widetilde{h_2}}(g_2 \oplus l_3) \oplus r_3 \\ g_3^* &= g_2 \oplus z' \oplus DES_{\widetilde{h_2}}(g_2 \oplus z' \oplus l_3 \oplus z') \oplus r_3 \oplus z' \\ &= g_2 \oplus DES_{\widetilde{h_2}}(g_2 \oplus l_3) \oplus r_3 \\ &= g_3 \\ h_3 &= g_2 \oplus c \oplus DES_{\widetilde{h_2}}(g_2 \oplus c \oplus l_3) \oplus r_3 \\ h_3^* &= g_2 \oplus z' \oplus c \oplus DES_{\widetilde{h_2}}(g_2 \oplus z' \oplus c \oplus l_3 \oplus z') \oplus r_3 \oplus z' \\ &= g_2 \oplus c \oplus DES_{\widetilde{h_2}}(g_2 \oplus c \oplus l_3) \oplus r_3 \\ &= h_3 \ . \end{aligned}$$

Therefore, after three iterations the differences in the chaining variables are $g_3' = g_3 \oplus g_3^* = 0$ and $h_3' = h_3 \oplus h_3^* = 0$. Since the difference of the chaining variables $g_0' = h_0' = 0$, we have constructed a 3-block bypass for DX-II. □

Theorem 3. *The iterated hash function DX-III possesses a 3-block bypass, since for every 3-block message $m = m_1, m_2, m_3$ the following message m^* satisfies the conditions of Definition 1:*

$$m^* = m_1 \oplus (u' \| v'), m_2 \oplus (0 \| z'), m_3 \oplus (0 \| (w' \oplus z')) \ , \qquad (11)$$

where $m_i = l_i \| r_i$, $|l_i| = 56$, $|r_i| = 64$, u', v' any value with $|u'| = 56$ and $|v'| = 64$, and 0 is the 56-bit all-zero binary string. Once the values u', v' have been chosen for the given input message block m_1, the differences w' and z' can be computed:

$$\begin{aligned} w' &= [g_0 \oplus c \oplus DES_{l_1}(g_0 \oplus c \oplus r_1) \oplus h_0] \\ &\oplus [g_0 \oplus c \oplus DES_{l_1 \oplus v'}(g_0 \oplus c \oplus r_1 \oplus u') \oplus h_0] \ , \\ z' &= [g_0 \oplus DES_{l_1}(g_0 \oplus r_1) \oplus h_0] \\ &\oplus [g_0 \oplus DES_{l_1 \oplus v'}(g_0 \oplus r_1 \oplus u') \oplus h_0] \end{aligned}$$

The proof of Theorem 3 works along the same lines as the proof of Theorem 1 and Theorem 2 and is given in Appendix A.

Corollary 2 *For the iterated hash function DX-II, respectively DX-III, and an arbitrary message* $m = m_1, \ldots, m_t$ *with* $t \geq 3$*, we can find at least*

$$\sum_{j=1}^{\lfloor t/3 \rfloor} \binom{t-2j}{j}$$

second preimages.

Proof. This is an immediate consequence of Lemma 1 with $b = 3$ and Theorem 2, respectively Theorem 3. □

6 Discussion

It is of common use to prove the security of certain cryptographic schemes based on the random-oracle model. In [2], John Black presented an overview on models for the security proof of different schemes. Additionally, he constructed a so-called uninstantiable block cipher based hash function which is provably secure in the ideal cipher model. Once the scheme is instantiated with a concrete block cipher it becomes insecure. This is referred to as an *uninstantiable scheme.* Our second preimage attack on DX does not imply that DBLH is an uninstantiable scheme. This is due to the fact that if DBLH is instantiated with for instance AES-192 or AES-256 [11], we did not find any attack. Furthermore, we have to point out that even if Kilian and Rogaway proved the security of the DESX construction in [7,8], they did not recommend DESX for the use in hash functions. As it has been shown in several papers DESX is vulnerable to related-key attacks [6,12]. This implies that DESX, or in general the FX construction, should not be used in a Davies-Meyer like hash function (cf. [6]). Indeed, there are some other schemes that become weak if DESX is used. For instance, the Preneel-Govaerts-Vandewalle scheme number 5 [13, Table 5.4, page 105], which is provably secure in the ideal cipher model (see Black *et al.* in [3]), is such a scheme. We can perform a similar second preimage attack as we did for DX. A sketch of the attack is given in Appendix B. As with DX, the input message for this construction is the key of the block cipher.

