On expander hash functions

Christophe Petit

Hash functions: applications

- Cryptographic hash functions are compared to *Swiss army knifes* because of their multiple tools and uses

Hash functions: properties

- **Compressing** functions: (key, message) $\longrightarrow$ hash value

$$H : \{0,1\}^\kappa \times \{0,1\}^* \longrightarrow \{0,1\}^\lambda$$

- **Main properties:**
  - Collision resistance
  - Preimage resistance
  - Second preimage resistance

Hash functions: properties

- **Compressing** functions: (key, message) $\longrightarrow$ hash value

$$H : \{0,1\}^{\kappa} \times \{0,1\}^{*} \longrightarrow \{0,1\}^{\lambda}$$

- **Main properties:**
  - Collision resistance
  - Preimage resistance
  - Second preimage resistance
- **Further properties:**
  - XOR resistance, ADD resistance, non-multiplicativity ...
  - PRF, random oracle, ...

Hash functions: constructions

- **Standards** exist (SHA)...

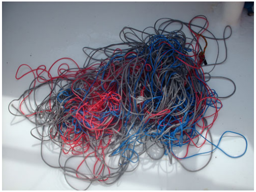
  ... but they are being **broken** !

Hash functions: constructions

- **Standards** exist (SHA)...

  ... but they are being **broken** !

- Current hash functions look like this:

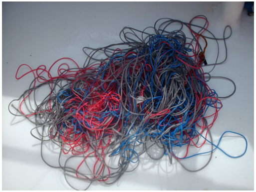


Hash functions: constructions

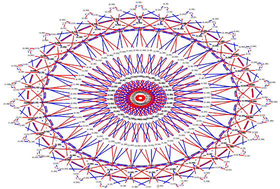
- **Standards** exist (SHA)...

  ... but they are being **broken** !

- Current hash functions look like this:



- While expander hashes look like this:



Thesis' objectives

- Study the **security** of expander hashes
  - Generic security Ch.4
  - Security of particular constructions Ch.5,6,7,D
  - Malleability Ch.8

Thesis' objectives

- Study the **security** of expander hashes
  - Generic security Ch.4
  - Security of particular constructions Ch.5,6,7,D
  - Malleability Ch.8
- Study the **efficiency** of expander hashes
  - Hardware and software efficiency of particular constructions Ch.4,7,9
  - Improved algorithms Ch.4,9,C
  - Parallelism Ch.4,8,9

Thesis' objectives

- Study the **security** of expander hashes
  - Generic security Ch.4
  - Security of particular constructions Ch.5,6,7,D
  - Malleability Ch.8
- Study the **efficiency** of expander hashes
  - Hardware and software efficiency of particular constructions Ch.4,7,9
  - Improved algorithms Ch.4,9,C
  - Parallelism Ch.4,8,9
- Study the **applications** of expander hashes
  - Model and use malleability Ch.8
  - Remove malleability Ch.9

Outline

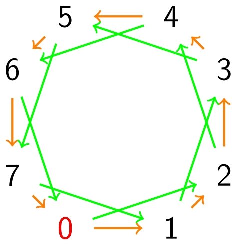
- Introduction
- Generic construction and attacks
- Known instances
  - Overview
  - Focus 1 : Cryptanalysis of LPS and Morgenstern hash functions
  - Focus 2 : Vectorial and projective Zémor-Tillich
- Perspectives
- Conclusion

Outline

- Introduction
- Generic construction and attacks
- Known instances
  - Overview
  - Focus 1 : Cryptanalysis of LPS and Morgenstern hash functions
  - Focus 2 : Vectorial and projective Zémor-Tillich
- Perspectives
- Conclusion

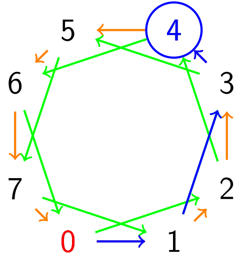
Expander hashes

- For a k-regular **directed** graph,
  color the edges with k colors, choose an initial vertex



Expander hashes

- For a k-regular **directed** graph,
  color the edges with k colors, choose an initial vertex

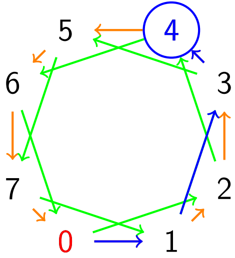


$m = 101$

- Given a message m, decompose it into k-digits $m = m_1 m_2 \ldots m_\mu$
- The digits fix a walk in the graph according to the coloring

Expander hashes

- For a k-regular **directed** graph,
  color the edges with k colors, choose an initial vertex



$m = 101$

- Given a message m, decompose it into k-digits $m = m_1 m_2 \ldots m_\mu$
- The digits fix a walk in the graph according to the coloring
- For undirected graphs: forbid backtracking

Cayley hashes

- Cayley graphs: graphs built from groups
- Cayley hashes: expander hashes built from Cayley graphs

  Example for $S = \{s_0, s_1\}$ and initial vertex 1:

$$H(11001) = s_1 s_1 s_0 s_0 s_1$$

Cayley hashes

- Cayley graphs: graphs built from groups
- Cayley hashes: expander hashes built from Cayley graphs

  Example for $S = \{s_0, s_1\}$ and initial vertex 1:

$$H(11001) = s_1 s_1 s_0 s_0 s_1$$

- Simplifies definition and study
- Allows parallelism: $H(m_1 || m_2) = H(m_1) \cdot H(m_2)$
  where $\cdot$ is group law

Cayley hashes: security properties

hash properties	**graph properties**	**group properties**
collision resistance	cycle / two-paths problem	representation / balance problem
preimage resistance	path-finding problem	factorization problem
output distribution	expanding properties	Kazhdan constant
minimal collision "distance"	girth	

Representation problem

- Given a group G and $S = \{s_1, \ldots s_k\} \subset G$, find a product in *reduced form*

$$\prod_{1 \leq i \leq N} s_{\theta(i)}^{e_i} = 1$$

where $e_i \in \mathbb{Z}^+$, $\theta : \{1, \ldots N\} \rightarrow \{1 \ldots k\}$ and $\sum e_i$ is "small".

By *reduced form*, we mean that for each i, $s_{\theta(i+1)} \neq s_{\theta(i)}, s_{\theta(i)}^{-1}$

Representation problem

- Given a group G and $S = \{s_1, ...s_k\} \subset G$,
  find a product in *reduced form*

$$\prod_{1 \leq i \leq N} s_{\theta(i)}^{e_i} = 1$$

  where $e_i \in \mathbb{Z}^+$, $\theta : \{1, ...N\} \to \{1...k\}$ and $\sum e_i$ is "small".

