A Simple Related-Key Attack on the Full SHACAL-1

Eli Biham[1]⋆, Orr Dunkelman[1,2]∗, Nathan Keller⋆⋆[3]

[1]Computer Science Department, Technion.
Haifa 32000, Israel
{biham,orrd}@cs.technion.ac.il
[2]Katholieke Universiteit Leuven, ESAT/SCD-COSIC
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
orr.dunkelman@esat.kuleuven.be
[3]Einstein Institute of Mathematics, Hebrew University.
Jerusalem 91904, Israel
nkeller@math.huji.ac.il

Abstract. SHACAL-1 is a 160-bit block cipher with variable key length of up to 512-bit key based on the hash function SHA-1. It was submitted to the NESSIE project and was accepted as a finalist for the 2nd phase of evaluation. Since its introduction, SHACAL-1 withstood extensive cryptanalytic efforts. The best known key recovery attack on the full cipher up to this paper has a time complexity of about 2^{420} encryptions. In this paper we use an observation due to Saarinen to present an elegant related-key attack on SHACAL-1. The attack can be mounted using two to eight unknown related keys, where each additional key reduces the time complexity of retrieving the actual values of the keys by a factor of 2^{62}. When all eight related-keys are used, the attack requires $2^{101.3}$ related-key chosen plaintexts and has a running time of $2^{101.3}$ encryptions. This is the first successful related-key key recovery attack on a cipher with varying round constants.

1 Introduction

In 1993, NIST has issued a standard hash function called Secure Hash Algorithm (FIPS-180) [27]. Later this version was named SHA-0, as NIST published a small tweak to this standard called SHA-1 in 1995. Both SHA-0 and SHA-1 are Merkle-Damgard hash functions with compression function that accept blocks of 512 bits and chaining values of 160 bits (which is also the digest size). Later, NIST has published three more standard hash functions as part of FIPS-180: SHA-256, SHA-384 and SHA-512. Each of the new hash functions has a digest size corresponding to its number, i.e., SHA-256 has a 256-bit digest, etc. Recently, NIST has issued another hash function, SHA-224, that has a digest size of 224 bits.

⋆ This work was supported in part by the Israel MOD Research and Technology Unit.
⋆⋆ The author was supported by the Adams fellowship.

Both SHA-0 and SHA-1 were subjected to a great deal of analysis [2, 3, 12, 30, 32, 34]. In the last two years there was a major progress in the attacks on both of the hash functions. This progress included finding a collision in SHA-0, and devising an algorithm that can find a collision in SHA-1 in less than 2^{63} SHA-1 applications [2, 3, 30, 32, 34]. The new techniques are based on finding good differentials of the compression function of SHA-1 and combining them with some novel plaintext modification techniques.

In 2000 it was suggested to use the compression function of SHA-1 as a block cipher [15]. Later this suggestion was named SHACAL-1 and submitted to the NESSIE project [16]. SHACAL-1 is a 160-bit block cipher with a variable key length (0–512 bits) and 80 rounds based on the compression function of SHA-1. The cipher was selected as a NESSIE finalist, but was not selected for the NESSIE portfolio [25].

Several papers analyze the strength of SHACAL-1 as a block cipher [6, 17, 20, 21]. These papers apply differential, amplified boomerang, rectangle and related-key rectangle attacks to reduced-round variants of SHACAL-1. The best known attack on SHACAL-1 that does not use related-keys is a rectangle attack on 49-round SHACAL-1 [6].

In a recent paper a transformation of the collision-producing differentials of SHA-1 presented in [32] was used to devise the first known attack on the full SHACAL-1 [13]. The attack is a related-key rectangle attack that requires $2^{159.8}$ chosen plaintexts encrypted under four related keys and has a time complexity of 2^{423} encryptions.

In [23], Saarinen observed that it is possible to construct slid pairs in the compression function of SHA-1 using about 2^{97} chosen chaining values (for two different blocks of message). Saarinen used the slid pairs to mount a related-key distinguishing attack against SHACAL-1 requiring 2^{97} chosen plaintexts encrypted under two related keys and time complexity of 2^{97} encryptions.

