Preimage algorithms for the Tillich-Zémor hash function

Christophe Petit and Jean-Jacques Quisquater
Hash functions and Cayley graphs

- Hash functions
  \( H : \{0, 1\}^* \rightarrow \{0, 1\}^n \)

- “Classical”
  hash functions

- Tillich-Zémor
  hash function
Tillich-Zémor hash function

- Mathematical structure: finite group, Cayley graph
- Proposed by Tillich-Zémor at CRYPTO’94 [TZ94] following previous (broken) scheme by Zémor [Z91]
- Trapdoor attack [SGGB00]
- Attacks on particular parameters [SGGB00,CP94,AK98]
- Until 13 months ago, best generic attacks were asymptotically inefficient [PQTZ08]
Tillich-Zémor hash function

- August’09: very efficient collision attack by Grassl, Illic, Magliveras, Steinwandt [GIMS09]
- This paper: preimage algorithms (also very efficient)
Outline

Introduction

Tillich-Zémor hash function

Grassl et al.’s collision attack

Preimage algorithms

Conclusion
Outline

Introduction

Tillich-Zémor hash function

Grassl et al.'s collision attack

Preimage algorithms

Conclusion
Tillich-Zémor hash function

- $p \in \mathbb{F}_2[X]$ irreducible of degree $n$
  
  $K = \mathbb{F}_2[X]/(p(X)) \approx \mathbb{F}_{2^n}$

- Group $G = SL(2, K)$
  
  Generators $S = \{ A_0 = (\begin{smallmatrix} x & 1 \\ 1 & 0 \end{smallmatrix}), A_1 = (\begin{smallmatrix} x & x+1 \\ 1 & 1 \end{smallmatrix}) \}$

- Message $m = m_1...m_N \in \{0, 1\}^N$

  $$H(m_1 m_2...m_N) := A_{m_1} A_{m_2}...A_{m_N} \mod p(X)$$
Hard (?) problems

- **Balance problem**: \((\Leftrightarrow \text{collisions})\)
  Given \(G\) and \(S = \{s_0, \ldots, s_{k-1}\} \subset G\), find two short products \(\prod s_{m_i} = \prod s_{m'_i}\).

- **Representation problem**: \((\Rightarrow \text{2nd preimages})\)
  Given \(G\) and \(S = \{s_0, \ldots, s_{k-1}\} \subset G\), find a short product \(\prod s_{m_i} = 1\).

- **Factorization problem**: \((\Leftrightarrow \text{preimages})\)
  Given \(G, g \in G\) and \(S = \{s_0, \ldots, s_{k-1}\} \subset G\), find a short product \(\prod s_{m_i} = g\)
Outline

Introduction

Tillich-Zémor hash function

Grassl et al.’s collision attack

Preimage algorithms

Conclusion
Changing the generators

Let $A'_0 := A_0^{-1}A_0A_0 = \begin{pmatrix} X & 1 \\ 1 & 0 \end{pmatrix}$,
Let $A'_1 := A_0^{-1}A_1A_0 = \begin{pmatrix} X+1 & 1 \\ 1 & 0 \end{pmatrix}$

Let $H'$ be $H$ but replacing $A_0, A_1$ by $A'_0, A'_1$
$$H'(m) = A_0^{-1}H(m)A_0$$

Collision for $H' \iff$ collision for $H$
Preimage of $g$ for $H' \iff$ preimage of $A_0gA_0^{-1}$ for $H$

! Notation : we write $A_0, A_1, H$ instead of $A'_0, A'_1, H'$
A_0 = \begin{pmatrix} X & 1 \\ 1 & 0 \end{pmatrix} \text{ and } A_1 = \begin{pmatrix} X+1 & 1 \\ 1 & 0 \end{pmatrix} \text{ are “Euclidean algorithm matrices”}

a_{i-1} = q_i a_i + a_{i+1} \iff \begin{pmatrix} a_i & a_{i-1} \end{pmatrix} = \begin{pmatrix} a_{i-1} & a_{i-2} \end{pmatrix} \begin{pmatrix} q_i & 1 \end{pmatrix}

Let h be H “without modular reductions”

h(m_1...m_n) := A_{m_1}...A_{m_N}

\begin{pmatrix} a & b \\ c & d \end{pmatrix} = h(m) \Rightarrow \text{the Euclidean algorithm applied to } (a, b) \text{ only produces quotients } X \text{ and } X + 1
Mesirov and Sweet’s algorithm