  By *reduced form*, we mean that for each i, $s_{\theta(i+1)} \neq s_{\theta(i)}, s_{\theta(i)}^{-1}$

- The hardness of this problem highly depends on G and S !
  Of course, G must be non-Abelian

Attacks on expander hashes

- Birthday attack in time $2^{\lambda/2}$
- Exhaustive search in time 2^{λ}

Attacks on expander hashes

- Birthday attack in time $2^{\lambda/2}$
- Exhaustive search in time 2^{λ}
- "Meet-in-the-middle" preimage attacks
  - Preimages in time $2^{\lambda/2}$
  - Because each step is invertible

Attacks on expander hashes

- Birthday attack in time $2^{\lambda/2}$
- Exhaustive search in time 2^{λ}
- "Meet-in-the-middle" preimage attacks
  - Preimages in time $2^{\lambda/2}$
  - Because each step is invertible
- Multicollision attacks
  - t-collisions in time $\log_2 t 2^{\lambda/2}$ [Joux04]
  - Because of iterative structure

Attacks on expander hashes

- Birthday attack in time $2^{\lambda/2}$
- Exhaustive search in time 2^{λ}
- "Meet-in-the-middle" preimage attacks
  - Preimages in time $2^{\lambda/2}$
  - Because each step is invertible
- Multicollision attacks
  - t-collisions in time $\log_2 t 2^{\lambda/2}$ [Joux04]
  - Because of iterative structure
- Trapdoor attacks
  - Choose initial vertex and/or graph parameters to help collision search

Attacks on expander hashes

- Subgroup attacks on Cayley hashes

Attacks on expander hashes

- Subgroup attacks on Cayley hashes
- Malleability
  - Cayley hashes: for any m, m'
    $$H(m||m') = H(m) \cdot H(m')$$
  - In general: given $H(m)$ and m', easy to compute $H(m||m')$...
    ... even if m itself cannot be computed from $H(m)$!

Outline

- Introduction
- Generic construction and attacks
- Known instances
  - Overview
  - Focus 1 : Cryptanalysis of LPS and Morgenstern hash functions
  - Focus 2 : Vectorial and projective Zémor-Tillich
- Perspectives
- Conclusion

Outline

- Introduction
- Generic construction and attacks
- Known instances
  - Overview
  - Focus 1 : Cryptanalysis of LPS and Morgenstern hash functions
  - Focus 2 : Vectorial and projective Zémor-Tillich
- Perspectives
- Conclusion

Zémor's first proposal

- Construction [Zém91,Zém94]:
  Cayley hash with $G = SL(2, \mathbb{F}_p)$, $v_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and

$$S = \{s_0 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, s_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}\}$$

Zémor's first proposal

- Construction [Zém91,Zém94]:
  Cayley hash with $G = SL(2, \mathbb{F}_p)$, $v_0 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ and

$$S = \left\{ s_0 = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right), s_1 = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right) \right\}$$

- Cryptanalysis [TZ93]:
  - Collision "lifting" attack:
    lift the representation problem from $SL(2, \mathbb{F}_p)$ to $SL(2, \mathbb{Z}^+)$
  - Also a preimage attack
  - This function is broken !

The Zémor-Tillich hash function (ZT)

- Construction : [TZ94]
  - Let $P_n(X) \in \mathbb{F}_2[X]$ irreducible, degree $n \in [130, 170]$
    Let $K = \mathbb{F}_2[X]/(P_n(X))$

The Zémor-Tillich hash function (ZT)

- Construction : [TZ94]
  - Let $P_n(X) \in \mathbb{F}_2[X]$ irreducible, degree $n \in [130, 170]$
    Let $K = \mathbb{F}_2[X]/(P_n(X))$
  - Take $G = SL(2, K)$,

$$S = \left\{ s_0 = \begin{pmatrix} X & 1 \\ 1 & 0 \end{pmatrix}, s_1 = \begin{pmatrix} X & X+1 \\ 1 & 1 \end{pmatrix} \right\}$$

    Take $v_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

The Zémor-Tillich hash function (ZT)

- Construction : [TZ94]
  - Let $P_n(X) \in \mathbb{F}_2[X]$ irreducible, degree $n \in [130, 170]$
    Let $K = \mathbb{F}_2[X]/(P_n(X))$
  - Take $G = SL(2, K)$,

$$S = \left\{ s_0 = \begin{pmatrix} X & 1 \\ 1 & 0 \end{pmatrix}, s_1 = \begin{pmatrix} X & X+1 \\ 1 & 1 \end{pmatrix} \right\}$$

    Take $v_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

- Reasonably efficient
  - 3 additions / bit
  - Binary fields arithmetic

The Zémor-Tillich hash function (ZT)

- Partial cryptanalysis...
  - Generic issues of Cayley hashes
  - Invertibility for short messages [SGGB00]
  - Trapdoor attacks on $P_n(X)$ [CP94,AK98,SGGB00]
  - Projection to finite fields [G96]
  - Subgroup attacks for composite n [SGGB00]
  - Generic collision and preimage subgroup attacks in time $2^{n/2}$ (instead of $2^{3n/2}$ and 2^{3n} for birthday and exhaustive) [PQTZ09]

The Zémor-Tillich hash function (ZT)

- Partial cryptanalysis...
  - Generic issues of Cayley hashes
  - Invertibility for short messages [SGGB00]
  - Trapdoor attacks on $P_n(X)$ [CP94,AK98,SGGB00]
  - Projection to finite fields [G96]
  - Subgroup attacks for composite n [SGGB00]
  - Generic collision and preimage subgroup attacks in time $2^{n/2}$ (instead of $2^{3n/2}$ and 2^{3n} for birthday and exhaustive) [PQTZ09]

... but fundamentally unbroken since 1994

The Zémor-Tillich hash function (ZT)

- Partial cryptanalysis...
  - Generic issues of Cayley hashes
  - Invertibility for short messages [SGGB00]
  - Trapdoor attacks on $P_n(X)$ [CP94,AK98,SGGB00]
  - Projection to finite fields [G96]
  - Subgroup attacks for composite n [SGGB00]
  - Generic collision and preimage subgroup attacks in time $2^{n/2}$ (instead of $2^{3n/2}$ and 2^{3n} for birthday and exhaustive) [PQTZ09]

  ... but fundamentally unbroken since 1994

- Vectorial and projective variants [PQTZ09]

LPS and Morgenstern hash functions

- LPS hash function: use LPS Ramanujan graphs [LPS88, CGL07]
- Reasonably efficient (a few additions / step)
- Extension to Morgenstern Ramanujan graphs [PLQ07]