In this paper we use the results of [23] to devise key-recovery attacks against the full SHACAL-1 with much lower data and time complexities than previously known. The attacks use between two and eight unknown related keys, where each additional key reduces the time complexity of retrieving the actual values of the keys by a factor of 2^{62}. When all eight related-keys are used, the attack requires $2^{101.3}$ related-key chosen plaintexts and has a running time of $2^{101.3}$ encryptions. A comparison of the known attacks on SHACAL-1 along with our new results is presented in Table 1.

This is the first time a related-key attack succeeds against a cipher with varying round constants. Moreover, this is the first case, where the related-key process is used with some probability without combining it with other attacks, e.g., related-key differentials [19] or related-key rectangle attack [7, 20, 17].

This paper is organized as follows: In Section 2 we describe the block cipher SHACAL-1. In Section 3 we describe the previously known results on SHACAL-1. We shortly describe Saarinen's main observation on SHA-1 in Section 4. In Section 5 we present our new related-key attack on SHACAL-1. We summarize the paper in Section 6.

Attack & Source	Number of		Rounds	Complexity	
	Keys	Rounds		Data	Time
Differential [21]	1	41	0–40	2^{141} CP	2^{491}
Amplified Boomerang [21]	1	47	0–46	$2^{158.5}$ CP	$2^{508.4}$
Rectangle [6]	1	47	0–46	$2^{151.9}$ CP	$2^{482.6}$
Rectangle [6]	1	49	29–77	$2^{151.9}$ CC	$2^{508.5}$
Related-Key Rectangle [20]	2	59	0–58	$2^{149.7}$ RK-CP	$2^{498.3}$
Related-Key Rectangle [17]	4	70	0–69	$2^{151.8}$ RK-CP	$2^{500.1}$
Related-Key Rectangle [13]	4	80	0–79	$2^{159.8}$ RK-CP	$2^{423.0}$
Related-Key Rectangle [13]	4	80	0–79	$2^{153.8}$ RK-CP	$2^{504.2}$
Slide † [23]	2	80	0–79	2^{97} RK-CP	2^{97}
Related Key (Section 5)	2	80	0–79	2^{97} RK-CP	2^{447}
Related Key (Section 5)	4	80	0–79	$2^{99.6}$ RK-CP	2^{321}
Related Key (Section 5)	8	80	0–79	$2^{101.3}$ RK-CP	$2^{101.3}$

Complexity is measured in encryption units.
† – Distinguishing attack
CP — Chosen Plaintexts, CC — Chosen Ciphertexts, RK — Related-Key

Table 1. Summary of Our Results and Previously Known Results on SHACAL-1.

2 Description of SHACAL-1

SHACAL-1 [16] is a 160-bit block cipher supporting variable key lengths (0–512 bits). It is based on the compression function of the hash function SHA-1 [27]. The cipher has 80 rounds (also referred as steps) grouped into four types of 20 rounds each.[1]

The 160-bit plaintext is divided into five 32-bit words – A, B, C, D and E. We denote by X_i the value of word X before the ith round, i.e., the plaintext P is divided into A_0, B_0, C_0, D_0 and E_0, and the ciphertext is composed of $A_{80}, B_{80}, C_{80}, D_{80}$ and E_{80}.

In each round the words are updated according to the following rule:

$$\begin{aligned}
A_{i+1} &= W_i + ROTL_5(A_i) + f_i(B_i, C_i, D_i) + E_i + K_i \\
B_{i+1} &= A_i \\
C_{i+1} &= ROTL_{30}(B_i) \\
D_{i+1} &= C_i \\
E_{i+1} &= D_i
\end{aligned}$$

where $+$ denotes addition modulo 2^{32}, $ROTL_j(X)$ represents rotation to the left by j bits, W_i is the round subkey, and K_i is the round constant.[2] There are

[1] To avoid confusion, we adopt the common notations for rounds. In [16] the notation step stands for round, where round is used for a group of 20 steps.

