J. Cryptol. (2011) 24: 148–156
DOI: 10.1007/s00145-010-9063-0

Journal of CRYPTOLOGY

Cryptanalysis of the Tillich–Zémor Hash Function

Markus Grassl
Centre for Quantum Technologies (CQT), National University of Singapore, S15 #03-11, 3 Science Drive 2, Singapore 117543, Singapore
markus.grassl@nus.edu.sg

Ivana Ilić, Spyros Magliveras, and Rainer Steinwandt
Center for Cryptology and Information Security, Department of Mathematical Sciences, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA
iilic@fau.edu; spyros@fau.edu; rsteinwa@fau.edu

Communicated by Ronald Cramer

Received 31 July 2009
Online publication 16 March 2010

Abstract. At CRYPTO '94, Tillich and Zémor proposed a family of hash functions, based on computing a suitable matrix product in groups of the form $SL_2(\mathbb{F}_{2^n})$. We show how to construct collisions between palindromic bit strings of length $2n+2$ for Tillich and Zémor's construction. The approach also yields collisions for related proposals by Petit et al. from ICECS '08 and CT-RSA '09.

It seems fair to consider our attack as practical: for parameters of interest, the colliding bit strings have a length of a few hundred bits and can be found on a standard PC within seconds.

Key words. Cryptanalysis, Hash function, Collision.

1. Introduction

In their paper *Hashing with SL_2*, Tillich and Zémor [12] propose a construction for obtaining a cryptographic hash function. The proposal builds on earlier work [11,13,14] and received significant cryptanalytic interest. Various cryptanalytic results were reported [1,3,5,8,10], and in recent years, interest in Tillich and Zémor's construction has increased again. More specifically, in [9] Petit, Veyrat-Charvillon, and Quisquater suggest a construction, building on parameter choices in [12] which are considered to be secure against known attack techniques. This approach has been explored further in [8] and [4], and to the best of our knowledge no collisions have been reported for the proposed parameter choices. In [4], de Meulenaer et al. conclude that a modification of the Tillich–Zémor construction has *significant advantages over dedicated and block cipher-based hash functions including scalability, parallelism and a collision resistance reducing to the hardness of a mathematical problem.*

Our Contribution We establish a connection between the Tillich–Zémor proposal and maximal length chains in the Euclidean algorithm for polynomials over the field of two elements. After developing some results on collisions between palindromic bit strings, we combine these facts with a result of Mesirov and Sweet [6]. This reduces the identification of collisions for the Tillich–Zémor hash function to a linear algebra problem which can be easily solved with a computer algebra system. Our attack also yields collisions for the recent proposals of Petit et al., and we demonstrate the practicability of our approach by giving collisions for all specific parameter choices considered in [4,9]. Standard Merkle–Damgård strengthening does not prevent the attack, as the obtained colliding messages are of equal length.

2. Preliminaries

2.1. *Notation*

Throughout, we denote by V the collection of all bit strings, i.e., $V := \{0, 1\}^*$. Further, if $v \in V$, we denote by $|v|$ the *length* of the bit string v. If $v = b_1 \dots b_m \in V$ is of length m, we denote by $v^{\mathrm{r}} := b_m \dots b_1$ the *reversal* of v, i.e., the reflection of v which interchanges b_1 with b_m, b_2 with b_{m-1}, etc. In our attack, we will make use of *palindromes*, i.e., bit strings $v \in V$ satisfying $v = v^{\mathrm{r}}$.