- **Theorem [MS87]**: for any irreducible \( a \in \mathbb{F}_2[X] \), there exists \( b \in \mathbb{F}_2[X] \) such that all quotients obtained by applying the Euclidean algorithm to \((a, b)\) belong to \( \{X, X + 1\} \)

- The proof is constructive
Building the collision

- Let $p$ be the polynomial defining the field in TZ hash function
- Apply [MS87] to $a = p$: we obtain $b$ and a message $m = m_1...m_N$ such that $H(m) = \begin{pmatrix} 0 & b \\ c & d \end{pmatrix}$
- Swap the first bit
  
  $H(\bar{m}_1m_2...m_N) = \begin{pmatrix} c & b+d \\ c & d \end{pmatrix}$

- Build the palindrome $\tilde{m} = m_N...m_2\bar{m}_1\bar{m}_1m_2...m_N$
  
  $H(\tilde{m}) = \begin{pmatrix} 0 & 1 \\ 1 & b^2 \end{pmatrix}$

- Observe collision
  
  $A_0H(\tilde{m})A_0 = A_1H(\tilde{m})A_1$
Outline

Introduction

Tillich-Zémor hash function

Grassl et al.’s collision attack

Preimage algorithms

Conclusion
Second preimages

- Apply [MS87] to $a = p$: we obtain a message $m = m_1...m_N$ such that $H(m) = \begin{pmatrix} 0 & b \\ c & d \end{pmatrix}$
- Build the palindrome $\tilde{m} = m_N...m_2 \tilde{m}_1 \tilde{m}_1 m_2...m_N$
- Observe
  - $H(0\tilde{m}) = \begin{pmatrix} 1 & X+b^2 \\ 0 & 1 \end{pmatrix}$ and $H(\tilde{m}0) = \begin{pmatrix} 1 & 0 \\ X+b^2 & 1 \end{pmatrix}$
  - Both matrices have order 2
    $\Rightarrow H(0\tilde{m}0\tilde{m}) = H(\tilde{m}0\tilde{m}0) = I$
- **Preimage of $I$** $\Rightarrow$ **second preimages** for any message $H(m_0) = I$ $\Rightarrow$ $H(mm_0) = H(m_0m) = H(m)$
Preimage algorithm

- **Precompute** preimages of \( \begin{pmatrix} 0 & b_i \\ c_i & d_i \end{pmatrix} \)
such that the set \( \{b_i^2 + X\} \) is a basis of \( \mathbb{F}_{2^n} / \mathbb{F}_2 \)

- Let \( m = m_1 \ldots m_N \) such that \( H(m) = \begin{pmatrix} 0 & b \\ c & d \end{pmatrix} \).
  Then \( H(\tilde{m}0) = \begin{pmatrix} 1 \\ X+b^2 & 1 \end{pmatrix} \) and \( H(0\tilde{m}) = \begin{pmatrix} 1 \\ X+b^2 & 1 \end{pmatrix} \)

- The “red matrices” belong to **Abelian subgroups**
  \( \left( \sum_{\alpha_i} \begin{pmatrix} 1 \\ \alpha_i & 1 \end{pmatrix} \right) \prod \left( \begin{pmatrix} 1 \\ \alpha_i & 1 \end{pmatrix} \right) \) and \( \left( \begin{pmatrix} 1 \\ \sum_{\beta_i} & 1 \end{pmatrix} \right) \prod \left( \begin{pmatrix} 1 \\ \beta_i & 1 \end{pmatrix} \right) \)
  Write any \( \alpha, \beta \) in the basis \( \{b_i^2 + X\} \) using linear algebra

- Any matrix can be written as
  \( \begin{pmatrix} A & B \\ C & D \end{pmatrix} = (X \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^\delta (1 \begin{pmatrix} 1 & 0 \\ \alpha & 1 \end{pmatrix} (X \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} (1 \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix} (X \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix})^3 (1 \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix}), \delta \in \{0,1\}. \)
First precomputing algorithm

- **Goal**: obtain \( n \) messages hashing to matrices \( \begin{pmatrix} 0 & b_i \\ c_i & d_i \end{pmatrix} \) such that the set \( \{ b_i^2 + X \} \) is a basis of \( \mathbb{F}_{2^n}/\mathbb{F}_2 \)

  Applying [MS87] to \( a = p \) we obtain one such message

- **Idea**: apply [MS87] to \( a = pp'_i \) where \( p'_i \) small degree

- **Issue**: [MS87] requires \( a \) irreducible
First precomputing algorithm

- **We extend** [MS87]: Let $p, p'$ be nonlinear irreducible polynomials and let $a = pp'$. If

$$\deg \left( \left[ X(X+1)p' \right]^{-1} \mod p \right) \leq \deg(p) - 2$$

then the Mesirov-Sweet’s algorithm provides $b$ such that all quotients computed by the Euclidean algorithm applied to $(a, b)$ belong to $\{X, X + 1\}$