LPS and Morgenstern hash functions

- LPS hash function: use LPS Ramanujan graphs [LPS88, CGL07]
- Reasonably efficient (a few additions / step)
- Extension to Morgenstern Ramanujan graphs [PLQ07]
- Cryptanalysis
  - Collision lifting attack [TZ08]
  - Extension to preimage attack [PLQ08]
  - Extension to collision and preimage for Morgenstern hash [PLQ08]

LPS and Morgenstern hash functions

- LPS hash function: use LPS Ramanujan graphs [LPS88, CGL07]
- Reasonably efficient (a few additions / step)
- Extension to Morgenstern Ramanujan graphs [PLQ07]
- Cryptanalysis
  - Collision lifting attack [TZ08]
  - Extension to preimage attack [PLQ08]
  - Extension to collision and preimage for Morgenstern hash [PLQ08]
- Both functions repaired by modifying S

The Pizer hash function

- Use Pizer's Ramanujan graphs [P90,CGL07]
  - (Not Cayley)
  - Vertices are supersingular elliptic curves
  - Edges are isogenies of fixed degree

The Pizer hash function

- Use Pizer's Ramanujan graphs [P90,CGL07]
  - (Not Cayley)
  - Vertices are supersingular elliptic curves
  - Edges are isogenies of fixed degree
- Not broken so far, but
  - Much slower than previous instances
  - No guarantee on the girth in general

Outline

- Introduction
- Generic construction and attacks
- Known instances
  - Overview
  - Focus 1 : Cryptanalysis of LPS and Morgenstern hash functions
  - Focus 2 : Vectorial and projective Zémor-Tillich
- Perspectives
- Conclusion

LPS hash function

- Construction: use LPS Ramanujan graphs [LPS88, CGL07]
  - Let l small prime, p large prime, $p \equiv l \equiv 1 \bmod 4$, $\left(\frac{l}{p}\right) = 1$
    Let $\mathbf{i}$ such that $\mathbf{i}^2 = -1 \bmod p$
  - Let $G = PSL(2, \mathbb{F}_p)$,
    Let $S = \{s_j, j = 1 \ldots l+1\}$, where

$$s_j = \begin{pmatrix} \alpha_j + \mathbf{i}\beta_j & \gamma_j + \mathbf{i}\delta_j \\ -\gamma_j + \mathbf{i}\delta_j & \alpha_j - \mathbf{i}\beta_j \end{pmatrix}, \qquad j = 0, \ldots, l;$$

    and $(\alpha_j, \beta_j, \gamma_j, \delta_j)$ are all the integer solutions of $\alpha^2 + \beta^2 + \gamma^2 + \delta^2 = l$, with $\alpha > 0$ and β, γ, δ

Collisions for LPS Hash [TZ08]

- Idea of Tillich-Zémor attack : **lift the representation problem** from $PSL(2,\mathbb{F}_p)$ to $\Omega \subset SL(2,\mathbb{Z}[i])$:

$$\xleftarrow[\text{mod } p]{}$$

$$\begin{array}{rcl}
\mathbf{i}^2 = -1 & \rightarrow & i^2 = -1 \\
\mathbb{F}_p & \rightarrow & \mathbb{Z}[i] \\
PSL(2,\mathbb{F}_p) & \rightarrow & \Omega \subset SL(2,\mathbb{Z}[i]) \\
\begin{pmatrix} g_{0,j} + \mathbf{i}g_{1,j} & g_{2,j} + \mathbf{i}g_{3,j} \\ -g_{2,j} + \mathbf{i}g_{3,j} & g_{0,j} - \mathbf{i}g_{1,j} \end{pmatrix} & \rightarrow & \begin{pmatrix} g_{0,j} + ig_{1,j} & g_{2,j} + ig_{3,j} \\ -g_{2,j} + ig_{3,j} & g_{0,j} - ig_{1,j} \end{pmatrix}
\end{array}$$

Collisions for LPS Hash [TZ08]

- Idea of Tillich-Zémor attack : **lift the representation problem** from $PSL(2,\mathbb{F}_p)$ to $\Omega \subset SL(2,\mathbb{Z}[i])$:

$$\xleftarrow[\text{mod } p]{}$$

$$\begin{array}{rcl}
\mathbf{i}^2 = -1 & \rightarrow & i^2 = -1 \\
\mathbb{F}_p & \rightarrow & \mathbb{Z}[i] \\
PSL(2,\mathbb{F}_p) & \rightarrow & \Omega \subset SL(2,\mathbb{Z}[i]) \\
\begin{pmatrix} g_{0,j}+\mathbf{i}g_{1,j} & g_{2,j}+\mathbf{i}g_{3,j} \\ -g_{2,j}+\mathbf{i}g_{3,j} & g_{0,j}-\mathbf{i}g_{1,j} \end{pmatrix} & \rightarrow & \begin{pmatrix} g_{0,j}+ig_{1,j} & g_{2,j}+ig_{3,j} \\ -g_{2,j}+ig_{3,j} & g_{0,j}-ig_{1,j} \end{pmatrix} \\
\lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \in PSL(2,\mathbb{F}_p) & \rightarrow & \begin{pmatrix} a+bi & c+di \\ -c+di & a-bi \end{pmatrix} \in \Omega
\end{array}$$

The lifted set Ω

- Properties required of Ω:
  - $\Omega \subset SL(2, \mathbb{Z}[i])$
  - A large proportion (actually all) of $m \in \Omega$ has a unique factorization in the lifted generators
  - This factorization is easily computed
  - We deduce a factorization in $PSL(2, \mathbb{F}_p)$ by reduction modulo p

The lifted set Ω

- Properties required of Ω:
  - $\Omega \subset SL(2, \mathbb{Z}[i])$
  - A large proportion (actually all) of $m \in \Omega$ has a unique factorization in the lifted generators
  - This factorization is easily computed
  - We deduce a factorization in $PSL(2, \mathbb{F}_p)$ by reduction modulo p

- For the Ω chosen in [TZ08], finding $m \in \Omega$ mainly amounts to finding $\lambda, w, x, y, z, e \in \mathbb{Z}$ solving

$$(\lambda + wp)^2 + 4(xp)^2 + 4(yp)^2 + 4(zp)^2 = l^e$$

The lifted set Ω

- Properties required of Ω:
  - $\Omega \subset SL(2, \mathbb{Z}[i])$
  - A large proportion (actually all) of $m \in \Omega$ has a unique factorization in the lifted generators
  - This factorization is easily computed
  - We deduce a factorization in $PSL(2, \mathbb{F}_p)$ by reduction modulo p

- For the Ω chosen in [TZ08], finding $m \in \Omega$ mainly amounts to finding $\lambda, w, x, y, z, e \in \mathbb{Z}$ solving

$$(\lambda + wp)^2 + 4(xp)^2 + 4(yp)^2 + 4(zp)^2 = l^e$$

Fix $\lambda + wp$, ...