[2] This time we adopt the notations of [16], and alert the reader of the somewhat confusing notations.

three different functions f_i, selected according to the round number:

$$\begin{array}{llr} f_i(X,Y,Z) = f_{if} = & (X\&Y)|(\neg X\&Z) & 0 \le i \le 19 \\ f_i(X,Y,Z) = f_{xor} = & (X \oplus Y \oplus Z) & 20 \le i \le 39, 60 \le i \le 79 \\ f_i(X,Y,Z) = f_{maj} = & ((X\&Y)|(X\&Z)|(Y\&Z)) & 40 \le i \le 59 \end{array}$$

There are also four round constants:

Rounds	K_i	Rounds	K_i
0–19	$5A827999_x$	20–39	$6ED9EBA1_x$
40–59	$8F1BBCDC_x$	60–79	$CA62C1D6_x$

In [16] it is strongly advised to use keys of at least 128 bits, even though shorter keys are supported. The first step in the key schedule algorithm is to pad the supplied key into a 512-bit key. Then, the 512-bit key is expanded into eighty 32-bit subkeys (or a total of 2560 bits of subkey material). The expansion is done in a linear manner using a linear feedback shift register (over $GF(2^{32})$).

The key schedule is as follows: Let $M_0, \dots, M_{15}$ be the 16 key words (32 bits each). Then the round subkeys $W_0, \dots, W_{79}$ are computed by the following algorithm:

$$W_i = \begin{cases} M_i & 0 \le i \le 15 \\ ROTL_1(W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16}) & 16 \le i \le 79 \end{cases}$$

3 Previous Results

A preliminary differential and linear analysis of the properties of the compression function of SHA-1 as a block cipher is presented in [15]. The found differentials are relatively short (10 rounds) and have probabilities varying between 2^{-13} and 2^{-26} (depending on the round functions).

In [28] these differentials are improved, and 20-round differentials with probability 2^{-41} are presented. In [21] another set of differentials of SHACAL-1 is presented, including a 30-round differential with probability 2^{-130}.

In [21] a 21-round differential for rounds 0–20 and a 15-round differential for rounds 21–35 are combined to devise an amplified boomerang distinguisher [18] for 36-round SHACAL-1. This distinguisher is used to attack 39-round SHACAL-1 using $2^{158.5}$ chosen plaintexts and about $2^{250.8}$ 39-round SHACAL-1 encryptions. The attack is based on guessing the subkeys of the three additional rounds, and then checking whether the distinguisher succeeds. This approach is further extended to attack 47-round SHACAL-1 before exhaustive key search becomes faster than this attack. Another attack presented in [21] is a differential attack on 41-round SHACAL-1. The success of these attacks was questioned and resolved in [6].

Besides resolving the problems with previous attacks, in [6] a rectangle attack on 49-round SHACAL-1 is presented. The attack requires $2^{151.9}$ chosen plaintexts, and has a running time equivalent to $2^{508.5}$ 49-round SHACAL-1 encryptions.

In [20] a related-key rectangle attack with two keys is presented against 59-round SHACAL-1. This attack has a data complexity of $2^{149.7}$ related-key chosen plaintexts and a time complexity of $2^{498.3}$ 59-round SHACAL-1 encryptions. This attack is improved in [17] to a related-key rectangle attack with four keys on 70-round SHACAL-1. The improved attack has a data complexity of $2^{151.8}$ related-key chosen plaintexts, and a time complexity of $2^{500.1}$ 70-round SHACAL-1 encryptions.

Using the improved differentials of SHA-1 found in [32] and some improved key recovery techniques, two related-key rectangle attacks with four keys on the full SHACAL-1 are given in [13]. The first has a data complexity of $2^{159.8}$ related-key chosen plaintexts and a time complexity of $2^{423.0}$ encryptions, and the second has a data complexity of $2^{153.8}$ related-key chosen plaintexts and a time complexity of $2^{504.2}$ encryptions.

Summarizing the known attacks on SHACAL-1, the best known attacks against SHACAL-1 use the rectangle technique. The best attack in the related-key model is applicable against the full SHACAL-1, while the best chosen plaintext attack is applicable for a 49-round reduced variant of the cipher. Both of the attacks require a very large amount of chosen plaintexts and a very high time complexity.

