Cryptanalysis of ESSENCE

María Naya-Plasencia[1,*], Andrea Röck[1], Jean-Philippe Aumasson[2,†], Yann Laigle-Chapuy[1], Gaëtan Leurent[3], Willi Meier[2,‡], and Thomas Peyrin[4]

[1] INRIA project-team SECRET, France
[2] FHNW, Windisch, Switzerland
[3] École Normale Supérieure, Paris, France
[4] Ingenico, France

Abstract. ESSENCE is a hash function submitted to the NIST Hash Competition that stands out as a hardware-friendly and highly parallelizable design, and that has thus far remained unbroken. Preliminary analysis in its documentation argues that it resists standard differential cryptanalysis. This paper disproves this claim, showing that advanced techniques can be used to significantly reduce the cost of such attacks: using a manually found differential characteristic and a nontrivial search algorithm, we obtain shortcut collision attacks on the full ESSENCE-256 and ESSENCE-512, with respective complexities 2^{91} and 2^{168}. As an aside, we show how to use these attacks for forging valid message/MAC pairs for HMAC-ESSENCE-256 and HMAC-ESSENCE-512, essentially at the same cost as a collision.

1 Introduction

Recent years have seen a surge of research on cryptographic hashing, since devastating attacks [12, 11, 2, 10] on the two most deployed hash functions, MD5 and SHA-1. The consequent lack of confidence in the current NIST standard SHA-2 [7], stemming from its similarity with those algorithms, motivated NIST to launch the *NIST Hash Competition*, a public competition to develop a new hash standard, which will be called SHA-3 and announced by 2012. NIST received 64 submissions, and accepted 51 as first round candidates. As of June 2009, more than 20 of those were shown to have significant weaknesses[5]. That competition catches the attention not only from many academics, but also from industry—with candidates from IBM, Hitachi, Intel, Sony—and from governmental organizations.

ESSENCE [3, 4] is a first round candidate in the NIST Hash Competition that like many others has two main instances, operating on 32- and 64-bit words, respectively: ESSENCE-256 and ESSENCE-512. These functions process messages

*Supported in part by the French Agence Nationale de la Recherche under contract ANR-06-SETI-013-RAPIDE.

†Supported by the Swiss National Science Foundation under project no. 113329.

‡Supported by GEBERT RÜF STIFTUNG, project no. GRS-069/07.

[5] See ECRYPT's SHA-3 Zoo: `http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo`.

using a binary tree structure, and use a simple compression algorithm based on two non-linear feedback shift registers (NFSR's). ESSENCE shares similarities with MD6 [9], for example the tree-hashing and the hardware-friendly compression function.

This paper presents shortcut collision attacks on ESSENCE-256 and ESSENCE-512. At the heart of our attacks is a single differential characteristic, and our main technical achievement is a nontrivial method for searching inputs conforming to this differential at a reduced cost. As an aside, we describe how to use these attacks for forging valid message/MAC pairs for HMAC-ESSENCE-256 and HMAC-ESSENCE-512 in far fewer than $2^{n/2}$ trials. These findings show that ESSENCE does not satisfy the security requirements set by NIST for the future SHA-3.

In a parallel work, Mouha et al. [6] present results on reduced versions of ESSENCE, including a pseudo-collision attack on ESSENCE-512 reduced to 31 steps. They exploit a differential characteristic of a different type than ours, and also use different techniques to search for conforming inputs.

The rest of the paper is organized as follows: §2 briefly introduces ESSENCE; §3 describes our method for searching collisions and its complexity analysis; §4 shows how to attack the HMAC construction when instantiated with ESSENCE, and finally §5 concludes.

2 Brief description of ESSENCE

We give a brief description of the ESSENCE hash functions, which should be sufficient to understand our attacks. A complete specification can be found in [3, 4]. Henceforth statements of (non-) linearity are relative to the field $\mathrm{GF}(2) = \{0, 1\}$ or to an extension thereof.

2.1 Structure

ESSENCE processes a message by constructing a balanced binary tree of bounded depth whose leaves correspond to calls to a compression function with message chunks as input. More precisely, each leaf corresponds to a series of three sequential compressions, with a unique initial value for each leaf that depends on several parameters of the hash function. Likewise, nodes correspond to series of three compressions, with children chaining values as input.

After creation of all message blocks, one appends a *final block* to the data to be hashed. This block contains parameters of the function as well as message-dependent information, and it potentially assists prevent near-collision attacks.

2.2 Compression Function

The compression function of ESSENCE takes as input an eight-word chaining value and an eight-word message block. Words are 32-bit for ESSENCE-256 and 64-bit for ESSENCE-512, so those values are respectively 256- and 512-bit.

Versions of ESSENCE with 224- and 384-bit digests are derived from the main instances by tweaking parameters and truncation of the final digest.

The compression function uses two NFSR's, each operating on a register of eight words:

- $r = (r0, \ldots, r7)$ is initialized with the chaining value, and
- $k = (k0, \ldots, k7)$ is initialized with the message block.