Preimages for LPS Hash [PLQ08]

- With the **same lifting strategy**, finding a preimage to a matrix $M = \begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \begin{pmatrix} A+B\mathbf{i} & C+D\mathbf{i} \\ -C+D\mathbf{i} & A-B\mathbf{i} \end{pmatrix}$ now amounts to solving

$$(A\lambda + wp)^2 + (B\lambda + xp)^2 + (C\lambda + yp)^2 + (D\lambda + zp)^2 = l^{2k}$$

Preimages for LPS Hash [PLQ08]

- With the **same lifting strategy**, finding a preimage to a matrix $M = \begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \begin{pmatrix} A+B\mathbf{i} & C+D\mathbf{i} \\ -C+D\mathbf{i} & A-B\mathbf{i} \end{pmatrix}$ now amounts to solving

$$(A\lambda + wp)^2 + (B\lambda + xp)^2 + (C\lambda + yp)^2 + (D\lambda + zp)^2 = l^{2k}$$

- Trivial extension does not work:
  - Fixing $A\lambda + wp$ to satisfy the equation modulo p...
  - ... does not permit simplifying by p^2 because of the term $2p(wA + xB + yC + zD)\lambda$.
  - Hence the coefficients of degree-2 terms are huge (at least p)...
  - ... so the resulting equation in x, y, z is most likely to have no solution.

Preimages for LPS Hash [PLQ08]

- Sketch of our solution:
  - Solve the preimage problem for *diagonal matrices*
    $(A\lambda + wp)^2 + (B\lambda + xp)^2 + (yp)^2 + (zp)^2 = l^{2k}$
  - Decompose any matrix as a product of *diagonal matrices* and *graph generators*
    $\begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \lambda \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix}$

Preimages for LPS Hash [PLQ08]

- Sketch of our solution:
  - Solve the preimage problem for *diagonal matrices*
    $(A\lambda + wp)^2 + (B\lambda + xp)^2 + (yp)^2 + (zp)^2 = l^{2k}$
  - Decompose any matrix as a product of *diagonal matrices* and *graph generators*
    $\begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \lambda \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix}$
- See [PLQ08] or Ch.6 for details

Cryptanalysis of Morgenstern Hash [PLQ08]

- LPS graphs for odd primes l
  Morgenstern graphs for l^k, including $l = 2$ [M1994]
  Morgenstern hashes use $l = 2$ [PLQ07]

Cryptanalysis of Morgenstern Hash [PLQ08]

- LPS graphs for odd primes l
  Morgenstern graphs for l^k, including $l = 2$ [M1994]
  Morgenstern hashes use $l = 2$ [PLQ07]

- Lifting attack from $SL(2, \mathbb{F}_{2^n})$ to $\Omega \in SL(2, \mathbb{A})$ where $\mathbb{A} = \mathbb{F}_2[x, y]/(y^2 + y + 1)$
- The resulting equations differ, but can be solved with the same techniques extended to polynomials

Cryptanalysis of Morgenstern Hash [PLQ08]

- LPS graphs for odd primes l
  Morgenstern graphs for l^k, including $l = 2$ [M1994]
  Morgenstern hashes use $l = 2$ [PLQ07]

- Lifting attack from $SL(2, \mathbb{F}_{2^n})$ to $\Omega \in SL(2, \mathbb{A})$ where $\mathbb{A} = \mathbb{F}_2[x, y]/(y^2 + y + 1)$
- The resulting equations differ, but can be solved with the same techniques extended to polynomials
- See [PLQ08] or Ch.6 for details

Outline

- Introduction
- Generic construction and attacks
- Known instances
  - Overview
  - Focus 1 : Cryptanalysis of LPS and Morgenstern hash functions
  - Focus 2 : Vectorial and projective Zémor-Tillich
- Perspectives
- Conclusion

The Zémor-Tillich hash function (ZT)

- Recall:
  - ZT is a Cayley hash with $G = SL(2, \mathbb{F}_{2^n})$, $v_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and

$$S = \left\{ s_0 = \begin{pmatrix} X & 1 \\ 1 & 0 \end{pmatrix}, s_1 = \begin{pmatrix} X & X+1 \\ 1 & 1 \end{pmatrix} \right\}$$

  - Generic collision and preimage subgroup attacks in time $2^{n/2}$ (instead of $2^{3n/2}$ and 2^{3n} for birthday and exhaustive) [PQTZ09]

The Zémor-Tillich hash function (ZT)

- Recall:
  - ZT is a Cayley hash with $G = SL(2, \mathbb{F}_{2^n})$, $v_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and

$$S = \left\{ s_0 = \begin{pmatrix} X & 1 \\ 1 & 0 \end{pmatrix}, s_1 = \begin{pmatrix} X & X+1 \\ 1 & 1 \end{pmatrix} \right\}$$

  - Generic collision and preimage subgroup attacks in time $2^{n/2}$ (instead of $2^{3n/2}$ and 2^{3n} for birthday and exhaustive) [PQTZ09]

- How to extract the secure bits ?

Vectorial and projective Zémor-Tillich

- Vectorial ZT: [PQTZ09]
  - For an initial vector $\begin{pmatrix} a_0 & b_0 \end{pmatrix}$ part of the key,

$$H_{ZT}^{vec}(m) = \begin{pmatrix} a_0 & b_0 \end{pmatrix} H_{ZT}(m)$$

  - Just as secure as the original ZT

Vectorial and projective Zémor-Tillich

- Vectorial ZT: [PQTZ09]
  - For an initial vector $\begin{pmatrix} a_0 & b_0 \end{pmatrix}$ part of the key,

$$H^{vec}_{ZT}(m) = \begin{pmatrix} a_0 & b_0 \end{pmatrix} H_{ZT}(m)$$

  - Just as secure as the original ZT

- Projective ZT: [PQTZ09]
  - For an initial vector $\begin{pmatrix} a_0 & b_0 \end{pmatrix}$ part of the key, returns the *projective point* $[a : b]$ if the vectorial ZT returns $\begin{pmatrix} a & b \end{pmatrix}$
  - "Nearly" as secure as the vectorial version

Vectorial and projective Zémor-Tillich

- (Nearly) as secure as the original version
- Reduced output sizes: $\approx 3n$ bits $\longrightarrow \approx 2n$ and $\approx n$ bits
- Keep hard components of the representation problem; remove easy components