4 Slid Pairs in the Compression Function of SHA-1

4.1 Related-Key Attacks and Slid Pairs

Related key attacks [1, 22] are attacks that exploit relations during encryption under different keys. Let us consider an iterated block cipher whose key schedule is simple enough such that for any key K_1, it is possible to find K_2 such that the round subkeys KR_i^1, KR_i^2 produced by K_1 and K_2, respectively, satisfy:

$$KR_i^1 = KR_{i+1}^2$$

Assume that like in many ciphers, all the rounds of the cipher are the same. For such pair of keys, if a pair of plaintexts (P_1, P_2) satisfies $P_1 = f_{KR_1^2}(P_2)$, where $f_{sk}(P)$ denotes one round encryption of P under the subkey sk, then the corresponding ciphertexts C_1 and C_2 satisfy $C_1 = f_{KR_r^1}(C_2)$, where r is the number of rounds. Given such a pair of plaintexts for these related keys, it is possible to find the keys using a very simple attack algorithm.

In [10] Biryukov and Wagner show that the related key attacks can be applied to ciphers that can be written as $E_k = f_k^l = f_k \circ f_k \circ \ldots \circ f_k$, where f_k is a "relatively simple" key-dependent function. The attack looks for two plaintexts P_1 and P_2 satisfying the relation $P_2 = f_k(P_1)$. Such a pair is called a *slid pair*, and can be used in an attack that is similar to the attack in the case of related keys.

In the slide attack, the attacker collects $2^{n/2}$ known plaintext/ciphertext pairs (where n is the block size). For every pair of plaintexts (P_1, P_2), the attacker checks whether it is a slid pair by treating the pair as a slid pair and trying to use it to attack f_k. The time complexity of the attack is 2^n applications of the attack on f_k given a slid pair. Note that the data and time complexities are independent of the number of rounds in E_k. The slide attack was further generalized in [8, 11, 14] to be applicable to a wider range of block ciphers.

4.2 Saarinen's Observation

Saarinen has observed that slid pairs can be found (with some probability) in the compression function of SHA-1, i.e., in SHACAL-1 [23]. The slid pairs are constructed under two related message blocks, or in the case of SHACAL-1, two related keys.

Let $W = (W_0, W_1, \ldots, W_{15})$ be the first key, and let $W^* = (W_0^*, W_1^*, \ldots, W_{15}^*)$, such that

$$W_i^* = \begin{cases} W_{i+1} & 0 \le i \le 14 \\ W_{16} = ROTL_1(W_{13} \oplus W_8 \oplus W_2 \oplus W_0) & i = 15 \end{cases}$$

These two keys satisfy $W_i^* = W_{i+1}$ for $0 \le i \le 78$.

Let $P = (A_0, B_0, C_0, D_0, E_0)$ and $P^* = (A_0^*, B_0^*, C_0^*, D_0^*, E_0^*)$ be two plaintexts encrypted under W and W^*, respectively. We denote the input to round i by $(A_i, B_i, C_i, D_i, E_i)$ (or with * when considering the encryption of P^* under W^*). If after the first round of the encryption of P under W the following holds:

$$A_1 = A_0^*; \quad B_1 = B_0^*; \quad C_1 = C_0^*; \quad D_1 = D_0^*; \quad E_1 = E_0^* \tag{1}$$

then this equality holds until round 20 of the encryption of P (or round 19 of the encryption of P^*). In order for the slid pair to retain its "slidness", the following equality must hold:

$$W_{20} + ROTL_5(A_{20}) + f_{20}(B_{20}, C_{20}, D_{20}) + E_{20} + K_{20} =$$

$$A_{21} \quad = \quad A_{20}^* =$$

$$W_{19}^* + ROTL_5(A_{19}^*) + f_{19}(B_{19}^*, C_{19}^*, D_{19}^*) + E_{19}^* + K_{19}$$

As $W_{20} = W_{19}^*$, $A_{20} = A_{19}^*$, $B_{20} = B_{19}^*$, $C_{20} = C_{19}^*$, $D_{20} = D_{19}^*$, $E_{20} = E_{19}^*$, the above holds whenever

$$f_{xor}(B_{20}, C_{20}, D_{20}) + K_{20} = f_{if}(B_{20}, C_{20}, D_{20}) + K_{19}. \tag{2}$$

For a random permutation, this equality holds with probability 2^{-32}. In the case of SHACAL-1, it was verified experimentally that the probability is close to 2^{-32} (see [10]).