At each step of the compression algorithm, the mechanism in Fig. 1 is clocked using a non-linear function F (see Fig. 2), and a linear function L that provides diffusion accross word slices. This mechanism defines a permutation and the compression function returns as new chaining value the XOR of the r register with its initial value, as in the Davies-Meyer scheme.

The documentation of ESSENCE recommends at least 24 steps, and set 32 steps in the actual submission for extra precaution [4, §4].

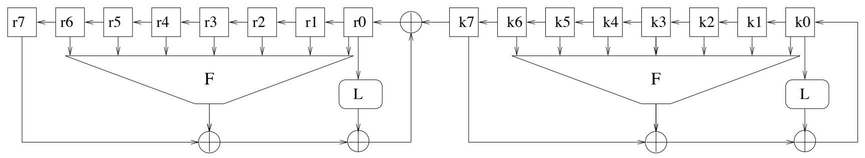

Fig. 1. Overview of the compression function of ESSENCE.

$$\begin{aligned} F(a,b,c,d,e,f,g) = \; & abcdefg + abcdef + abcefg + acdefg + abceg + \\ & abdef + abdeg + abefg + acdef + acdfg + acefg + \\ & adefg + bcdfg + bdefg + cdefg + abcf + abcg + \\ & abdg + acdf + adef + adeg + adfg + bcde + \\ & bceg + bdeg + cdef + abc + abe + abf + abg + \\ & acg + adf + adg + aef + aeg + bcf + bcg + bde + \\ & bdf + beg + bfg + cde + cdf + def + deg + dfg + \\ & ad + ae + bc + bd + cd + ce + df + dg + ef + fg + \\ & a + b + c + f + 1 \end{aligned}$$

Fig. 2. The F function of ESSENCE, which takes seven words as input and operates in a bitsliced way (that is, the i-th bit of the output word only depends on the i-th bits of the input words).

Table 1. Differential characteristic for finding collisions on (both versions of) ESSENCE; α and β are differences such that $\beta = L(\alpha)$ and $\alpha \vee \beta \vee L(\beta) = \alpha \vee \beta$. A "·" denotes an absence of difference. Values in the column "Pr" are *heuristical approximations* of the probability to reach the *next* difference (exact probabilities significantly differ, and can be estimated empirically, cf. §§3.3).

Pr	Chaining value part									Message part								Pr
1	·	·	·	·	·	·	·	·	0	α	β	·	·	·	·	·	·	$2^{-\vert\beta\vert}$
$2^{-\vert\alpha\vert}$	·	·	·	·	·	·	·	α	1	β	·	·	·	·	·	·	α	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	·	·	·	·	·	α	·	2	·	·	·	·	·	·	α	·	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	·	·	·	·	α	·	·	3	·	·	·	·	·	α	·	·	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	·	·	·	α	·	·	·	4	·	·	·	·	α	·	·	·	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	·	·	α	·	·	·	·	5	·	·	·	α	·	·	·	·	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	·	α	·	·	·	·	·	6	·	·	α	·	·	·	·	·	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	α	·	·	·	·	·	·	7	·	α	·	·	·	·	·	·	$2^{-\vert\alpha\vert}$
1	α	·	·	·	·	·	·	·	8	α	·	·	·	·	·	·	·	1
1	·	·	·	·	·	·	·	·	9	·	·	·	·	·	·	·	α	$2^{-\vert\alpha\vert}$
1	·	·	·	·	·	·	·	·	10	·	·	·	·	·	·	α	β	$2^{-\vert\alpha\vee\beta\vert}$
1	·	·	·	·	·	·	·	·	11	·	·	·	·	·	α	β	·	$2^{-\vert\alpha\vee\beta\vert}$
1	·	·	·	·	·	·	·	·	12	·	·	·	·	α	β	·	·	$2^{-\vert\alpha\vee\beta\vert}$
1	·	·	·	·	·	·	·	·	13	·	·	·	α	β	·	·	·	$2^{-\vert\alpha\vee\beta\vert}$
1	·	·	·	·	·	·	·	·	14	·	·	α	β	·	·	·	·	$2^{-\vert\alpha\vee\beta\vert}$
1	·	·	·	·	·	·	·	·	15	·	α	β	·	·	·	·	·	$2^{-\vert\alpha\vee\beta\vert}$
1	·	·	·	·	·	·	·	·	16	α	β	·	·	·	·	·	·	$2^{-\vert\beta\vert}$
$2^{-\vert\alpha\vert}$	·	·	·	·	·	·	·	α	17	β	·	·	·	·	·	·	α	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	·	·	·	·	·	α	·	18	·	·	·	·	·	·	α	·	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	·	·	·	·	α	·	·	19	·	·	·	·	·	α	·	·	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	·	·	·	α	·	·	·	20	·	·	·	·	α	·	·	·	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	·	·	α	·	·	·	·	21	·	·	·	α	·	·	·	·	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	·	α	·	·	·	·	·	22	·	·	α	·	·	·	·	·	$2^{-\vert\alpha\vert}$
$2^{-\vert\alpha\vert}$	·	α	·	·	·	·	·	·	23	·	α	·	·	·	·	·	·	$2^{-\vert\alpha\vert}$
1	α	·	·	·	·	·	·	·	24	α	·	·	·	·	·	·	·	1
1	·	·	·	·	·	·	·	·	25	·	·	·	·	·	·	·	α	1
1	·	·	·	·	·	·	·	·	26	·	·	·	·	·	·	α	?	1
1	·	·	·	·	·	·	·	·	27	·	·	·	·	·	α	?	?	1
1	·	·	·	·	·	·	·	·	28	·	·	·	·	α	?	?	?	1
1	·	·	·	·	·	·	·	·	29	·	·	·	α	?	?	?	?	1
1	·	·	·	·	·	·	·	·	30	·	·	α	?	?	?	?	?	1
1	·	·	·	·	·	·	·	·	31	·	α	?	?	?	?	?	?	1
1	·	·	·	·	·	·	·	·	32	α	?	?	?	?	?	?	?	1