Vectorial and projective Zémor-Tillich

- (Nearly) as secure as the original version
- Reduced output sizes: $\approx 3n$ bits $\longrightarrow \approx 2n$ and $\approx n$ bits
- Keep hard components of the representation problem; remove easy components
- Also more efficient:
  - Always for vectorial version
  - Always but on short messages for projective version

Vectorial and projective Zémor-Tillich

- (Nearly) as secure as the original version
- Reduced output sizes: $\approx 3n$ bits $\longrightarrow \approx 2n$ and $\approx n$ bits
- Keep hard components of the representation problem; remove easy components
- Also more efficient:
  - Always for vectorial version
  - Always but on short messages for projective version
- Used in our new function ZesT

Outline

- Introduction
- Generic construction and attacks
- Known instances
  - Overview
  - Focus 1 : Cryptanalysis of LPS and Morgenstern hash functions
  - Focus 2 : Vectorial and projective Zémor-Tillich
- Perspectives
- Conclusion

Malleability

- Inherent to expander hash design

Malleability

- Inherent to expander hash design
- Formalization:
  - Correlation intractability [CGH98]
  - Non-malleability [BCFW08]

Malleability

- Inherent to expander hash design
- Formalization:
  - Correlation intractability [CGH98]
  - Non-malleability [BCFW08]
- Applications:
  - Undesirable for auctions, ...
  - OK if collision resistance suffices
  - Useful for parallelism, [LM08], [QJ97]

Malleability

- Inherent to expander hash design
- Formalization:
  - Correlation intractability [CGH98]
  - Non-malleability [BCFW08]
- Applications:
  - Undesirable for auctions, ...
  - OK if collision resistance suffices
  - Useful for parallelism, [LM08], [QJ97]
- May be removed with additional design: ZesT

ZesT: an all-purpose hash function based on Zémor-Tillich [PVQ08,PdMQTVZ09]

- **ZT is appealing:** security proof for collision resistance, graph and group perspectives, parallelism, good efficiency...

ZesT: an all-purpose hash function based on Zémor-Tillich [PVQ08,PdMQTVZ09]

- **ZT is appealing:** security proof for collision resistance, graph and group perspectives, parallelism, good efficiency...
- **ZT has important issues:** malleability, invertibility on short messages, suboptimal collision and preimage resistances

ZesT: an all-purpose hash function based on Zémor-Tillich [PVQ08,PdMQTVZ09]

- **ZT is appealing:** security proof for collision resistance, graph and group perspectives, parallelism, good efficiency...
- **ZT has important issues:** malleability, invertibility on short messages, suboptimal collision and preimage resistances
- **ZesT** is Zémor-Tillich with Enhanced Security inside

ZesT: an all-purpose hash function based on Zémor-Tillich [PVQ08,PdMQTVZ09]

- Use vectorial and projective ZT as building blocks

ZesT: an all-purpose hash function based on Zémor-Tillich [PVQ08,PdMQTVZ09]

- Use vectorial and projective ZT as building blocks
- Collision resistance reduces to the representation problem of ZT
- Weaknesses of ZT are removed

ZesT: an all-purpose hash function based on Zémor-Tillich [PVQ08,PdMQTVZ09]

- Use vectorial and projective ZT as building blocks
- Collision resistance reduces to the representation problem of ZT
- Weaknesses of ZT are removed
- Ultra-lightweight ASIC implementations [dMPQ09]
- Throughput comparable to SHA on FPGA [dMPQ09]
- (Currently) 4 to 10 times as slow as SHA in software
- Parallelism still to be used

Outline

- Introduction
- Generic construction and attacks
- Known instances
  - Overview
  - Focus 1 : Cryptanalysis of LPS and Morgenstern hash functions
  - Focus 2 : Vectorial and projective Zémor-Tillich
- Perspectives
- Conclusion

Conclusion

- Today:
  - Zémor's first proposal broken
  - ZT unbroken since 1994
  - LPS, Morgenstern hashes broken (and repaired)
  - Pizer hash unbroken
  - Vectorial and projective ZT as secure as ZT

Conclusion

- Elegant, clear, simple design

Conclusion

- Elegant, clear, simple design
- Provable security, stated as graph and group properties

Conclusion

- Elegant, clear, simple design
- Provable security, stated as graph and group properties
- May be very efficient in software and hardware

Conclusion

- Elegant, clear, simple design
- Provable security, stated as graph and group properties
- May be very efficient in software and hardware
- Parallelism (Cayley hashes)

Conclusion

- Elegant, clear, simple design
- Provable security, stated as graph and group properties
- May be very efficient in software and hardware
- Parallelism (Cayley hashes)
- Main design issues (malleability,...) can be removed with additional design

Conclusion

- Ramanujan property not so benefic after all

Conclusion

- Ramanujan property not so benefic after all
- Underlying hard problems should be further studied

Conclusion

- Ramanujan property not so benefic after all
- Underlying hard problems should be further studied
- Malleability of hash functions should be further studied

Conclusion

- Ramanujan property not so benefic after all
- Underlying hard problems should be further studied
- Malleability of hash functions should be further studied

- Very interesting design !

Publications and preprints on expander hashes

- **ZesT : an all-purpose hash function based on Zémor-Tillich**
  Christophe Petit, Giacomo de Meulenaer, Jean-Jacques Quisquater, Jean-Pierre Tillich, Nicolas Veyrat-Charvillon and Gilles Zémor
  Preprint (2009)
- **Hardware Implementations of a Variant of the Zémor-Tillich Hash Function**
  Giacomo de Meulenaer, Christophe Petit and Jean-Jacques Quisquater
  Preprint (2009)

Publications and preprints on expander hashes

- **Hard and Easy Components of Collision Search in the Zémor-Tillich Hash Function : New Instances and Reduced Variants with Equivalent Security**
  Christophe Petit, Jean-Jacques Quisquater, Jean-Pierre Tillich and Gilles Zémor
  To appear in CT-RSA 2009
- **Full Cryptanalysis of LPS and Morgenstern Hash Functions**
  Christophe Petit, Kristin Lauter, and Jean-Jacques Quisquater
  SCN 2008 - Sixth Conference on Security and Cryptography for Networks