If the transition between the IF rounds and the XOR rounds is successful, then the equality remains until round 40 of the encryption of P. Again, with probability close to 2^{-32} the transition in round 40 does not affect the equality between the respective intermediate encryption values. The same holds for the last transition in round 60.

Thus, Saarinen concluded that the probability that (P, P^*) is a slid pair assuming that it satisfies the condition of Equation (1) is 2^{-96}. To achieve such pairs, pairs of structures of 2^{32} chosen plaintexts are chosen, such that $SetP = (A, B, C, D, x)$ for some fixed A, B, C, D and all possible values of x, and $SetP^* = (y, A, ROTL_{30}(B), C, D)$ for all possible values of y. These structures ensure that for each $P \in SetP$ there exists $P^* \in SetP^*$ such that the pair (P, P^*) satisfies the condition in Equation (1). Hence, 2^{64} pairs of structures are expected to contain a slid pair with a relatively high probability.

In order to detect the slid pairs, we note that for a slid pair we have

$$A_{80} = A^*_{79}; \quad B_{80} = B^*_{79}; \quad C_{80} = C^*_{79}; \quad D_{80} = D^*_{79}; \quad E_{80} = E^*_{79}, \qquad (3)$$

and this gives a 128-bit filtering on the ciphertexts that can be easily executed using a hash table. However, since we check a total number of 2^{128} pairs and our filtering is only on 128 bits, we expect that for a random permutation, one pair can also pass the filtering. In order to overcome this problem, we take 2^{65} pairs of structures, thus expecting to detect two slid pairs. Then we check whether the pairs suggest the same value for the subkey of the last round. If not, we collect some additional structures, find another slid pair and check again. With a high probability, the right subkey will be suggested at least twice, while for a random permutation the probability that a subkey is suggested twice is extremely low.

Therefore, the attack can distinguish between SHACAL-1 and a random permutation using about 2^{98} chosen plaintexts encrypted under two related keys, with time complexity of about 2^{98} encryptions.

We note that Saarinen has also proposed an algorithm that requires only 2^{32} operations to find such a slid pair in SHA-1. However, the algorithm assumes that the attacker can control the message block (i.e., the keys) directly, and thus it is not applicable to SHACAL-1 (unless it is in a chosen-key attack).

5 A Simple Related-Key Attack on SHACAL-1

The basic stage of our attack on SHACAL-1 is similar to the distinguishing attack presented by Saarinen [23]. We encrypt some pairs of structures under the two related keys and detect the candidate slid pairs. Now, each candidate slid pair suggests a value for two key words – the subkey used in the last round of the encryption under the key W^* and the subkey used in the first round of the encryption under W. At this stage, there is a difference between our attack and the attack in [23]: In order to reduce the data complexity of the attack we do not wait until the same key word is suggested twice, but rather continue with all the suggested subkey values. The wrong key values can be easily discarded in the last stage of our attack that will be described later, and hence we only need that the right key value will be amongst the suggested ones.

The algorithm of the basic stage of the attack is as follows:

1. Repeat the following M times, when M will be specified later:
   - Choose A, B, C, and D at random and ask for the encryption of $SetP = (A, B, C, D, x)$ for all possible values of x under W and of $SetP^* = (y, A, ROTL_{30}(B), C, D)$ for all possible values of y under W^*.
   - Search for candidate slid pairs, i.e., pairs of ciphertexts $T = (a, b, c, d, e) \in SetP$ and $T^* = (a^*, b^*, c^*, d^*, e^*) \in SetP^*$ such that $b^* = a, c^* = ROTL_{30}(b), d^* = c$, and $e^* = d$. Pass to the next stage all the candidate slid pairs, and the respective plaintext pairs denoted by $P = (A, B, C, D, X)$ and $P^* = (Y^*, A, ROTL_{30}(B), C, D)$, respectively.
2. For each candidate slid pair, compute W_0 and W_{80} using the formulas:

$$W_0 = Y^* - [ROTL_5(A) + f_{if}(B, C, D) + X + K_0] \tag{4}$$

$$W_{80} = a^* - [ROTL_5(a) + f_{xor}(b, c, d) + e + K_{79}] \tag{5}$$

3. Output all the pairs of values (W_0, W_{80}) suggested by candidate pairs.

After using the pair of related keys (W, W^*), we can repeat the attack with the related keys (W^*, W^{**}) that satisfy the relation