3 Collision Attacks on ESSENCE

Table 1 presents a differential characteristic for finding collisions on the compression function of ESSENCE. It is used for both ESSENCE-256 and ESSENCE-512. We found this characteristic manually, i.e., without the assistance of any automated search. Because it has no input difference in the chaining value, it can directly be used for searching colliding message blocks with respect to a same chaining value. The collision attack will then consist in

1. Finding one message block that fulfills the differential on the right part.
2. Trying chaining values until one conforms to the differential on the left part.

For the second phase of the attack, distinct pseudorandom chaining values are obtained by picking a first pseudorandom (sequence of) message block(s), and then checking differences after the insertion of the next message block.

The subsequent sections work out the details of the attack as follows:

- §§3.1 explains how the characteristic works.
- §§3.2 presents an efficient method for finding a message block that conforms to the differential.
- §§3.3 discusses calculation of the complexity; contrary to many similar differential attacks, an approximation solely based on Hamming weight is insufficient to obtain accurate probability estimates. Actually such heuristics *underestimate* the actual complexity of the attack, as we will see later.

Thereafter we use the following notations: $\vee$ for logical OR between two bits (or two words); $\wedge$ for logical AND; $\neg$ for bitwise negation; $|w|$ for the Hamming weight of word w; w_i for the i-th bit of word w, $0 \leq i < 32$ for ESSENCE-256, and $0 \leq i < 64$ for ESSENCE-512.

3.1 The Characteristic

The differential characteristic on Table 1 starts with a difference in the message block, and no difference in the chaining value. To follow the characteristic, the only assumption that we will make is that the function F will "absorb" certain differences (actually most of them) and "preserve" some others (at step 11). Therefore, the probability that a randomly chosen input conforms to the differential will essentially depend on the Hamming weight of the wordwise differences α and $\beta = L(\alpha)$. Critical steps are listed below:

- **Step 0**: α is fed back to the rightmost cell of the right register via an XOR, and it does not enter F, unlike β. To ensure that no difference will appear in the output of F, we need all the $|\beta|$ bit differences be absorbed, which is expected to occur with probability $2^{-|\beta|}$ (such heuristical estimates should not be used systematically, as discussed later).

- **Step 1**: the relation $\beta = L(\alpha)$ makes differences introduced in the new rightmost cell vanish. This always works, but we also need that α adds no difference, that is, F needs to absorb $|\alpha|$ bit differences, thus the probability $2^{-|\alpha|}$ on both parts.
- **Steps 2 to 7**: we assume again that the $|\alpha|$ differences introduced in F are aborbed.
- **Step 8**: the two α differences cancel out in the middle of the mechanism, but α is also fed back to the rightmost cell of the message register.
- **Step 9**: unlike at step 1, α will introduce a difference $L(\alpha) = \beta$, which will propagate during steps 11 to 17.
- **Step 10**: to avoid the introduction of new differences, we need the output of F to have differences $L(\beta)$, in order the differences to vanish in the feedback operation. This will only be possible if $\alpha \vee \beta \vee L(\beta) = \alpha \vee \beta$. As we will see later, for avoiding impossibilities of differential paths, we also have to add the condition $L(\beta) \wedge \alpha \wedge \neg\beta = 0$.
- **Steps 16 to 24**: the characteristic is the same as in steps 0 to 8.
- **Steps 25 to 32**: note that differences in the right side after 32 steps do not affect the value returned by the compression function. We thus put no condition on those particular differences.

After finding this generic characteristic, it remains to search for an α that minimizes the cost of the attack. But before that, we will present a generic method for finding a message block conforming to the right part of the differential.