Publications and preprints on expander hashes

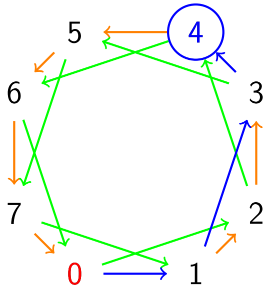
- **Efficiency and Pseudo-Randomness of a Variant of Zémor-Tillich Hash Function**
  Christophe Petit, Nicolas Veyrat-Charvillon, and Jean-Jacques Quisquater
  WIC'2008 - Symposium on Information Theory and Communication in the Bénélux
  ISECS'2008 - The 15th IEEE International Conference on Electronics, Circuits and Systems (invited paper)
- **Cayley Hashes: A Class of Efficient Graph-based Hash Functions**
  Christophe Petit, Kristin Lauter, and Jean-Jacques Quisquater
  Preprint (2007)

Other publications

- **Fault Attacks on Public Key Elements: Application to DLP based Schemes**
  Chong Hee Kim, Philippe Bulens, Christophe Petit, and Jean-Jacques Quisquater
  EUROPKI 2008
- **A Block Cipher based Pseudo Random Number Generator Secure Against Side-Channel Key Recovery**
  Christophe Petit, François-Xavier Standaert, Olivier Pereira, Tal G. Malkin, Moti Yung
  ASIACCS'08

Collisions and preimages

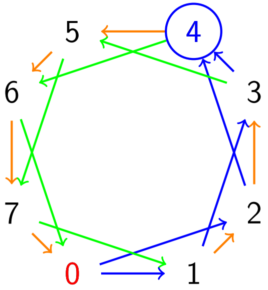
- Finding a *preimage* is finding *a path* starting at the origin and ending at some given vertex.



(For Crypto, $\geq 2^{160}$ vertices)

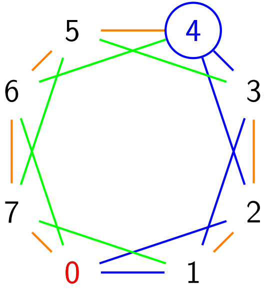
Collisions and preimages

- Finding a *collision* amounts to finding *two paths* starting at the origin and ending at the same vertex.



Collisions and preimages

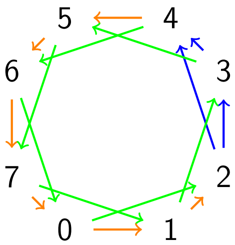
- If the graph is undirected, this amounts to finding a *cycle* through the origin.



Girth

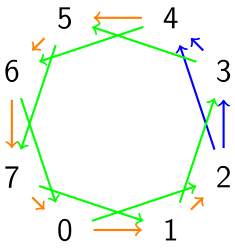
- For undirected graphs, the girth is the size of the smallest cycle

  For directed graphs



Girth

- For undirected graphs, the girth is the size of the smallest cycle
  For directed graphs



- The minimal “distance” between any collisions is given by the *girth* of the graph

Spectral expansion

- $\lambda := \max_{i \neq 0} |\lambda_i| \qquad (\lambda_i = \text{eigenvalues of the graph})$
  - k-regular graphs : $\lambda_0 = k \geq |\lambda_i|$
  - k-regular undirected graphs : $\lambda_0 = k \geq \lambda_1 \geq ... \geq \lambda_{n-1} \geq -k$

Spectral expansion

- $\lambda := \max_{i \neq 0} |\lambda_i|$ (λ_i = eigenvalues of the graph)
  - k-regular graphs : $\lambda_0 = k \geq |\lambda_i|$
  - k-regular undirected graphs : $\lambda_0 = k \geq \lambda_1 \geq \ldots \geq \lambda_{n-1} \geq -k$

- Uniform distribution of outputs iff convergence of random walks iff $\lambda < k$
- λ gives the rate of convergence
- **Expander graph** : λ small

Spectral expansion

- $\lambda := \max_{i \neq 0} |\lambda_i|$ $\quad$ (λ_i = eigenvalues of the graph)
  - k-regular graphs : $\lambda_0 = k \geq |\lambda_i|$
  - k-regular undirected graphs : $\lambda_0 = k \geq \lambda_1 \geq \ldots \geq \lambda_{n-1} \geq -k$

- Uniform distribution of outputs iff convergence of random walks iff $\lambda < k$
- λ gives the rate of convergence
- **Expander graph** : λ small

- Alon-Boppana (undirected graphs): $\liminf_{|V| \to +\infty} \lambda_1 \geq 2\sqrt{k-1}$
- **Ramanujan graph family** : $\liminf_{|V| \to +\infty} \lambda_1 = 2\sqrt{k-1}$

Expander hashes: security properties

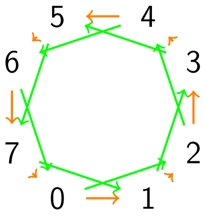
hash properties	**graph properties**
collision resistance	cycle / two-paths problem
preimage resistance	path-finding problem
output distribution	expanding properties
minimal collision "distance"	girth

Cayley hashes

- $\mathcal{C}_{G,S} = (V, E)$: for a *group* G and $S \subset G$, add
  - a vertex v_g for each $g \in G$
  - an edge (v_{g_1}, v_{g_2}) iff $\exists s \in S$ with $g_2 = g_1 s$

Cayley hashes

- $\mathcal{C}_{G,S} = (V, E)$: for a *group* G and $S \subset G$, add
  - a vertex v_g for each $g \in G$
  - an edge (v_{g_1}, v_{g_2}) iff $\exists s \in S$ with $g_2 = g_1 s$
- Example : $G = (\mathbb{Z}/8\mathbb{Z}, +)$, $S = \{1, 2\}$



Balance problem

- Find two products

$$\prod_{1 \le i \le N} s_{\theta(i)}^{e_i} = \prod_{1 \le i \le N'} s_{\theta'(i)}^{e'_i}$$

where...

Security of Cayley Hashes

- Similarly: balance, factorization problems
- For Cayley hashes,
  Solving the representation problem
  $\Rightarrow$ Finding collisions
  $\Rightarrow$ Solving the balance problem
- Equivalence if the graph is undirected

Security of Cayley Hashes

- Similarly: balance, factorization problems
- For Cayley hashes,
  Solving the representation problem
  $\Rightarrow$ Finding collisions
  $\Rightarrow$ Solving the balance problem
- Equivalence if the graph is undirected

- The hardness of these problems highly depends on G and S !
  Of course, G must be non-Abelian

Subgroup attacks on Cayley hashes

- Exploit group structure: $H(m||m') = H(m) \cdot H(m')$

Subgroup attacks on Cayley hashes

- Exploit group structure: $H(m||m') = H(m) \cdot H(m')$
- $s_i^{\mathrm{ord}(s_i)} = 1$ for any $s_i \in S$