$$W_i^{**} = \begin{cases} W_{i+1}^* & 0 \le i \le 14 \\ W_{16}^* = ROTL_1(W_{13}^* \oplus W_8^* \oplus W_2^* \oplus W_0^*) & i = 15 \end{cases}$$

to retrieve two additional key words. This time the attack retrieves W_0^* and W_{80}^*, but due to the relation between W and W^*, these values are equal to W_1 and W_{81}, respectively. Then, the attack can be applied again. As all the keys are linearly dependent of each other, it is possible to combine the knowledge of each instance of the attack into a knowledge on the original key W. Therefore, if we repeat the attack 7 times, thus requiring 8 related keys, we get $64 \cdot 7 = 448$ linear equations in the bits of the key, and the rest of the key can be found by exhaustive search with time complexity of only 2^{64} encryptions. Note that if several candidates for the subkey values were suggested in some of the basic attacks, we perform the last stage of the attack for all the possible candidates, and still the time complexity is much below 2^{100} encryptions.

Now we want to compute the minimal possible value of M such that, with a high probability, in all the 7 applications of the basic stage of the attack the right key will be amongst the suggested ones. Using a Poisson approximation we get that if $M = 2^{64} \cdot t$ then the probability that in a single application of the basic attack no real slid pair will be found is e^{-t}. Hence, assuming that the basic stages are independent we get that the probability that in all the applications there will be at least one real slid pair is $(1 - e^{-t})^7$. For $t = 2^{1.5}$, we get $(1 - e^{-t})^7 \approx 0.65$, and hence the success probability of the attack is about 0.65. If we want a greater success probability, we can increase the data complexity. For example, for $t = 4$ the success probability is about 0.88 and for $t = 8$, the success probability is greater than 0.99.

Therefore, the total data complexity of the attack is $2^{1.5} \cdot 2^{64} \cdot 2^{33} \cdot 7 \approx 2^{101.3}$ related-key chosen plaintexts, and the time complexity is dominated solely by the encryption time.

The data complexity can be slightly reduced using an adaptive attack. We can ask for structures of plaintexts to be encrypted under two related keys, until two candidate slid pairs suggest the same subkeys (thus, with high probability, these are the right subkeys). Once this happens, there is no need to further encrypt plaintexts under these two related keys.

If only a smaller number of $k < 8$ related keys is available, we can perform the basic stage of the attack $k-1$ times and find the rest of the key bits using exhaustive key search. We note that it is possible to reduce the time complexity of this search by a factor of four, by considering the fact that for slid pairs there is a special relation between the intermediate encryption values in round 20 (presented in Equation (2)). If this relation is not satisfied, the key can be easily discarded without completing the full encryption.

Note that if the attack is performed less than seven times, we can reduce the number of chosen plaintexts and still expect that with a high probability, in all the applications of the basic attack there will be at least one slid pair. For example, for $k = 2$ we can take $t = 1$ and get success probability of 0.63, and for $k = 4$ we can take $t = 2$ and get success probability of 0.65.

The exact time complexity of the last stage depends on the number of subkey candidates suggested in the basic stages. Assuming that in each stage, besides the right subkey we encounter three wrong candidate values (that cannot be discarded) at most, the total time complexity of the attack is at most $2^{510+2(k-1)-64(k-1)} = 2^{510-62(k-1)}$ trial encryptions.

We conclude that our attack with k related keys has a data complexity of at most $(k-1)\cdot 2^{98.5}$ related-key chosen plaintexts, and a running time of at most $\max\{(k-1)\cdot 2^{98.5}, 2^{510-62(k-1)}\}$ trial encryptions.

6 Summary and Conclusions

In this paper we presented a simple related-key attack on SHACAL-1. The attack can be performed using between two and eight related-keys. The variant of the attack with eight related keys requires $2^{101.3}$ chosen plaintexts and has time complexity of $2^{101.3}$ encryptions. The attack is by far better than all the previously known related-key attacks on SHACAL-1.