3.2 Efficient Search for a Conforming Block

Once we have found low-weight α and $\beta = L(\alpha)$ such that

$$\begin{aligned} &\alpha \vee \beta \vee L(\beta) = \alpha \vee \beta \text{ and} \\ &L(\beta) \wedge \alpha \wedge \neg\beta = 0 \ , \end{aligned}$$

the complexity of finding a conforming block by repeated trials is heuristically

$$2^{15|\alpha|+2|\beta|+6|\alpha\vee\beta|} \ .$$

This complexity is well above the birthday bound $2^{n/2}$ for all differences we found, let alone the fact that it underestimates the real complexity. For example, for the difference that we use to attack ESSENCE-256, the above expression yields a complexity 2^{210}, whereas a birthday attack needs only 2^{128} trials.

To find a conforming block at a reduced cost, we use a strategy somewhat similar to that of the rebound attack [5], namely, we start by finding conforming values for the low-probability differential in the middle, then we check that they follow the simpler differentials in both directions. What we call the *middle section* will corresponds to *steps 10 to 17*, inclusive. More precisely, we will

1. Find many values that conform to the middle section (i.e., steps 10 to 17);

2. Search, among those values, one that conforms to the differential in steps 0 to 9, and 18 to 23 (any such value will then follow the characteristic up to step 32).

Note that we need to find approximately $2^{15|\alpha|+|\beta|}$ messages in the first phase, in order to have a conforming one with high probability in the second phase. Below we expose our strategy for efficiently finding many values conforming to the differential between steps 10 and 17.

First we introduce some notations, to describe the state during the middle section: in Table 2 each x_i corresponds to a 32 or 64-bit word, depending on the version used. We write $\mathcal{S}$ the set of all indices where $\alpha \vee \beta$ is nonzero, that is,

$$\mathcal{S} = \{i, 0 \leq i < 32, \alpha_i \vee \beta_i = 1\} \quad \text{for ESSENCE-256,}$$
$$\mathcal{S} = \{i, 0 \leq i < 64, \alpha_i \vee \beta_i = 1\} \quad \text{for ESSENCE-512.}$$

We write $s = |\alpha \vee \beta| = |\mathcal{S}|$ the cardinality of $\mathcal{S}$. For example, if $\alpha = \texttt{80000000}$ and $\beta = \texttt{00000004}$, then $\alpha_{31} = \beta_2 = 1$, and so $\mathcal{S} = \{2, 31\}$ and $s = 2$. We also write ℓ for the word bitlength (32 or 64, depending on the version of ESSENCE).

Table 2. Message part in steps 10-17.

10	x_0	x_1	x_2	x_3	x_4	x_5	$x_6 \oplus \alpha$	$x_7 \oplus \beta$
11	x_1	x_2	x_3	x_4	x_5	$x_6 \oplus \alpha$	$x_7 \oplus \beta$	x_8
12	x_2	x_3	x_4	x_5	$x_6 \oplus \alpha$	$x_7 \oplus \beta$	x_8	x_9
13	x_3	x_4	x_5	$x_6 \oplus \alpha$	$x_7 \oplus \beta$	x_8	x_9	x_{10}
14	x_4	x_5	$x_6 \oplus \alpha$	$x_7 \oplus \beta$	x_8	x_9	x_{10}	x_{11}
15	x_5	$x_6 \oplus \alpha$	$x_7 \oplus \beta$	x_8	x_9	x_{10}	x_{11}	x_{12}
16	$x_6 \oplus \alpha$	$x_7 \oplus \beta$	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}
17	$x_7 \oplus \beta$	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}

To search for values conforming to the middle section, we first look at an arbitrary slice i, and we count the number of possible tuples $(x_{1,i}, \ldots, x_{13,i})$ that fulfill the path between steps 10 and 17. Writing $\gamma = L(\beta)$, and considering the example $(\alpha_i, \beta_i, \gamma_i) = (1, 0, 0)$, this corresponds to all tuples that satisfy the subsequent equations:

$$F(x_{1,i}, x_{2,i}, x_{3,i}, x_{4,i}, x_{5,i}, x_{6,i}, x_{7,i}) = F(x_{1,i}, x_{2,i}, x_{3,i}, x_{4,i}, x_{5,i}, x_{6,i} \oplus 1, x_{7,i})$$
$$F(x_{2,i}, x_{3,i}, x_{4,i}, x_{5,i}, x_{6,i}, x_{7,i}, x_{8,i}) = F(x_{2,i}, x_{3,i}, x_{4,i}, x_{5,i}, x_{6,i} \oplus 1, x_{7,i}, x_{8,i})$$
$$F(x_{3,i}, x_{4,i}, x_{5,i}, x_{6,i}, x_{7,i}, x_{8,i}, x_{9,i}) = F(x_{3,i}, x_{4,i}, x_{5,i}, x_{6,i} \oplus 1, x_{7,i}, x_{8,i}, x_{9,i})$$
$$F(x_{4,i}, x_{5,i}, x_{6,i}, x_{7,i}, x_{8,i}, x_{9,i}, x_{10,i},) = F(x_{4,i}, x_{5,i}, x_{6,i} \oplus 1, x_{7,i}, x_{8,i}, x_{9,i}, x_{10,i},)$$
$$F(x_{5,i}, x_{6,i}, x_{7,i}, x_{8,i}, x_{9,i}, x_{10,i}, x_{11,i}) = F(x_{5,i}, x_{6,i} \oplus 1, x_{6,i}, x_{8,i}, x_{9,i}, x_{10,i}, x_{11,i})$$
$$F(x_{6,i}, x_{7,i}, x_{8,i}, x_{9,i}, x_{10,i}, x_{11,i}, x_{10,i}) = F(x_{6,i} \oplus 1, x_{7,i}, x_{8,i}, x_{9,i}, x_{10,i}, x_{11,i}, x_{12,i})$$