Let $\mathbb{F}_{2^n}$ be a finite field represented as $\mathbb{F}_{2^n} := \mathbb{F}_2[x]/(q(x))$ with an irreducible polynomial $q(x)$ of degree n. We denote by α a root of $q(x)$, and it will be convenient to denote by G the group $SL_2(\mathbb{F}_{2^n})$, i.e., the group of 2×2 matrices of determinant 1 over $\mathbb{F}_{2^n}$. If $a_1, a_2, \dots, a_r \in G$, $\prod_{i=1}^{r} a_i$ means the 2×2 identity matrix over $\mathbb{F}_2$ when $r = 0$, and $(\prod_{i=1}^{r-1} a_i) \cdot a_r$ when $r > 0$. Eventually, we need the specific matrices

$$s_0 := \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix}, \qquad s_1 := \begin{pmatrix} \alpha & \alpha + 1 \\ 1 & 1 \end{pmatrix} \in G.$$

With the above notation, according to the Tillich–Zémor proposal in [12], hashing a bit string $v = b_1 \dots b_m \in V$ translates into applying the function $\check{h} \colon V \to G$ defined by:

$$\check{h}(b_1 \dots b_m) := s_{b_1} \cdots s_{b_m} \in G.$$

The goal of our attack is to find a *collision* for $\check{h}$, i.e., a pair $(u, v) \in V \times V$ such that $u \neq v$ and $\check{h}(u) = \check{h}(v)$.

Remark 1. At ICECS '08 [9] and CT-RSA '09 [8] Petit et al. propose vectorial and projective variants of the Tillich–Zémor construction, and in [4] de Meulenaer et al. combine ideas from [8,9]. By construction, any collision for the original Tillich–Zémor construction also yields a collision for these more recent proposals. Hence, throughout we restrict to constructing collisions for the original proposal from CRYPTO '94.

2.2. *Challenge Parameters*

Originally, Tillich and Zémor suggest a value of $n \in \{130, \dots, 170\}$ for their construction, and in view of [10] imposing n to be prime now seems to be common. The

most up-to-date suggestions we are aware of originate from [9], where the following choices of the irreducible polynomial $q(x)$ are proposed to define the underlying $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/(q(x))$:

$$\begin{aligned}
&x^{127} + x + 1,\\
&x^{251} + x^7 + x^4 + x^2 + 1,\\
&x^{509} + x^8 + x^7 + x^3 + 1,\\
&x^{1021} + x^5 + x^2 + x + 1,\\
&x^{2039} + x^{10} + x^9 + x^8 + x^7 + x^5 + x^4 + x^2 + 1.
\end{aligned}$$

We do not know if a trapdoor has been embedded in any of these polynomials (cf. [10]), but according to [4] the above five polynomials are safe with respect to the attacks in [1,5,10]. More specifically, for a collision resistance comparable to SHA-1, de Meulenaer et al. in [4] suggest to use the polynomial $x^{127} + x + 1$. For a collision resistance comparable to SHA-256 the use of $x^{251} + x^7 + x^4 + x^2 + 1$ is suggested.

The use of a field $\mathbb{F}_{2^n}$ with a prime number n close to 1024 has already been considered in [7], but as we demonstrate below, also for such a parameter choice, collisions can be identified efficiently.

3. Finding Short Palindrome Collisions

To identify collisions for the Tillich–Zémor hash function, we proceed in three steps.

3.1. *Collision Preserving Change of Generators*

First, we prove that the search for collisions for the Tillich–Zémor hash function can be transferred to the search for collisions when new, specific generators c_0 and c_1 are used. We set $c_0 = s_0$ and obtain the matrix c_1 by conjugating s_1 by s_0, i.e.,

$$c_1 := s_0^{-1} s_1 s_0.$$

Then, direct computation yields

$$c_1 = \begin{pmatrix} \alpha + 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

In particular, c_1 is symmetric and differs from c_0 only in a single entry. Based on these new generators, we define $h\colon V \to G$ by:

$$h(b_1 b_2 \ldots b_m) := c_{b_1} \cdots c_{b_m} \in G.$$

The following proposition establishes that finding collisions for the Tillich–Zémor scheme is equivalent to finding collisions for h.