For the attack on DBLH with DESX or with any other FX construction, we did not exploit the related-key attack vulnerability but the fact that we gain additional structure within the hash scheme. This additional structure can be exploited to control the output differences for each iteration in such a way that we can construct a b-block bypass ($b = 2, 3$) as shown in Section 5. We have also shown that the attacker has a lot of freedom if an iterated hash function possesses a b-block bypass. In other words, for a given message the attacker can construct a huge amount of second preimages for the single given message.

Since the DBLH scheme has been proven in the ideal cipher model, the reader should not get the feeling that this proofs are worthless once the scheme is instantiated with a real-word block cipher. Even if there exist doubts about what one can expect from schemes that are proven in the ideal cipher model once they are instantiated (see for instance [2]), we point out that these proofs assume the ideal cipher model. It is clear that DESX does not fit into this model. Nevertheless, we used the instantiation of DBLH with DESX as underlying block cipher to show how second preimages can be constructed based on a b-block bypass for the hash function. We hope that researchers feel motivated to come up with more definitions about special properties of iterated hash functions.

7 Conclusion

In this article, we have shown that the underlying block cipher is important for the hash function proposal DBLH. We understand, that nobody would really instantiate DBLH with DESX as underlying block cipher but would rather use AES-192 or AES-256. However, we introduced a new property for iterated hash functions: a b-block bypass. We also showed how to construct them based on the DBLH construction with DESX. If an attacker can construct a b-block bypass then he has more degrees of freedom to construct second preimages. This makes the second preimage attack more powerful.

Currently, we are trying to find other hash function schemes that posses a b-block bypass. Furthermore, we are investigating the impact of the huge amount of second preimages an attacker can construct with our approach for a single given message.

References

1. Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. Journal of Cryptology, 4(1):3–72, 1991.
2. John Black. The Ideal-Cipher Model, Revisited: An Uninstantiable Blockcipher-Based Hash Function. In Matt Robshaw, editor, *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Pre-Proceedings*, 2006.
3. John Black, Phillip Rogaway, and Thomas Shrimpton. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *LNCS*, pages 320–335. Springer, 2002.
4. William Feller. *An introduction to probability theory and its applications. Vol. I.* Third edition. John Wiley & Sons Inc., New York, 1968.
5. Shoichi Hirose. Some Plausible Constructions of Double-Block-Length Hash Functions. In Matt Robshaw, editor, *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Pre-Proceedings*, 2006.
6. John Kelsey, Bruce Schneier, and David Wagner. Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In Yongfei Han, Tatsuaki Okamoto, and Sihan Qing, editors, *Information and Communication Security, First International Conference, ICICS'97, Beijing, China, November 11-14, 1997, Proceedings*, volume 1334 of *Lecture Notes in Computer Science*, pages 233–246. Springer, 1997.

7. Joe Kilian and Phillip Rogaway. How to Protect DES Against Exhaustive Key Search. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109, pages 252–267. Springer, 1996.
8. Joe Kilian and Phillip Rogaway. How to Protect DES Against Exhaustive Key Search (an Analysis of DESX). *J. Cryptology*, 14(1):17–35, 2001.
9. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boston, 2001.
10. National Institute of Standards and Technology (NIST). FIPS-46-3: Data Encryption Standard, October 1999. Available online at `http://www.itl.nist.gov/fipspubs/`.
11. National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard, November 2001. Available online at `http://www.itl.nist.gov/fipspubs/`.
12. Raphael Chung-Wei Phan. Related-Key Attacks on Triple-DES and DESX Variants. In Tatsuaki Okamoto, editor, *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, volume 2964 of *Lecture Notes in Computer Science*, pages 15–24. Springer, 2004.
13. Bart Preneel. *Analysis and Design of Cryptographic Hash Functions*. PhD thesis, Katholieke Universiteit Leuven, 1993.

A Proof of Theorem 3

In this section, we prove Theorem 3 given in Section 5.2. As for the proof of Theorem 1 and Theorem 2, we show that for the 3-block messages m and m^*, where $m = m_1, m_2, m_3 = (l_1\|r_1), (l_2\|r_2), (l_3\|r_3)$ and

$$\begin{aligned} m^* &= m_1 \oplus (u'\|v'), m_2 \oplus (0\|z'), m_3 \oplus (0\|(w' \oplus z')) = (l_1^*\|r_1^*), (l_2^*\|r_2^*), (l_3^*\|r_3^*) \\ l_1^* &= l_1 \oplus u', \quad r_1^* = r_1 \oplus v' \\ l_2^* &= l_2 \oplus 0, \quad r_2^* = r_2 \oplus z' \\ l_3^* &= l_3 \oplus 0, \quad r_3^* = r_3 \oplus (w' \oplus z') \ , \end{aligned}$$