For slices such that $\gamma_i = 1$, we have to produce a difference in F in order to erase γ_i. Table 3 reports the number of solutions for the x_i's depending on $(\alpha_i, \beta_i, \gamma_i)$.

The case $(1,0,1)$ is never going to be used, because it leads to an impossibility of following the differential.

Table 3. Number of solutions for the $(x_0,\ldots,x_{13})$ depending on the input differences.

γ_i	(α_i,β_i) $(0,1)$	$(1,0)$	$(1,1)$
0	24	56	58
1	32	64	82

Then, for each slice $i \in \mathcal{S}$ we fix one of these tuples and try to compute the missing bits. The number of possibilities to choose the tuples for $i \in \mathcal{S}$ is

$$N_\alpha = 56^{|\alpha\wedge\neg\beta\wedge\neg\gamma|} \times 58^{|\alpha\wedge\beta\wedge\neg\gamma|} \times 24^{|\neg\alpha\wedge\beta\wedge\neg\gamma|} \times 82^{|\alpha\wedge\beta\wedge\gamma|} \times 32^{|\neg\alpha\wedge\beta\wedge\gamma|} .$$

Note that to follow the characteristic, the equations below (directly derived from the ESSENCE mechanism) must hold:

$$L(\overbrace{x_7}^{s\text{ bits fixed}}) = x_0 \oplus \overbrace{x_8 \oplus L(\beta) \oplus F(x_1,x_2,x_3,x_4,x_5,x_6\oplus\alpha,x_7\oplus\beta)}^{s\text{ bits fixed}} \quad (1)$$

$$L(\overbrace{x_8}^{s\text{ bits fixed}}) = \overbrace{x_1 \oplus x_9 \oplus F(x_2,x_3,x_4,x_5,x_6\oplus\alpha,x_7\oplus\beta,x_8)}^{s\text{ bits fixed}} \quad (2)$$

$$L(\overbrace{x_9}^{s\text{ bits fixed}}) = \overbrace{x_2 \oplus x_{10} \oplus F(x_3,x_4,x_5,x_6\oplus\alpha,x_7\oplus\beta,x_8,x_9)}^{s\text{ bits fixed}} \quad (3)$$

$$L(\overbrace{x_{10}}^{s\text{ bits fixed}}) = \overbrace{x_3 \oplus x_{11} \oplus F(x_4,x_5,x_6\oplus\alpha,x_7\oplus\beta,x_8,x_9,x_{10})}^{s\text{ bits fixed}} \quad (4)$$

$$L(\overbrace{x_{11}}^{s\text{ bits fixed}}) = \overbrace{x_4 \oplus x_{12} \oplus F(x_5,x_6\oplus\alpha,x_7\oplus\beta,x_8,x_9,x_{10},x_{11})}^{s\text{ bits fixed}} \quad (5)$$

$$L(\overbrace{x_{12}}^{s\text{ bits fixed}}) = \overbrace{x_5 \oplus x_{13} \oplus F(x_6\oplus\alpha,x_7\oplus\beta,x_8,x_9,x_{10},x_{11},x_{12})}^{s\text{ bits fixed}} \quad (6)$$

$$L(\overbrace{x_{13}}^{s\text{ bits fixed}}) = x_{14} \oplus \overbrace{x_6 \oplus F(x_7\oplus\beta,x_8,x_9,x_{10},x_{11},x_{12},x_{13})}^{s\text{ bits fixed}} \quad (7)$$

The bits fixed in $x_1,\ldots,x_{13}$ are those in the slices $i \in \mathcal{S}$. Each of Eq. (2)-(6) corresponds to a system of ℓ equations between bits, wherein $2s$ variables are fixed and $2(\ell-s)$ are free. We shall assume that for a random choice of the $2s$ fixed variables, each system admits a solution with probability approximately $2^{\ell-2s}$. Therefore, for a given choice of $(x_{1,i},\ldots,x_{13,i})$, $i \in \mathcal{S}$, we have a probability of $2^{5(\ell-2s)}$ of a valid solution in Eq. (2)-(6). A solution will give us values conforming to the differential. To solve the systems of all seven equations, one then proceed as follows:

1. Fix the s bits in each $x_1, \dots, x_{13}$ to one of the N_α admissible values;
2. Solve the systems of Eq. (2) to (6) for the bits at the s positions, by determining the $(\ell - s)$ values left of $x_8, x_9, x_{10}, x_{11}, x_{12}$. This step succeeds with probability $2^{5(\ell-2s)}$, as previously explained. Finding one solution thus requires approximately $2^{5(2s-\ell)}$ trials.
   Once a solution is found, all the values entering L and all the bits in slices in $\mathcal{S}$ are fixed. Furthermore, all the bits in $x_8, x_9, x_{10}, x_{11}, x_{12}$ are fixed, and for all slices $i \notin \mathcal{S}$ we know the values of
   $$\begin{aligned} &x_{1,i} \oplus x_{9,i} \oplus F(x_{2,i}, x_{3,i}, x_{4,i}, x_{5,i}, x_{6,i}, x_{7,i}, x_{8,i})_i \\ &x_{2,i} \oplus x_{10,i} \oplus F(x_{3,i}, x_{4,i}, x_{5,i}, x_{6,i}, x_{7,i}, x_{8,i}, x_{9,i})_i \\ &x_{3,i} \oplus x_{11,i} \oplus F(x_{4,i}, x_{5,i}, x_{6,i}, x_{7,i}, x_{8,i}, x_{9,i}, x_{10,i})_i \\ &x_{4,i} \oplus x_{12,i} \oplus F(x_{5,i}, x_{6,i}, x_{7,i}, x_{8,i}, x_{9,i}, x_{10,i}, x_{11,i})_i \\ &x_{5,i} \oplus x_{13,i} \oplus F(x_{6,i}, x_{7,i}, x_{8,i}, x_{9,i}, x_{10,i}, x_{11,i}, x_{12,i})_i \end{aligned}$$
3. Now we can choose freely the $(\ell - s)$ bits left on x_5, x_6, x_7, and for each valuation we determine the remaining bits of x_{13} (from Eq. (6)), of x_{14} (from Eq. (7)), and of x_4, x_3, x_2, x_1 (from Eq. (5) to (1)). Thus, each valid solution to Eq. (1)-(7) gives $2^{3(\ell-s)}$ additional results by exploiting the extra degrees of freedom.

We obtain in total about $N_\alpha \cdot 2^{5(\ell-2s)} \cdot 2^{3(\ell-s)} \cdot 2^{-1}$ possible pairs that satisfy the path from step 10 to 17. The factor 2^{-1} comes from the fact that we counted each possible pair twice.

The subsequent sections discuss the complexity of performing the search of the rest of the path, and give concrete complexity estimates for each instance of ESSENCE.

3.3 Finding Accurate Probabilities

Relying only on the Hamming weight to approximate the probability of the differential gives unacceptably inaccurate approximations. Indeed, for a given word slice, probabilities to be absorbed at each step are not independent, and neglecting this leads to estimates far from actual values. For example, a single bit difference is absorbed during seven steps with probability $2^{-8.41}$, which is significantly lower than the heuristical estimate 2^{-7}. However, for the differentials considered, the dependency between word slices seems negligible. We thus give complexities with respect to empirical estimates, computed independently for each word slice. That is, we compute the probability of the differential as 32 (or 64) independent characteristics, i.e., one for each slice. We could then estimate the real probability of our path for any given difference α. We found that having $\alpha = 1$, $\beta = 0$ and $L(\beta) = 1$ leads to an impossibility (the differential will never be satisfied for that α). This is why we need the condition

$$L(\beta) \wedge \alpha \wedge \neg\beta = 0 \ .$$

When considering the middle section, we also computed the real probability of verifying the sliced path once this part of the differential is satisfied. The complexities given in the next section were computed with respect to those empirical estimates, not with the heuristical values based only on the Hamming weight.

Reusing the notations α, β, γ from §§3.2, we give below the probabilities for *a given slice* i to follow the complete path on 32 steps (the impossible cases—of probability zero—are not included), depending on $(\alpha_i, \beta_i, \gamma_i) \in \{0,1\}^3$:

- $(0,1,0) : 2^{-9.5}$
- $(0,1,1) : 2^{-9}$
- $(1,0,0) : 2^{-24.4}$
- $(1,1,0) : 2^{-23}$
- $(1,1,1) : 2^{-26}$

The probability that a random input follows the path is then the product of those probabilities, with each raised to a power that equals the number of slices corresponding to this case. For the α's used in our attacks, we obtain probabilities 2^{-236} and 2^{-461}, respectively for ESSENCE-256 and ESSENCE-512.

Taking into account our technique in §§3.2 for solving the middle at a reduced cost, we obtain the probabilities

- $(0,1,0) : 2^{-1}$
- $(0,1,1) : 2^{-1}$
- $(1,0,0) : 2^{-17.2}$
- $(1,1,0) : 2^{-15.9}$
- $(1,1,1) : 2^{-19.4}$

Given those numbers, we find that the probability that a value conforming to the middle section follows the rest of the differential is 2^{-91} for ESSENCE-256 and $2^{-168.2}$ for ESSENCE-512.