Our results, following the results in [23], are based on the original variant of the *related-key attack* presented by Biham in 1993 [1].[3] Usually, slid pairs or their equivalent related-key counterparts can be found only if the round functions of the cipher are identical (for all the rounds). Hence, inserting a round constant to the round function of a block cipher seems to be sufficient to protect it with respect to related-key attacks.

[3] In 1996, Kelsey et al.[19] presented the *related-key differential attack*. This attack uses different ideas and the only similarity between it and Biham's attack is the fact that both attacks perform in the related-key model. The related-key rectangle technique, that was used in previous attacks on SHACAL-1, is an expansion of the related-key differential attack.

In SHACAL-1, round constants are used in all the round functions but the fact that the constants are changed only once every 20 rounds and the fact that the round functions are also slightly changed in the same places can be used to construct the required pairs of plaintexts. Hence, the results of [23] and our attack show that related-key attacks can be mounted also on ciphers using round constants, if the constants are not chosen carefully.

We conclude that our key-recovery attack demonstrates, once again, that using a linear key schedule algorithm and relatively similar round functions is not a good way to design a secure block cipher.

References

1. Eli Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, Journal of Cryptology, Vol. 7, No. 4, pp. 229–246, Springer-Verlag, 1994.
2. Eli Biham, Rafi Chen, *Near-Collisions of SHA-0*, Advances in Cryptology, proceedings of CRYPTO 2004, Lecture Notes in Computer Science 3152, pp. 290–305, Springer-Verlag, 2004.
3. Eli Biham, Rafi Chen, Antoine Joux, Patrick Carribault, Christophe Lemuet, William Jalby, *Collisions of SHA-0 and Reduced SHA-1*, Advances in Cryptology, proceedings of EUROCRYPT 2005, Lecture Notes in Computer Science 3621, pp. 36–57, 2005.
4. Eli Biham, Orr Dunkelman, Nathan Keller, *The Rectangle Attack – Rectangling the Serpent*, Advances in Cryptology, proceedings of EUROCRYPT '01, Lecture Notes in Computer Science 2045, pp. 340–357, Springer-Verlag, 2001.
5. Eli Biham, Orr Dunkelman, Nathan Keller, *New Results on Boomerang and Rectangle Attacks*, proceedings of Fast Software Encryption 9, Lecture Notes in Computer Science 2365, pp. 1–16, Springer-Verlag, 2002.
6. Eli Biham, Orr Dunkelman, Nathan Keller, *Rectangle Attacks on 49-Round SHACAL-1*, proceedings of Fast Software Encryption 10, Lecture Notes in Computer Science 2887, pp. 22–35, Springer-Verlag, 2003.
7. Eli Biham, Orr Dunkelman, Nathan Keller, *Related-Key Boomerang and Rectangle Attacks*, Advances in Cryptology, proceedings of EUROCRYPT '05, Lecture Notes in Computer Science 3494, pp. 507–525, Springer-Verlag, 2005.
8. Eli Biham, Orr Dunkelman, Nathan Keller, *Improved Slide Attacks*, private communication.
9. Eli Biham, Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
10. Alex Biryukov, David Wagner, *Slide Attacks*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 245–259, Springer-Verlag, 1999.
11. Alex Biryukov, David Wagner, *Advanced Slide Attacks*, Advances in Cryptology, proceedings of EUROCRYPT 2000, Lecture Notes in Computer Science 1807, pp. 586–606, Springer-Verlag, 2000.
12. Florent Chabaud, Antoine Joux, *Differential Collisions in SHA-0*, Advances in Cryptology, proceedings of CRYPTO 1998, Lecture Notes in Computer Science 1462, pp. 56–71, Springer-Verlag, 1998.
13. Orr Dunkelman, Nathan Keller, Jongsung Kim, *Related-Key Rectangle Attack on the Full SHACAL-1*, accepted to Selected Areas in Cryptography 2006, to appear in Lecture Notes in Computer Science.

