Improved Cryptanalysis of SHAMATA-BC

Adem Atalay, Orhun Kara * and Ferhat Karakoç

TÜBİTAK UEKAE
{adematalay,orhun,ferhatk}@uekae.tubitak.gov.tr

Abstract. We state the design flaws of the 1-round block cipher SHAMATA-BC, designed by Fleishmann and Gorski by using the building blocks of SHAMATA hash function. We fix the flaws and then show that the amended version of SHAMATA-BC is much weaker. We believe that there is no connection between security level of SHAMATA as a hash function and that of SHAMATA-BC as a block cipher.

Key words: Hash function, SHAMATA, SHAMATA-BC, SHA-3.

1 Introduction

SHAMATA is a register based hash function, submitted to NIST as one of the contestant algorithms for SHA-3 competition [1].

SHAMATA is not a block cipher based hash function and hence it has no internal block cipher. On the other hand, in a very recent study, E. Fleismann and M. Gorski describe a one-round block cipher by using the building blocks of SHAMATA, which they call "SHAMATA-BC" and show that SHAMATA-BC is extremely very weak by deducing the key from one plaintext/ciphertext pair [5]. Nevertheless, we should emphasize that this attempt can neither make SHAMATA a block cipher based hash function nor assign SHAMATA-BC as the internal block cipher of SHAMATA.

We have seen that there are some flaws in both the definition and the design of SHAMATA-BC. In this note, we state the design flaws and fix them. Then, we show that the amended version of SHAMATA-BC is much weaker.

In SHAMATA, data is loaded into the registers and then the registers are clocked like sponge constructions [2]. The clocking mechanism is invertible. Roughly speaking, the clocking and data loading mechanism is defined as a block cipher in [5], so called SHAMATA-BC. Hence, it is in fact somehow trivial that SHAMATA-BC is a very weak block cipher.

* Supported in part by the European Commission through the project FP-7 ICE under the project grant number 206546.

Indeed, one can recover the current data (which is perceived as key in [5]) if the current internal state (which is perceived as plaintext in [5]) and the next internal state (which is perceived as ciphertext in [5]) is known. Almost all the register based hash constructions including sponge constructions have such a property. If one can construct an attack on SHAMATA hash function by making use of the attack on SHAMATA-BC, then it is no doubt that this type of attack can be easily generalized for almost all sponge constructions having invertible next state function.

2 Description of SHAMATA-BC and Its Analysis

SHAMATA-BC is defined by E. Fleishmann and M. Gorski ([5]) by making use of the building blocks of SHAMATA. This block cipher is of the form:

$$\text{SHAMATA-BC} : \{0,1\}^{512} \times \{0,1\}^{128} \rightarrow \{0,1\}^{512}$$

where a 512 bit message block is encrypted using a 128 bit key. Following the notation in [5], a plaintext is given as

$$X = B[0]||B[1]||B[2]||B[3]$$

and its corresponding ciphertext is

$$C = B^*[0]||B^*[1]||B^*[2]||B^*[3].$$

Let MC be the `MixColumns` operation. Let D be the key entered the encryption, $P = MC(D), Q = MC(D^T)$, where D^T is the transpose of data D and let $P' = P(1)||Q(0)$ and $Q' = Q(1)||P(0)$. The encryption of X is described as:

$$\begin{aligned}
B[2] &= B[2] \oplus P \oplus blockno, \\
B[3] &= B[3] \oplus Q \oplus blockno, \\
K[3] &= K[3] \oplus P', \\
K[5] &= K[5] \oplus Q, \\
K[7] &= K[7] \oplus P, \\
K[11] &= K[11] \oplus Q', \\
B^*[0] &= B[2], \\
B^*[1] &= B[3], \\
B^*[2] &= ARF^r(B[2]) \oplus B[0] \oplus K[9] \oplus K[0], \\
B^*[3] &= ARF^r(B[3]) \oplus B[1] \oplus K[10] \oplus K[1].
\end{aligned} \tag{1}$$

The Attack: E. Fleishmann and M. Gorski propose an attack on SHAMATA-BC. The key can be extracted using only one plaintext/ciphertext pair by assuming that K is empty at the beginning. The key is deduced by solving:

$$SB^{-1}(SR^{-1}(MC^{-1}(B^*[2] \oplus B[0]))) \oplus B[2] \oplus blockno = P \quad (2)$$
$$SB^{-1}(SR^{-1}(MC^{-1}(\Delta B^*[3] \oplus \Delta B[1]))) \oplus B[3] \oplus blockno = Q. \quad (3)$$

3 Design Flaws and Improved Cryptanalysis

Flaws: It is evident that SHAMATA-BC is not well-defined. First of all, it is given as a function with a domain space and a range space given in [5] as

$$\text{SHAMATA-BC } : \{0,1\}^{512} \times \{0,1\}^{128} \rightarrow \{0,1\}^{512}.$$

However, the input space of SHAMATA-BC includes some block of 1536-bit register K as well as B-register and the data D. Hence, the domain space does not fit the given parameters. It seems that SHAMATA-BC takes some parameters different from plaintext and key. People may ask what, for instance, $K[0]$ is in the definition of SHAMATA-BC. Well, indeed, we do not know as well. In addition, we should notice that we could not give a meaning to the Δ operations in Equation 3.