Proposition 1. *Let $v, v' \in V$. Then $\breve{h}(v) = \breve{h}(v')$ if and only if $h(v) = h(v')$.*

Proof. Suppose that $v = b_1 b_2 \dots b_m$ and $v' = b'_1 b'_2 \dots b'_r$ are bit strings in V. Then, $\breve{h}(v) = \breve{h}(v')$ if and only if $s_{b_1} \cdots s_{b_m} = s_{b'_1} \cdots s_{b'_r}$. Conjugating by s_0, we see that the latter equality holds if and only if $s_0^{-1}(s_{b_1} \cdots s_{b_m})s_0 = s_0^{-1}(s_{b'_1} \cdots s_{b'_r})s_0$, which in turn is equivalent to the condition $\prod_{i=1}^m (s_0^{-1} s_{b_i} s_0) = \prod_{i=1}^r (s_0^{-1} s_{b'_i} s_0)$. Finally, because of $s_0^{-1} s_0 s_0 = s_0 = c_0$ and $s_0^{-1} s_1 s_0 = c_1$, the last equality is equivalent to $\prod_{i=1}^m c_{b_i} = \prod_{i=1}^r c_{b'_i}$. □

Remark 2. For any element $t \in G$, the conjugating map $\phi_t \colon g \mapsto g^t = t^{-1} g t$ will replace the two generators s_0, s_1 by s_0^t, s_1^t. Just as in Proposition 1 above, bit string collisions for a hash function based on s_0, s_1 will be preserved by ϕ_t to produce identical bit string collisions for the corresponding hash function based on s_0^t, s_1^t. The specific choice $t = s_0$ turns out to be very helpful for our purposes.

3.2. *Palindromic Collisions*

In this section, we work inside the group $SL_2(\mathbb{F}_2[x])$ of unimodular matrices over the polynomial ring $\mathbb{F}_2[x]$ rather than over a field $\mathbb{F}_{2^n}$. Accordingly, we define matrices $C_0, C_1 \in SL_2(\mathbb{F}_2[x])$, now with polynomial entries, as follows:

$$C_0 := \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix}, \qquad C_1 := \begin{pmatrix} x+1 & 1 \\ 1 & 0 \end{pmatrix} \in SL_2\big(\mathbb{F}_2[x]\big),$$

and we define $H \colon V \to SL_2(\mathbb{F}_2[x])$ by:

$$H(b_1 b_2 \dots b_m) := C_{b_1} \cdots C_{b_m} \in SL_2\big(\mathbb{F}_2[x]\big).$$

In other words, H is defined as h in Sect. 3.1, except that now $H(v) \in SL_2(\mathbb{F}_2[x])$. We apply H to a particular subset of elements of V, namely, the set of all palindromes in V, and obtain the following results.

Lemma 1. *Let $v \in V$ be a palindrome, and write $H(v) = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$. Then $b = c$, i.e., $H(v)$ is symmetric. Moreover, a has degree* $\deg(a) = |v|$, *and we have* $\max(\deg(b), \deg(d)) \le |v|$.[1]

Proof. The proof is by induction on the length $|v|$ of v. For $|v| \le 1$, the statement holds, as $H(v)$ is the identity matrix or C_0 or C_1, all three of which satisfy the property. For a palindrome w of length m, $H(w)$ is of the form $C_\beta \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) C_\beta$ where $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = H(v)$ for a palindrome v of length $m-2$, and where $\beta \in \{0, 1\}$. Now, direct computation yields:

$$C_\beta \begin{pmatrix} a & b \\ c & d \end{pmatrix} C_\beta = \begin{pmatrix} ax^2 + (b+c)x + d + \beta(a+b+c) & a(x+\beta) + c \\ a(x+\beta) + b & a \end{pmatrix}.$$

By the induction hypothesis, $b = c$, and the first part of the result follows. We easily check that the statement about the degrees is also true. □

[1] We use the convention $\deg(0) = -\infty$.