- **Heuristic arguments + experiments**:
  - Small $\deg(p_i')$ suffice
  - Preimages of length $O(n^2)$ for TZ
  - Probabilistic time $O(n^4)$
Second precomputing algorithm

- **Goal**: obtain \( n \) messages hashing to matrices \( \begin{pmatrix} 0 & b_i \\ c_i & d_i \end{pmatrix} \) such that the set \( \{ b_i^2 + X \} \) is a basis of \( \mathbb{F}_{2^n} / \mathbb{F}_2 \)

Applying [MS87] to \( a = p \) we obtain one such message \( m_1 \)

- **Idea**: build those messages recursively
  - Define \( m_i := m_{i-1}0m_1 \)
  - **We prove** that \( H(m_i) = \begin{pmatrix} 0 & b_i^i \\ c_i & d_i \end{pmatrix} \) for some \( c_i, d_i \)

- Do the elements \( b_i^2 + X \) generate a basis of \( \mathbb{F}_{2^n} / \mathbb{F}_2 \)?
Second precomputing algorithm

- **We prove**: If the minimal polynomial of $b_1$ has degree $n$, then we can extract a basis from \{ $b_i^2 + X, i = 1, \ldots, 2n$ \}

- When $n$ is prime: always succeeds
  - Preimage of length $O(n^3)$ for TZ
  - Deterministic time $O(n^3)$

- When $n$ is not prime
  - Succeeds with very high probability, same complexities (the analysis is partially heuristic)
  - Always succeeds in practice
  - (Other attacks exist)
Outline

Introduction

Tillich-Zémor hash function

Grassl et al.'s collision attack

Preimage algorithms

Conclusion
Preimage algorithms for TZ hash function

- Preimages in time $O(n^3)$ given some precomputation
- First precomputing algorithm:
  - Preimages of length $O(n^2)$ in probabilistic time $O(n^4)$
- Second precomputing algorithm:
  - Preimages of length $O(n^3)$ in deterministic time $O(n^3)$
  - Full proof when $n$ is prime
- The case $n$ prime proves a conjecture of Babai [BS92] for those particular parameters
Hash functions and Cayley graphs: the end of the story?

- Similar functions have been broken as well (Zémor, LPS, Morgenstern)
- However, all these functions used very special parameters in a sense
- Strong connections with well-known problems in graph theory and group theory, with many applications in computer science (expander graphs...)
- Next challenge: $SL(2, F_{2^n})$ with $A_0 = \begin{pmatrix} t_0 & 1 \\ 1 & 0 \end{pmatrix}$, $A_1 = \begin{pmatrix} t_1 & 1 \\ 1 & 0 \end{pmatrix}$ and $t_0 + t_1 \neq 1$
References

- [TZ94] JP Tillich & G Zémor, *Group-theoretic hash functions*
- [Z91] G Zémor, *Hash functions and graphs with large girths*
- [SGGB00] R Steinwandt, M Grassl, W Geiselmann, T Beth, *Weaknesses in the $SL_2(F_{2^n})$ Hashing Scheme*
- [CP94] C Charnes, J Pieprzyk, *Attacking the SL2 hashing scheme*
- [AK98] K Abdukhalikov, C Kim, *On the security of the hashing scheme based on SL2*
References

- [GIMS09] M Grassl, I Ilic, S Magliveras, R Steinwandt, *Cryptanalysis of the Tillich-Zémor hash function*
- [MS87] JP Mesirov, MM Sweet, *Continued fraction expansions of rational expressions with irreducible denominators in characteristic 2*
- [BS92] L Babai, A Seress, *On the diameter of permutation groups*
References

- [CGL09] D Charles, E Goren, K Lauter, Cryptographic hash functions from expander graphs
- [PLQ07] C Petit, K Lauter, JJ Quisquater, Cayley Hashes: A Class of Efficient Graph-based Hash Functions
- [LPS88] A Lubotzky, R Phillips, P Sarnak, Ramanujan Graphs
- [TZ08] JP Tillich, G Zémor, Collisions for the LPS Expander Graph Hash Function
- [PLQ08] C Petit, K Lauter, JJ Quisquater, Full Cryptanalysis of LPS and Morgenstern Hash Functions