Subgroup attacks on Cayley hashes

- Exploit group structure: $H(m||m') = H(m) \cdot H(m')$
- $s_i^{\mathrm{ord}(s_i)} = 1$ for any $s_i \in S$
- Choose graph parameters such that $\mathrm{ord}(s_i)$ is small (trapdoor attack)

Subgroup attacks on Cayley hashes

- Exploit group structure: $H(m||m') = H(m) \cdot H(m')$
- $s_i^{\mathrm{ord}(s_i)} = 1$ for any $s_i \in S$
- Choose graph parameters such that $\mathrm{ord}(s_i)$ is small (trapdoor attack)
- If there is a subgroup tower sequence $G = G_0 \supset G_1 \supset G_2 \supset \ldots \supset G_N = \{I\}$ such that $|G_{i-1}|/|G_i| \leq B$ for all i:
  use subgroup structure and birthday searches to get collisions in time $\sqrt{B}$

Zémor's first proposal

- Collision "lifting" attack [TZ93]
  - Find a lift of the identity:
    a matrix $M \in SL(2, \mathbb{Z})$ with $M = I \bmod p$
  - Solve the factorization problem in $SL(2, \mathbb{Z})$
    with a variant of the Euclidean algorithm
  - Very efficient algorithm

Zémor's first proposal

- Collision "lifting" attack [TZ93]
  - Find a lift of the identity:
    a matrix $M \in SL(2, \mathbb{Z})$ with $M = I \bmod p$
  - Solve the factorization problem in $SL(2, \mathbb{Z})$ with a variant of the Euclidean algorithm
  - Very efficient algorithm

- Trivially extends to a preimage attack
- This function is broken !

Vectorial Zémor-Tillich [PQTZ09]

- [PQTZ09]: the output of ZT is $3n$ bits while its security is n bits: how to extract the secure bits ?

Vectorial Zémor-Tillich [PQTZ09]

- [PQTZ09]: the output of ZT is $3n$ bits while its security is n bits: how to extract the secure bits ?
- Vectorial ZT:
  - Outputs $2n$ bits
  - For an initial vector $\begin{pmatrix} a_0 & b_0 \end{pmatrix}$ part of the key,

$$H_{ZT}^{vec}(m) = \begin{pmatrix} a_0 & b_0 \end{pmatrix} H_{ZT}(m)$$

Vectorial Zémor-Tillich [PQTZ09]

- [PQTZ09]: the output of ZT is $3n$ bits while its security is n bits: how to extract the secure bits ?
- Vectorial ZT:
  - Outputs $2n$ bits
  - For an initial vector $\begin{pmatrix} a_0 & b_0 \end{pmatrix}$ part of the key,

$$H_{ZT}^{vec}(m) = \begin{pmatrix} a_0 & b_0 \end{pmatrix} H_{ZT}(m)$$

- If the initial vector is chosen randomly,
  just as secure as the original matrix version

Equivalence between vectorial and matrix versions

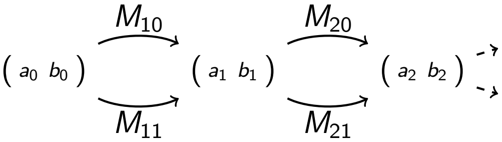
- Suppose $\exists$ algorithm finding collisions for the vectorial version...

$$\begin{pmatrix} a_0 & b_0 \end{pmatrix} \overset{M_{10}}{\underset{M_{11}}{\rightrightarrows}} \begin{pmatrix} a_1 & b_1 \end{pmatrix}$$

  - Run it on a random $\begin{pmatrix} a_0 & b_0 \end{pmatrix}$
    to get $\begin{pmatrix} a_1 & b_1 \end{pmatrix} := \begin{pmatrix} a_0 & b_0 \end{pmatrix} M_{10} = \begin{pmatrix} a_0 & b_0 \end{pmatrix} M_{11}$
    where M_{10} and M_{11} are the ZT hash values of the colliding messages

Equivalence between vectorial and matrix versions

- Suppose $\exists$ algorithm finding collisions for the vectorial version...



  - Run it on a random $(\, a_0 \;\; b_0 \,)$
    to get $(\, a_1 \;\; b_1 \,) := (\, a_0 \;\; b_0 \,)\, M_{10} = (\, a_0 \;\; b_0 \,)\, M_{11}$
    where M_{10} and M_{11} are the ZT hash values of the colliding messages
  - Run it on $(\, a_1 \;\; b_1 \,)$ to get $(\, a_2 \;\; b_2 \,) := (\, a_1 \;\; b_1 \,)\, M_{20} = (\, a_1 \;\; b_1 \,)\, M_{21}$
  - Repeat $n+1$ times

Equivalence between vectorial and matrix versions

- Key observations
  - $M_{1j} = \begin{pmatrix} a_0^{-1} & b_0 \\ 0 & a_0 \end{pmatrix} \begin{pmatrix} a_1 & b_1 \\ 0 & a_1^{-1} \end{pmatrix} + \epsilon_{1j} \begin{pmatrix} b_0 \\ a_0 \end{pmatrix} \begin{pmatrix} a_1 & b_1 \end{pmatrix}$

Equivalence between vectorial and matrix versions

- Key observations
  - $M_{1j} = \begin{pmatrix} a_0^{-1} & b_0 \\ 0 & a_0 \end{pmatrix} \begin{pmatrix} a_1 & b_1 \\ 0 & a_1^{-1} \end{pmatrix} + \epsilon_{1j} \begin{pmatrix} b_0 \\ a_0 \end{pmatrix} \begin{pmatrix} a_1 & b_1 \end{pmatrix}$
  - $M_{1j_1} M_{2j_2} = \begin{pmatrix} a_0^{-1} & b_0 \\ 0 & a_0 \end{pmatrix} \begin{pmatrix} a_2 & b_2 \\ 0 & a_2^{-1} \end{pmatrix} + (\epsilon_{1j_1} + \epsilon_{2j_2}) \begin{pmatrix} b_0 \\ a_0 \end{pmatrix} \begin{pmatrix} a_2 & b_2 \end{pmatrix}$