Next, we examine the function $\rho\colon V \to \mathbb{F}_2[x]^{2\times 2}$ defined by

$$\rho(v) := H(0v0) + H(1v1).$$

We are interested in evaluating ρ modulo a given irreducible polynomial because $\rho(v) \equiv \left(\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right) \bmod q(x)$ if and only if $h(0v0) = h(1v1)$ is indeed a collision in $SL_2(\mathbb{F}_2[x]/(q(x))) = G$. We have the following:

Proposition 2. *If $v \in V$ is a palindrome of length $|v|$, then $\rho(v) = \left(\begin{smallmatrix} a & a \\ a & 0 \end{smallmatrix}\right)$, where $a \in \mathbb{F}_2[x]$ has degree $|v|$. Moreover, a is the upper left entry of $H(v)$.*

Proof. Let $H(v) = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, and consider $\rho(v)$. Direct computation yields:

$$\rho(v) = C_0 H(v) C_0 + C_1 H(v) C_1 = \begin{pmatrix} a+b+c & a \\ a & 0 \end{pmatrix}.$$

Since v is a palindrome, by Lemma 1 we have that $b = c$, so that

$$\rho(v) = \begin{pmatrix} a & a \\ a & 0 \end{pmatrix},$$

and the claim follows with the degree statement in Lemma 1. □

For our attack we are interested in palindromes of even length, i.e., palindromes that can be written in the form vv^{r} for some $v \in V$. Here the following holds.

Proposition 3. *If $v \in V$ is a palindrome of even length, then $H(v) = \left(\begin{smallmatrix} a^2 & b \\ b & d^2 \end{smallmatrix}\right)$ for some $a, b, d \in \mathbb{F}_2[x]$.*

Proof. Let $v = ww^{\mathrm{r}}$ for some $w \in V$. The proof is by induction on $|w|$. When $|w| = 0$ the hash $H(ww^{\mathrm{r}})$ is the identity matrix and the statement holds trivially.

Suppose now we extend a string w of given length by one bit, yielding a palindrome $\beta v \beta = (\beta w)(w^{\mathrm{r}}\beta)$ with $\beta \in \{0, 1\}$. By the induction hypothesis we have that $H(v) = H(ww^{\mathrm{r}}) = \left(\begin{smallmatrix} a^2 & b \\ b & d^2 \end{smallmatrix}\right)$, so that:

$$H(\beta v \beta) = C_\beta \begin{pmatrix} a^2 & b \\ b & d^2 \end{pmatrix} C_\beta = \begin{pmatrix} (x+\beta)^2 a^2 + d^2 & (x+\beta)a^2 + b \\ (x+\beta)a^2 + b & a^2 \end{pmatrix}.$$

Consequently, both diagonal entries of $H(\beta v \beta)$ are squares, and the result follows. □

Combining Propositions 2 and 3, we obtain the following corollary.

Corollary 1. *Let $v \in V$ be a palindrome of even length. Then $\rho(v) = \left(\begin{smallmatrix} a^2 & a^2 \\ a^2 & 0 \end{smallmatrix}\right)$ for some $a \in \mathbb{F}_2[x]$ with $\deg(a) = |v|/2$. More specifically, a^2 is the upper left entry of $H(v)$.*

Further, from the proof of Proposition 3 we are able to deduce the following recurrence relation:

Corollary 2. *Let $b_n \dots b_1 b_1 \dots b_n \in V$ be a palindrome of length $2n$. Then, for $0 \le i \le n$, the square root p_i of the upper left entry of $H(b_i \dots b_1 b_1 \dots b_i)$ is given by*

$$p_i = \begin{cases} 1, & \text{if } i = 0; \\ x + b_1 + 1, & \text{if } i = 1; \\ (x + b_i)p_{i-1} + p_{i-2}, & \text{if } 1 < i \le n. \end{cases}$$

3.3. *Maximal Length Chains in the Euclidean Algorithm*

Now, for the given irreducible polynomial $q = q(x) \in \mathbb{F}_2[x]$ of degree n, used to define the field $\mathbb{F}_{2^n}$, we seek a palindrome $v \in V$ of length $2n$ such that $\rho(v) = H(0v0) + H(1v1)$ is the zero 2×2 matrix over $\mathbb{F}_2[x]/(q(x))$. In view of Corollaries 1 and 2, we can accomplish this task by determining a second polynomial $p(x) \in \mathbb{F}_2[x]$ of degree $n-1$ such that $\gcd(q(x), p(x)) = 1$ and the following holds: during the execution of the Euclidean algorithm with input $(q(x), p(x))$, the successive quotients are all of degree 1, and the degree of each remainder is only one less than the degree of the respective divisor. This will ensure a "Euclidean algorithm chain" of maximal length and adherence to the recurrence relation in Corollary 2. The existence of such a polynomial $p(x)$ follows from the subsequent result by Mesirov and Sweet [6].