Another flaw is the ambiguous definition of blocks of the B register. For instance, $B[2]$ is updated by Equation 1. However, $B[2]$ used in Equation 2 is the original plaintext block without updating. In general, the updating procedure should update the outputs (B^*s) rather than the inputs (B's). So, we can rewrite the update procedure simply as

$$\begin{aligned}
B^*[2] &= B[2] \oplus P \oplus blockno,\\
B^*[3] &= B[3] \oplus Q \oplus blockno,\\
K^*[3] &= K[3] \oplus P',\\
K^*[5] &= K[5] \oplus Q,\\
K^*[7] &= K[7] \oplus P,\\
K^*[11] &= K[11] \oplus Q',\\
B^*[0] &= B^*[2],\\
B^*[1] &= B^*[3],\\
B^*[2] &= ARF^r(B^*[2]) \oplus B[0] \oplus K[9] \oplus K[0],\\
B^*[3] &= ARF^r(B^*[3]) \oplus B[1] \oplus K[10] \oplus K[1].
\end{aligned}$$

One more flaw is the definition $B^*[0]$ and $B^*[1]$. They should be defined as

$$B^*[0] = B[2] \oplus P \oplus blockno,$$
$$B^*[1] = B[3] \oplus Q \oplus blockno$$

rather than

$$B^*[0] = B[2], \tag{4}$$
$$B^*[1] = B[3] \tag{5}$$

according to the set of equations given in the description of SHAMATA-BC. It is clear from the equations in the definition of SHAMATA-BC that the half of the plaintext, i.e., $B[2]$ and $B[3]$, is influenced by the cipher key. However, the designers of SHAMATA-BC claims the opposite by pointing at the misleading equations, e.g., Equation 4:

> At this stage it is obvious that half if [1] the plaintext, i.e., $B[2]$ and $B[3]$, is not influenced by the cipher key.

Improved Attack: Deducing the key from the amended block cipher is much easier this time. Even, we do not need to assume that K is empty. It is obvious that one can deduce both P and Q just by XOR'ing the left part of the ciphertext with the right part of the corresponding plaintext:

$$P = B^*[0] \oplus B[2] \oplus blockno \text{ and}$$

$$Q = B^*[1] \oplus B[3] \oplus blockno.$$

So, we do not need to operate the inverse functions of `SubBytes`, `ShiftRows` and `MixColumns` to extract P and Q.

4 Conclusion and Discussion

In this note, we have shown some flaws in the design of the block cipher SHAMATA-BC. We fix the flaws and then improve the attack on SHAMATA-BC mounted by the designers of SHAMATA-BC. Hence, we have shown that the block cipher SHAMATA-BC is extremely weak.

SHAMATA is a register based hash function. Data is loaded into the registers and then the registers are clocked like sponge constructions [2]. The clocking mechanism is invertible. Roughly speaking, the clocking and

[1] should be "of" we guess!

data loading mechanism is described as SHAMATA-BC. Hence, It is in fact somehow trivial that SHAMATA-BC is a very weak block cipher. Indeed, one can recover the current data (which is perceived as key in [5]) if the current internal state (which is perceived as plaintext in [5]) and the next internal state (which is perceived as ciphertext in [5]) is known. Almost all the register based hash constructions including sponge constructions have such a property.

The security of a hash function based on a block cipher depends on the security of its internal block cipher like SHA-1[6]. The underlying block cipher SHACAL-1 has been successfully attacked in [3, 4] which influences the security level of SHA-1. Consequently, it is a plausible and also a common strategy to analyze the underlying block cipher of a block cipher based hash function to deduce information about its security.

SHAMATA is not a block cipher based hash function and it has no internal block cipher. However, several block ciphers can be designed by using the building blocks of SHAMATA. One of them, and maybe the weakest one, is SHAMATA-BC. Nevertheless, we should emphasize that this attempt can neither make SHAMATA a block cipher based hash function nor assign SHAMATA-BC as the internal block cipher of SHAMATA.

Disclaimer

We, as the designers of SHAMATA, have no concern with the block cipher SHAMATA-BC and cannot give any guarantee or warranty that SHAMATA-BC is fit for any particular purpose.

References

1. A.Atalay, O.Kara, F.Karakoc, C.Manap. SHAMATA Hash Function Algorithm Specifications, 2008. Available online at `http://www.uekae.tubitak.gov.tr/uekae_content_files/crypto/SHAMATA_Specification.pdf`
2. G.Bertoni, J.Daemen, M. Peeters, G.V. Assche. Sponge Functions. *Ecrypt Hash Workshop 2007*
3. E.Biham, O.Dunkelman, N.Keller. A simple Related-Key Attack on the Full SHACAL-1. In M. Abe, editor. *CT_RSA*, LNCS volume 4377, p20-30, Springer 2007.
4. O.Dunkelman, N.Keller, J.Kim. Related-Key Rectangle Attack on the Full SHACAL-1. In E.Biham and A.M.Youssef, editors, *Selected Areas in Cryptography*, LNCS volume 4356, p28-44,Springer,2006.
5. E.Fleishmann and M.Gorski. Some Observations on SHAMATA, 2008. Available online at `http://www.uni-weimar.de/cms/fileadmin/medien/medsicherheit/Research/SHA3/Observations_for_SHAMATA.pdf`

6. National Institute of Standards and Technology. Cryptographic Hash Project. Available online at `http://csrc.nist.gov /groups /ST/hash/index.html`
7. X.Wang, Y.Lisa, H. Yu. Finding Collisions in the Full SHA-1. In V. Shoup, editor *CRYPTO*, LNCS volume 3621, p17-36, Springer, 2005.