Equivalence between vectorial and matrix versions

- Key observations
  - $M_{1j} = \begin{pmatrix} a_0^{-1} & b_0 \\ 0 & a_0 \end{pmatrix} \begin{pmatrix} a_1 & b_1 \\ 0 & a_1^{-1} \end{pmatrix} + \epsilon_{1j} \begin{pmatrix} b_0 \\ a_0 \end{pmatrix} \begin{pmatrix} a_1 & b_1 \end{pmatrix}$
  - $M_{1j_1} M_{2j_2} = \begin{pmatrix} a_0^{-1} & b_0 \\ 0 & a_0 \end{pmatrix} \begin{pmatrix} a_2 & b_2 \\ 0 & a_2^{-1} \end{pmatrix} + (\epsilon_{1j_1} + \epsilon_{2j_2}) \begin{pmatrix} b_0 \\ a_0 \end{pmatrix} \begin{pmatrix} a_2 & b_2 \end{pmatrix}$
  - "Homomorphism"
    $\prod_{i=1}^{k} M_{ij_i} = \begin{pmatrix} a_0^{-1} & b_0 \\ 0 & a_0 \end{pmatrix} \begin{pmatrix} a_k & b_k \\ 0 & a_k^{-1} \end{pmatrix} + \left(\sum_{i=1}^{k} \epsilon_{ij_i}\right) \begin{pmatrix} b_0 \\ a_0 \end{pmatrix} \begin{pmatrix} a_k & b_k \end{pmatrix}$

Equivalence between vectorial and matrix versions

- Key observations
  - $M_{1j} = \begin{pmatrix} a_0^{-1} & b_0 \\ 0 & a_0 \end{pmatrix} \begin{pmatrix} a_1 & b_1 \\ 0 & a_1^{-1} \end{pmatrix} + \epsilon_{1j} \begin{pmatrix} b_0 \\ a_0 \end{pmatrix} \begin{pmatrix} a_1 & b_1 \end{pmatrix}$
  - $M_{1j_1} M_{2j_2} = \begin{pmatrix} a_0^{-1} & b_0 \\ 0 & a_0 \end{pmatrix} \begin{pmatrix} a_2 & b_2 \\ 0 & a_2^{-1} \end{pmatrix} + (\epsilon_{1j_1} + \epsilon_{2j_2}) \begin{pmatrix} b_0 \\ a_0 \end{pmatrix} \begin{pmatrix} a_2 & b_2 \end{pmatrix}$
  - "Homomorphism"
    $\prod_{i=1}^{k} M_{ij_i} = \begin{pmatrix} a_0^{-1} & b_0 \\ 0 & a_0 \end{pmatrix} \begin{pmatrix} a_k & b_k \\ 0 & a_k^{-1} \end{pmatrix} + \left(\sum_{i=1}^{k} \epsilon_{ij_i} \right) \begin{pmatrix} b_0 \\ a_0 \end{pmatrix} \begin{pmatrix} a_k & b_k \end{pmatrix}$

- To find a collision
  - Let $\epsilon_i := \epsilon_{i0} + \epsilon_{i1}$
  - Find $I \subset \{1, 2, ..., n+1\}$ such that $\sum_{i \in I} \epsilon_i = 0$

Equivalence between vectorial and matrix versions

- Colliding messages:
  - $m = m_{10}||m_{20}||...||m_{n+1,0}$
  - $m' = m_{1e_1}||m_{2e_2}||...||m_{n+1,e_{n+1}}$ where $e_i = 1$ if $i \in I$

- The two messages collide to the value

$$\begin{aligned} H_{ZT}(m) &= \begin{pmatrix} a_0^{-1} & b_0 \\ 0 & a_0 \end{pmatrix} \begin{pmatrix} a_{n+1} & b_{n+1} \\ 0 & a_{n+1}^{-1} \end{pmatrix} + \left(\sum_{i=1}^{n+1} \epsilon_{i0} \right) \begin{pmatrix} b_0 \\ a_0 \end{pmatrix} \begin{pmatrix} a_{n+1} & b_{n+1} \end{pmatrix} \\ &= H_{ZT}(m') \end{aligned}$$

Projective Zémor-Tillich [PQTZ09]

- [PQTZ09]: the output of ZT is $3n$ bits while its security is n bits: how to extract the secure bits ?

Projective Zémor-Tillich [PQTZ09]

- [PQTZ09]: the output of ZT is $3n$ bits while its security is n bits: how to extract the secure bits ?
- Projective ZT:
  - Outputs n bits
  - Returns $[a : b] \in \mathbb{P}^1(\mathbb{F}_{2^n})$ if the vectorial version returns $\begin{pmatrix} a & b \end{pmatrix}$

Projective Zémor-Tillich [PQTZ09]

- [PQTZ09]: the output of ZT is $3n$ bits while its security is n bits: how to extract the secure bits ?
- Projective ZT:
  - Outputs n bits
  - Returns $[a : b] \in \mathbb{P}^1(\mathbb{F}_{2^n})$ if the vectorial version returns $\begin{pmatrix} a & b \end{pmatrix}$
- If the initial vector is chosen randomly,
  "nearly" as secure as the original matrix version

"Quasi" equivalence between projective and vectorial versions

- Suppose $\exists$ algorithm finding collision for the projective version...
  - Run it on $\begin{pmatrix} a_0 & b_0 \end{pmatrix}$ to get $\begin{pmatrix} a_{10} & b_{10} \end{pmatrix}$ and $\begin{pmatrix} a_{11} & b_{11} \end{pmatrix} = \lambda_1 \begin{pmatrix} a_{10} & b_{10} \end{pmatrix}$
  - Run it on $\begin{pmatrix} a_{10} & b_{10} \end{pmatrix}$ to get $\begin{pmatrix} a_{20} & b_{20} \end{pmatrix}$ and $\begin{pmatrix} a_{21} & b_{21} \end{pmatrix} = \lambda_2 \begin{pmatrix} a_{20} & b_{20} \end{pmatrix}$
  - After n' steps, find $I \subset \{1, 2, ..., n'\}$ such that $\prod_{i \in I} \lambda_i = 1$

"Quasi" equivalence between projective and vectorial versions

- Suppose $\exists$ algorithm finding collision for the projective version...
  - Run it on $\begin{pmatrix} a_0 & b_0 \end{pmatrix}$ to get $\begin{pmatrix} a_{10} & b_{10} \end{pmatrix}$ and $\begin{pmatrix} a_{11} & b_{11} \end{pmatrix} = \lambda_1 \begin{pmatrix} a_{10} & b_{10} \end{pmatrix}$
  - Run it on $\begin{pmatrix} a_{10} & b_{10} \end{pmatrix}$ to get $\begin{pmatrix} a_{20} & b_{20} \end{pmatrix}$ and $\begin{pmatrix} a_{21} & b_{21} \end{pmatrix} = \lambda_2 \begin{pmatrix} a_{20} & b_{20} \end{pmatrix}$
  - After n' steps, find $I \subset \{1, 2, ..., n'\}$ such that $\prod_{i \in I} \lambda_i = 1$

- Complexity of last step
  - Hard asymptotically
    n' discrete logarithms problems + one subset sum problem
  - Feasible for $n \leq 170$