Proposition 4 (Mesirov and Sweet [6]). *Given any irreducible polynomial q of degree n over $\mathbb{F}_2$, there is a sequence of polynomials $p_n, p_{n-1}, \dots, p_0$ with $p_n = q$ and $p_0 = 1$, and additionally, the degree of p_i is equal to i and $p_i \equiv p_{i-2} \bmod p_{i-1}$.*

Note that once we know a polynomial $p = p_{n-1}$ as mentioned in Proposition 4 which matches our given polynomial $p_n = q$, the Euclidean algorithm will uniquely complete the sequence $p_n, p_{n-1}, \dots, p_1, p_0 = 1$. The linear quotients $x + \beta_i$ $(i = 1, \dots, n)$ occurring in Euclid's algorithm allow us to derive the bits b_i of the palindrome in Corollary 2.

Remark 3. Note that $p_1 = x + b_1 + 1$ and therefore $b_1 = \beta_1 + 1$, while $b_i = \beta_i$ for $i > 1$, i.e., the bit β_1 has to be inverted.

Combining this in turn with Corollary 1, we obtain the desired collision

$$h(0\beta_n \dots \overline{\beta}_1 \overline{\beta}_1 \dots \beta_n 0) = h(1\beta_n \dots \overline{\beta}_1 \overline{\beta}_1 \dots \beta_n 1),$$

where $\overline{\beta}_1$ indicates the necessary inversion of β_1.

Mesirov and Sweet prove Proposition 4 by considering the field $\mathbb{F}_2((x^{-1}))$ of formal power series in x^{-1} and continued fraction expansions of elements in this field, and their proof actually contains an algorithm to compute $p = p_{n-1}$. More specifically, on input of an irreducible polynomial $q = q(x)$ of degree n, the following algorithm from [6] always produces exactly two solutions for p:

(1) Construct a matrix $A \in \mathbb{F}_2^{(n+1)\times n}$ from the $n+1$ polynomials

$$g_0 = x^0 \bmod q(x),$$
$$g_i = x^{i-1} + x^{2i-1} + x^{2i} \bmod q(x), \quad \text{for } i = 1, 2, \dots, n,$$

placing in the ith row of A the coefficients $a_{i,0}, a_{i,1}, \ldots, a_{i,n-1}$ of the polynomial $g_i = a_{i,0} + a_{i,1}x + \cdots + a_{i,n-1}x^{n-1}$.

(2) Solve the linear system $Au^{\mathrm{T}} = (1, 0, \ldots, 0, 1)^{\mathrm{T}}$ where $u = (u_1, u_2, \ldots, u_n)$.

(3) Compute $p(x)$ by multiplying $q(x)$ by $\sum_{i=1}^{n} u_i x^{-i}$ and taking only the non-negative powers of x.

Before demonstrating our attack with the challenge parameters from Sect. 2.2, we note that a palindrome collision immediately yields a different (non-palindromic) collision with bit strings of the same length. One easily checks the following.

Remark 4. Let $v \in V$ be a palindrome. Then $h(0v0) = h(1v1)$ if and only if $h(0v1) = h(1v0)$.