14. Soichi Furuya, *Slide Attacks with a Known-Plaintext Cryptanalysis*, proceedings of Information and Communication Security 2001, Lecture Notes in Computer Science 2288, pp. 214–225, Springer-Verlag, 2002.
15. Helena Handschuh, Lars R. Knudsen, Matthew J. Robshaw, *Analysis of SHA-1 in Encryption Mode*, proceedings of CT-RSA 2001, Springer-Verlag Lecture Notes in Computer Science, vol. 2020, pp. 70–83, 2001.
16. Helena Handschuh, David Naccache, *SHACAL*, preproceedings of NESSIE first workshop, Leuven, 2000.
17. Seokhie Hong, Jongsung Kim, Guil Kim, Sangjin Lee, Bart Preneel, *Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192*, proceedings of Fast Software Encryption 12, Lecture Notes in Computer Science 3557, pp. 368–383, Springer-Verlag, 2005.
18. John Kelsey, Tadayoshi Kohno, Bruce Schneier, *Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent*, proceedings of Fast Software Encryption 7, Lecture Notes in Computer Science 1978, pp. 75–93, Springer-Verlag, 2000.
19. John Kelsey, Bruce Schneier, David Wagner, *Key-Schedule Cryptoanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES*, Advances in Cryptology, proceedings of CRYPTO '96, Lecture Notes in Computer Science 1109, pp. 237–251, Springer-Verlag, 1996.
20. Jongsung Kim, Guil Kim, Seokhie Hong, Sangjin Lee, Dowon Hong, *The Related-Key Rectangle Attack — Application to SHACAL-1*, proceedings of ACISP 2004, Lecture Notes in Computer Science 3108, pp. 123–136, Springer-Verlag, 2004.
21. Jongsung Kim, Dukjae Moon, Wonil Lee, Seokhie Hong, Sangjin Lee, Seokwon Jung, *Amplified Boomerang Attack against Reduced-Round SHACAL*, Advances in Cryptology, proceedings of ASIACRYPT 2002, Lecture Notes in Computer Science 2501, pp. 243-253, Springer-Verlag, 2002.
22. Lars R. Knudsen, *Cryptanalysis of LOKI91*, proceedings of Auscrypt 1992, Lecture Notes in Computer Science 718, pp. 196–208, Springer-Verlag, 1993.
23. Markku-Juhani O. Saarinen, *Cryptanalysis of Block Ciphers Based on SHA-1 and MD5*, proceedings of Fast Software Encryption 10, Lecture Notes in Computer Science 2887, pp. 36–44, Springer-Verlag, 2003.
24. NESSIE – New European Schemes for Signatures, Integrity and Encryption. *http://www.nessie.eu.org/nessie*
25. NESSIE, *Portfolio of recommended cryptographic primitives.*
26. NESSIE, *Performance of Optimized Implementations of the NESSIE Primitives*, NES/DOC/TEC/WP6/D21/2.
27. US National Bureau of Standards, *Secure Hash Standard*, Federal Information Processing Standards Publications No. 180-2, 2002.
28. Eteinee Van Den Bogeart, Vincent Rijmen, *Differential Analysis of SHACAL*, NESSIE internal report NES/DOC/KUL/WP3/009/a, 2001.
29. David Wagner, *The Boomerang Attack*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 156–170, 1999.
30. Xiaoyun Wang, Andrew C. Yao, Frances Yao, *Cryptanalysis on SHA-1*, Cryptographic Hash Workshop, NIST, Gaithersburg, 2005.
31. Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, Xiuyuan Yu, *Cryptanalysis of the Hash Functions MD4 and RIPEMD*, Advances in Cryptology, proceedings of EUROCRYPT 2005, Lecture Notes in Computer Science 3494, pp. 1–18, 2005.
32. Xiaoyun Wang, Yiqun Lisa Yin, Hongbo Yu, *Finding Collisions in the Full SHA-1*, Advances in Cryptology, proceedings of CRYPTO 2005, Lecture Notes in Computer Science 3621, pp. 17–36, 2005.

33. Xiaoyun Wang, Hongbo Yu, *How to Break MD5 and Other Hash Functions*, Advances in Cryptology, proceedings of EUROCRYPT 2005, Lecture Notes in Computer Science 3494, pp. 19–35, 2005.
34. Xiaoyun Wang, Hongbo Yu, Yiqun Lisa Yin, *Efficient Collision Search Attacks on SHA-0*, Advances in Cryptology, proceedings of CRYPTO 2005, Lecture Notes in Computer Science 3621, pp. 1–16, 2005.