There are at least two ways to compute the total number of message pairs that is going to satisfy the whole path. As we will get nearly the same result with both of them, we can verify that our estimations are correct. First, some additional notations are required: we let $\rho_0, \dots, \rho_{\ell-1}$ denote the probabilities for each slice in $0, \dots, \ell-1$ of conforming to the differential, i.e., each ρ_i will lie in $\{2^{-9.5}, 2^{9}, 2^{24.4}, 2^{23}, 2^{26}\}$; and we let $\tau_0, \dots, \tau_{\ell-1}$ be the conditional probabilities for each slice to follow the differential, assuming that the differential of the middle section is satisfied. Now, the two equivalent ways to express the number of conforming messages are:

1. The probability of the whole path is $\prod_{i=0}^{\ell-1} \rho_i$, hence the number of pairs of conforming messages is
$$2^{8\ell} \cdot \prod_{i=0}^{\ell-1} \rho_i \ ,$$
where 8ℓ is the digest bit length.

2. The probability of the path once the middle section is satisfied is $\prod_{i=0}^{\ell-1}\tau_i$; calling N the number of pairs conforming to the middle section, the number of conforming message pairs is then
$$N \cdot \prod_{i=0}^{\ell-1}\tau_i \ .$$

We verified that these two ways of computing the total number yield almost identical values, which gives evidence of the accuracy of our estimates.

3.4 Collisions for ESSENCE-256

For ESSENCE-256, we could perform an exhaustive search over all differences and found as optimal value $\alpha = \texttt{80102040}$, for which $|\alpha| = 4$, $|\beta| = 18$, and $|\alpha \vee \beta| = s = 19$. Heuristic estimates suggest that we need about $2^{15\times4+18} = 2^{78}$ messages that conform to the middle section to find at least one conforming to the differential on the right side. However, the real complexity is (cf. §§3.3) approximately 2^{91}.

Solving the right side. For that α, we have in total
$$24^6 \times 56^1 \times 32^9 \times 82^3 \approx 2^{97.7}$$
possibilities to set the bits in $\mathcal{S}$. We have a probability $2^{5(32-2\times19)} = 2^{-30}$ of finding a solution to the five systems defined by Eq. (2) to (6). Following our assumption in §§3.2, we get about $2^{67.7}$ solutions. For each solution, we obtain 2^{39} additional solutions by varying bits in slices i not in $\mathcal{S}$, yielding in total up to $2^{106.7}$ solutions.

For each message pair found, we must check that it satisfies the rest of the differential. As found in §§3.3, we need about 2^{91} values conforming to the middle section to find one value following the rest of the differential. Below we detail the cost of finding those messages.

For each solution of Eq. (2)-(6) we obtain 2^{39} additional pairs essentially "for free", thus we only need to find $2^{91-39} = 2^{52}$ solutions to the system. Since each solution costs about 2^{30} trials, we have to test about $2^{52} \times 2^{30} = 2^{82}$ distinct choices of s fixed tuples. We thus have a complexity 2^{82} for finding 2^{91} pairs, and a complexity 2^{91} for checking that they follow the differential. To recap, the total cost of the attack is less than that of computing 2^{91} compressions of ESSENCE-256, since a majority of the 2^{91} trials need compute at most two steps of ESSENCE, and the others at most 23 steps.

Solving the left side. Once a conforming pair of message blocks is found, we just need to try approximately 2^{64} distinct random chaining values to find a collision (for comparison, the heuristical estimate is $2^{14\times4} = 2^{56}$). Without increasing the total complexity, we can find $2^{91-64} = 2^{27}$ collisions, whereas this requires 2^{155} trials with the generic attack. Note that our attack can be carried out with negligible memory (the 2^{91} messages that satisfy the middle section don't have to be stored: we test repeatedly each candidate message, and discard it if it does not conform to the full differential).

3.5 Collisions for ESSENCE-512

For ESSENCE-512, the best difference found is $\alpha = \texttt{8408400000480082}$, giving $|\alpha| = 8$, $|\beta| = 35$, and $|\alpha \vee \beta| = s = 39$. As discussed in §§3.3, we need about $2^{168.2}$ solutions of the middle section to find one solution for the right side of the differential (against 2^{155} with heuristical estimates).

Solving the right side. For our α we have

$$24^{14} \times 56^{4} \times 58^{3} \times 32^{17} \times 82^{1} \times \approx 2^{196.4}$$

possibilities for the tuples at the indices $i \in \mathcal{S}$ and a probability of about 2^{-70} to find a solution for all the systems of Eq. (2)-(6). Thus, we expect about $2^{126.4}$ solutions. Using the free bits, we get for each solution $2^{3\times(64-39)} = 2^{79}$ additional solutions. In total, there are thus about $2^{205.4}$ solutions, which will be good enough for finding one conformig to the full differential (trying $2^{168.2}$ is sufficient).

To compute the $2^{168.2}$ pairs we have to find $2^{168.2-79} = 2^{89.2}$ solutions to the linear system derived from Eq. (2)-(6). Each solution requires about 2^{70} trials, thus we have to test about $2^{159.2}$ tuples. We obtain a total cost equivalent to less than $2^{168.2}$ evaluations of the compression function.