4. Collisions for the Challenge Parameters

With the attack just presented, it is a matter of seconds to derive collisions for the challenge parameters from Sect. 2.2. We implemented our attack in the computer algebra system Magma [2] on a standard PC. As expected, for each choice of $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/(p(x))$ we obtain two bit strings $v_1, v_2 \in \{0, 1\}^n$ with

$$h\big(0v_i v_i^{\mathrm{r}} 0\big) = h\big(1v_i v_i^{\mathrm{r}} 1\big) \quad (i = 1, 2),$$

i.e., we obtain two collisions of bit strings of length $2n + 2$. For simplicity, below we restrict to listing one bit string v_1 for each challenge parameter—the value v_2 can be obtained by reversing v_1 followed by inverting the first and last bit. To specify v_1, we use hexadecimal notation where each hexadecimal digit represents 4 bits ($\mathtt{0} - 0000, \mathtt{1} - 0001, \ldots, \mathtt{E} - 1110, \mathtt{F} - 1111$). Spaces are for readability only.

4.1. *A Collision for* $SL_2(\mathbb{F}_2[x]/(x^{127} + x + 1))$

Here we may choose

$$v_1 = \mathtt{8000\ 0000\ 0000\ 0003\ 0000\ 0000\ 0000\ 000}$$

followed by the three bit sequence 000.

4.2. *A Collision for* $SL_2(\mathbb{F}_2[x]/(x^{251} + x^7 + x^4 + x^2 + 1))$

Here we may choose

$$v_1 = \mathtt{4451\ 04E5\ 4DAB\ 26EB\ 91D3\ 5201\ 0EBD\ E579\ 54F7\ AE10\ 0959\ 713A\ EC9A\ B654\ E411\ 44}$$

followed by the three bit sequence 011.

4.3. *A Collision for* $SL_2(\mathbb{F}_2[x]/(x^{509}+x^8+x^7+x^3+1))$

Here we may choose

$v_1 =$ 10BB E68D B808 2B84 9A1C 569C 9043 7170 8D98 E3EB C923 4CF8
44F4 552C 8B49 1D45 25C4 9689 A551 7910 F996 249E BE38 CD88
7476 1049 CB51 C2C9 0EA0 80ED 8B3E E84

followed by the single bit 1.

4.4. *A Collision for* $SL_2(\mathbb{F}_2[x]/(x^{1021}+x^5+x^2+x+1))$

Here we may choose

$v_1 =$ 7EDE B9C6 F43F 3707 050D 36F7 0DA4 C665 CD36 41ED 101D F09A
258F 8C09 1176 82FF 42A1 6475 21B2 8901 143D DB01 10FE FD61
C4A9 C498 4005 0C28 F705 C7DA 6449 1D97 CDC4 9132 DF1D 0778
A185 0010 C91C A91C 35FB F844 06DD E144 048A 6C25 7134 2A17
FA0B 7444 818F 8D22 C87D C045 BC13 659D 3319 2D87 7B65 8507
0767 E17B 1CEB DBF

followed by the single bit 1.

4.5. *A Collision for* $SL_2(\mathbb{F}_2[x]/(x^{2039}+x^{10}+x^9+x^8+x^7+x^5+x^4+x^2+1))$

Here we may choose

$v_1 =$ 5DB1 31E2 BFD6 5D34 A98C 7FEF 8049 6043 1918 8835 7F23 1BEF
CF42 391A E5AF A211 BACE 74DF F1B3 4B0D 372F 1A17 4D0C FE33
6064 292E 790A 57C7 DF43 5E17 E424 49EA 3BE4 C978 3D58 1F53
ECDA DE3A 6B60 06DC 5EDD 8E80 E201 B9C8 23A7 0998 3521 A78D
8D49 1239 8700 9071 2D47 943F A369 C3C9 ABF7 7E05 FC66 FA4E
607C 0D22 433E 8368 42F9 8489 607C 0CE4 BECC 7F40 FDDF AB27
872D 8BF8 53C5 691C 1201 C338 9125 6363 CB09 5833 21CB 8827
3B00 8E02 E376 F476 C00D ACB8 F6B6 6F95 F035 783D 264F B8AF
2448 4FD0 F585 F7C7 D4A1 3CE9 284C 0D98 FE61 65D0 B1E9 D961
A59B 1FF6 5CE6 BB10 8BEB 4EB1 3885 E7EF B189 FD58 2231 3184
0D24 03EF FC63 2A59 74D7 FA8F 191B 7

followed by the three bit sequence 011.