the output difference equals zero after three iterations, *i.e.* $g_3' = h_3' = 0$. After the first iteration, we have

$$\begin{aligned} g_1 &= g_0 \oplus DES_{l_1}(g_0 \oplus r_1) \oplus h_0 \\ g_1^* &= g_1 \oplus z', \text{where} \\ z' &= [g_0 \oplus DES_{l_1}(g_0 \oplus r_1) \oplus h_0] \\ &\quad \oplus [g_0 \oplus DES_{l_1 \oplus v'}(g_0 \oplus r_1 \oplus u') \oplus h_0] \ \text{, and} \\ h_1 &= g_0 \oplus c \oplus DES_{l_1}(g_0 \oplus c \oplus r_1) \oplus h_0 \\ h_1^* &= h_1 \oplus w', \text{where} \\ w' &= [g_0 \oplus c \oplus DES_{l_1}(g_0 \oplus c \oplus r_1) \oplus h_0] \\ &\quad \oplus [g_0 \oplus c \oplus DES_{l_1 \oplus v'}(g_0 \oplus c \oplus r_1 \oplus u') \oplus h_0] \ . \end{aligned}$$

The difference of the chaining variables after two iterations is

$$\begin{aligned}
g_2 &= g_1 \oplus DES_{l_2}(g_1 \oplus r_2) \oplus h_1 \\
g_2^* &= g_1 \oplus z' \oplus DES_{l_2}(g_1 \oplus z' \oplus r_2 \oplus z') \oplus h_1 \oplus w' \\
&= g_2 \oplus (w' \oplus z') \text{ , and} \\
h_2 &= g_1 \oplus c \oplus DES_{l_2}(g_1 \oplus c \oplus r_2) \oplus h_1 \\
h_2^* &= g_1 \oplus z' \oplus c \oplus DES_{l_2}(g_1 \oplus z' \oplus c \oplus r_2 \oplus z') \oplus h_1 \oplus w' \\
&= h_2 \oplus (w' \oplus z') \ .
\end{aligned}$$

The output difference after three iterations is computed as follows. For the sake of clearness, we write $y' = w' \oplus z'$:

$$\begin{aligned}
g_3 &= g_2 \oplus DES_{l_3}(g_2 \oplus r_3) \oplus h_2 \\
g_3^* &= g_2 \oplus y' \oplus DES_{l_3}(g_3 \oplus y' \oplus r_3 \oplus y') \oplus h_2 \oplus y' \\
&= g_3 \text{ , and} \\
h_3 &= g_2 \oplus c \oplus DES_{l_3}(g_2 \oplus c \oplus r_3) \oplus h_2 \\
h_3^* &= g_2 \oplus y' \oplus c \oplus DES_{l_3}(g_2 \oplus y' \oplus c \oplus r_3 \oplus y') \oplus h_2 \oplus y' \\
&= h_3
\end{aligned}$$

Hence, $g'_3 = g_3 \oplus g_3^* = 0$ and $h'_3 = h_3 \oplus h_3^* = 0$. Since the difference of the chaining variables $g'_0 = h'_0 = 0$, we have constructed a 3-block bypass for DX-III. □

B Second Preimage Attack on PGV Scheme Number 5 with DESX

In Figure 4, the PGV scheme number 5 (see [13, Table 5.4, page 105]) is schematically depicted on the left hand side. The scheme instantiated with DESX is shown on the right hand side.

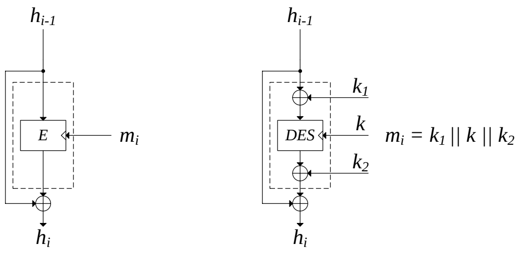


Fig. 4. PGV scheme number 5 with block cipher E (left) and instantiated with DESX (right)

It is easy to verify that for the two 1-block messages m_i and m_i^*, where $m_i = k_1 \| k \| k_2$ and $m_i^* = k_1 \oplus u' \| k \oplus v' \| k_2 \oplus w'$, with

$$w' = [DES_k(h_{i-1} \oplus k_1)] \oplus [DES_{k \oplus v'}(h_{i-1} \oplus k_1 \oplus u')]$$

and u', v' any value, the difference in the chaining variables equals zero, *i.e.* $h_i' = h_i \oplus h_i^* = 0$ and we have thus constructed a 1-block bypass.