Solving the left side. Now, we have a pair of messages that verify the differential path. The probability for a random chaining value of verifying the differential path is approximately 2^{-128}. Hence, once we have the valid pair of messages, we can find a collision within about 2^{128} trials. Still for a total complexity of $2^{168.2}$, we can compute up to $2^{168.2-128} = 2^{40.2}$ collisions, whereas this should require $2^{40.2} \times 2^{256} \approx 2^{296.2}$ trials ideally.

4 Attacking HMAC-ESSENCE

HMAC [1] is a widely used construction for building message authentication codes out of hash functions. Proposed in 1996 by Bellare, Canetti, and Krawczyk, HMAC has been standardized by NIST in 2002 [8] and requirements for SHA-3 include compatibility with HMAC.

The results in §3 can directly be turned into a distinguisher for ESSENCE-256 and ESSENCE-512 when used in keyed mode, be it with an unknown prefix message, or within HMAC. We just make the standard assumption that we can query an oracle (non-adaptively) with messages, and that this returns the digests produced by the keyed ESSENCE with this message as input, for a randomly preselected key.

A distinguisher then works as follows:

```
1. Find a pair of blocks $(x, y)$ that conforms to the message part differential.
2. Repeat until a collision is found:
3.     Pick a unique prefix $m$.
4.     Query for oracle with $m\|x$ and $m\|y$.
```

Ideally 2^{256} trials are expected before a collision for ESSENCE-256, but here we'll make only 2^{64} trials in average, after a precomputation of complexity $2^{90.98}$. For ESSENCE-512, we have a complexity $2^{168.24}$ instead of 2^{512} ideally.

We can also mount an *existential forgery* attack by making one additional adaptive query:

1. Run the distinguisher above to obtain blocks m, x, y such that $m\|x$ and $m\|y$ collide by HMAC-ESSENCE.
2. Pick an arbitrary block m'.
3. Query the oracle for the MAC of $m\|x\|m'$, obtain a value z.
4. Return z as forgery of $m\|y\|m'$.

The complexity of this attack is essentially the same as that of the simple distinguisher.

5 Conclusion

We presented collision attacks on ESSENCE-256 and ESSENCE-512 of respective complexities 2^{91} and $2^{168.2}$. More precisely, these values are upper bounds on the cost of running our attacks, in terms of compression-equivalent units. Implementations of our attacks can use negligible memory, and in particular avoid expensive memory accesses. These attacks also apply to the versions of ESSENCE with 224- and 384-bit digests.

We showed a direct application of those collision attacks to the HMAC construction instantiated with ESSENCE, giving a distinguisher and an existential forgery attack with same complexity as the collision attacks.

Although far from practical, our attacks reveal significant weaknesses in the version of ESSENCE submitted to NIST. However, they do not highlight any flaw inherent to the ESSENCE design, but rather exploit the low number of rounds.

Acknowledgments

We would like to thank for their help: Anne Canteaut, Stéphane Jacob, Yann Laigle-Chapuy, Nicky Mouha, Gautham Sekar, and Fabien Viger.

References

1. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *CRYPTO*, volume 1109 of *LNCS*, pages 1–15. Springer, 1996.
2. Christophe De Cannière and Christian Rechberger. Finding SHA-1 characteristics: General results and applications. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT*, volume 4284 of *LNCS*, pages 1–20. Springer, 2006.
3. Jason Worth Martin. ESSENCE: A candidate hashing algorithm for the NIST competition. Submission to NIST, 2008.
4. Jason Worth Martin. ESSENCE: A family of cryptographic hashing algorithms. Submission to NIST, 2008.
5. Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. The rebound attack: Cryptanalysis of reduced Whirlpool and Grøstl. In Orr Dunkelman, editor, *FSE*, LNCS. Springer, 2009. to appear.
6. Nicky Mouha, Gautham Sekar, Jean-Philippe Aumasson, Thomas Peyrin, Søren S. Thomsen, Meltem Sönmez Turan, and Bart Preneel. Cryptanalysis of the ESSENCE compression function. Available online at `http://www.nickymouha.be/papers/Essence-MouhaSekar.pdf`, 2009.
7. NIST. FIPS 180-2 – secure hash standard, 2002.
8. NIST. FIPS 198 – the keyed-hash message authentication code (HMAC), 2002.
9. Ronald L. Rivest. The MD6 hash function – a proposal to NIST for SHA-3. Submission to NIST, 2008.
10. Marc Stevens, Arjen K. Lenstra, and Benne de Weger. Chosen-prefix collisions for MD5 and colliding X.509 certificates for different identities. In Moni Naor, editor, *EUROCRYPT*, volume 4515 of *LNCS*, pages 1–22. Springer, 2007.
11. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *LNCS*, pages 17–36. Springer, 2005.
12. Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *LNCS*, pages 19–35. Springer, 2005.