5. Conclusion

The above discussion shows that neither the Tillich–Zémor hash function from CRYPTO '94 nor its variants from ICECS '08 and CRT-RSA '09 should be used in applications where collision resistance is essential.

Acknowledgement

We thank Aaron Meyerowitz for interesting discussions.

References

[1] K.S. Abdukhalikov, C. Kim, On the security of the hashing scheme based on SL_2, in *Fast Software Encryption—FSE '98*, ed. by S. Vaudenay. Lecture Notes in Computer Science, vol. 1372 (Springer, Berlin, 1998), pp. 93–102

[2] W. Bosma, J. Cannon, C. Playoust, The magma algebra system I: the user language. *J. Symb. Comput.* **24**, 235–265 (1997)

[3] C. Charnes, J. Pieprzyk, Attacking the SL_2 hashing scheme, in *Advances in Cryptology—ASIACRYPT '94*, ed. by J. Pieprzyk, R. Safavi-Naini. Lecture Notes in Computer Science, vol. 917 (Springer, Berlin, 1995), pp. 322–330

[4] G. de Meulenaer, C. Petit, J.-J. Quisquater, Hardware implementations of a variant of the Zémor–Tillich hash function: can a provably secure hash function be very efficient? May 2009. Available at http://eprint.iacr.org/2009/229

[5] W. Geiselmann, A note on the hash function of Tillich and Zémor, in *Cryptography and Coding*, ed. by C. Boyd. Lecture Notes in Computer Science, vol. 1025 (Springer, Berlin, 1995), pp. 257–263

[6] J.P. Mesirov, M.M. Sweet, Continued fraction expansions of rational expressions with irreducible denominators in characteristic 2. *J. Number Theory* **27**, 144–148 (1987)

[7] C. Petit, K. Lauter, J.-J. Quisquater, Cayley hashes: a class of efficient graph-based hash functions. Preprint (2007). Available at http://www.dice.ucl.ac.be/petit/files/Cayley.pdf

[8] C. Petit, J.-J. Quisquater, J.-P. Tillich, G. Zémor, Hard and easy components of collision search in the Zémor–Tillich hash function: new attacks and reduced variants with equivalent security, in *Topics in Cryptology—CT-RSA 2009*, ed. by M. Fischlin. Lecture Notes in Computer Science, vol. 5473 (Springer, Berlin, 2009), pp. 182–194

[9] C. Petit, N. Veyrat-Charvillon, J.-J. Quisquater, Efficiency and pseudo-randomness of a variant of Zémor–Tillich hash function, in *IEEE International Conference on Electronics, Circuits, and Systems ICECS 2008* (2008)

[10] R. Steinwandt, M. Grassl, W. Geiselmann, T. Beth, Weaknesses in the $SL_2(\mathbb{F}_{2^n})$ hashing scheme, in *Advances in Cryptology—CRYPTO 2000*, ed. by M. Bellare. Lecture Notes in Computer Science, vol. 1880 (Springer, Berlin, 2000), pp. 287–299

[11] J.-P. Tillich, G. Zémor, Group-theoretic hash functions, in *Algebraic Coding, First French–Israeli Workshop*, ed. by G.D. Cohen, S. Litsyn, A. Lobstein, G. Zémor. Lecture Notes in Computer Science, vol. 781 (Springer, Berlin, 1994), pp. 90–110

[12] J.-P. Tillich, G. Zémor, Hashing with SL_2, in *Advances in Cryptology—CRYPTO '94*, ed. by Y. Desmedt. Lecture Notes in Computer Science, vol. 839 (Springer, Berlin, 1994), pp. 40–49

[13] G. Zémor, Hash functions and graphs with large girths, in *Advances in Cryptology—EUROCRYPT '91*, ed. by D.W. Davies. Lecture Notes in Computer Science, vol. 547 (Springer, Berlin, 1991), pp. 508–511

[14] G. Zémor, Hash functions and Cayley graphs. *Des. Codes Cryptogr.* **4**(4), 381–394 (1994)