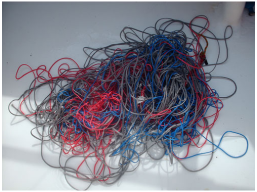
Full Cryptanalysis of LPS and Morgenstern Hash Functions

Christophe Petit, Kristin Lauter, Jean-Jacques Quisquater

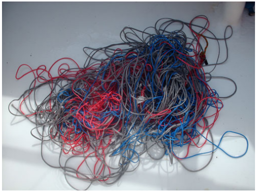
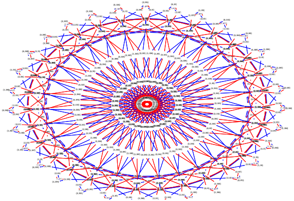
Cryptographic Hash Functions from Expander Graphs

- Idea of "Expander Hashes" [ZT, CGL]

replace 

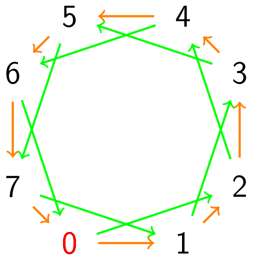
Cryptographic Hash Functions from Expander Graphs

- Idea of "Expander Hashes" [ZT, CGL]

replace  by 

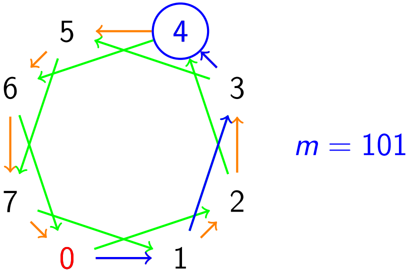
Hash Function from a Regular Graph

- Take a k-regular (directed) graph
- Decompose the message in k-digits $m = m_1 m_2 \ldots m_N$



Hash Function from a Regular Graph

- Take a k-regular (directed) graph
- Decompose the message in k-digits $m = m_1 m_2 ... m_N$



- The message determines a walk in the graph

Expander Hashes

- $+$ Clear, simple design
- $+$ Graph properties $\Rightarrow$ hash properties
  girth, expanding constant, cycles
- $+$ Cayley Hashes: parallel computation
- $\pm$ Security reduction to (not so well-known) mathematical problems

Expander Hashes

- $+$ Clear, simple design
- $+$ Graph properties $\Rightarrow$ hash properties
  girth, expanding constant, cycles
- $+$ Cayley Hashes: parallel computation
- $\pm$ Security reduction to (not so well-known) mathematical problems
  - Zémor-Tillich Hash [ZT1994]: unbroken
  - LPS Hash [CGL2007]: collisions [TZ2008], now preimages
  - Morgenstern Hash: now collisions and preimages
  - Pizer Hashes [CGL2007]: unbroken

Outline

- **Introduction**
- LPS Hash Function
- Tillich-Zémor Collisions Attack
- Extension to a Preimage Attack
- Extension to Morgenstern Hashes
- Conclusion

Outline

- Introduction
- **LPS Hash Function**
- Tillich-Zémor Collisions Attack
- Extension to a Preimage Attack
- Extension to Morgenstern Hashes
- Conclusion

Cayley Hashes

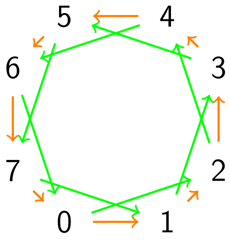
- LPS hash is a *Cayley hash*

Cayley Hashes

- LPS hash is a *Cayley hash*
- Cayley graph $\mathcal{C}_{G,S} = (V, E)$: 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

- LPS hash is a *Cayley hash*
- Cayley graph $\mathcal{C}_{G,S} = (V, E)$: 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 of Cayley graph : $G = (\mathbb{Z}/8\mathbb{Z}, +)$, $S = \{1, 2\}$



LPS Hash Function

- Construction : use LPS hashes [LPS1988]

LPS Hash Function

- Construction : use LPS hashes [LPS1988]
  - 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 $\mathbf{i}^2 = -1 \bmod p$

LPS Hash Function

- Construction : use LPS hashes [LPS1988]
  - 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 $\mathbf{i}^2 = -1 \bmod p$
  - G is $PSL(2, \mathbb{F}_p)$
  - S is $\{G_j, j = 1...l+1\}$, where

$$G_j = \begin{pmatrix} g_{0,j} + \mathbf{i}g_{1,j} & g_{2,j} + \mathbf{i}g_{3,j} \\ -g_{2,j} + \mathbf{i}g_{3,j} & g_{0,j} - \mathbf{i}g_{1,j} \end{pmatrix}, \qquad j = 1, ...l+1;$$

$(g_{0,j}, g_{1,j}, g_{2,j}, g_{3,j})$ are all the solutions of
$g_0^2 + g_1^2 + g_2^2 + g_3^2 = l$, with $g_0 > 0$ and g_1, g_2, g_3 even

LPS Hash Function

- Construction : use LPS hashes [LPS1988]
  - 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 $\mathbf{i}^2 = -1 \bmod p$
  - G is $PSL(2, \mathbb{F}_p)$
  - S is $\{G_j, j = 1...l+1\}$, where

$$G_j = \begin{pmatrix} g_{0,j} + \mathbf{i}g_{1,j} & g_{2,j} + \mathbf{i}g_{3,j} \\ -g_{2,j} + \mathbf{i}g_{3,j} & g_{0,j} - \mathbf{i}g_{1,j} \end{pmatrix}, \qquad j = 1, ...l+1;$$

  $(g_{0,j}, g_{1,j}, g_{2,j}, g_{3,j})$ are all the solutions of
  $g_0^2 + g_1^2 + g_2^2 + g_3^2 = l$, with $g_0 > 0$ and g_1, g_2, g_3 even
- Undirected Cayley hash, but backtracking is not allowed

The Representation Problem

- Finding collisions for LPS hash is as hard as solving the corresponding **Representation Problem** [CGL2007]

Find a product (in reduced form)

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

The Representation Problem

- Finding collisions for LPS hash is as hard as solving the corresponding **Representation Problem** [CGL2007]

Find a product (in reduced form)

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

where e_i are integers, $\theta : \{1, ...N\} \to \{1...k\}$ and $\sum e_i$ is "small" in the size of G.
Reduced form: for each i, $G_{\theta(i+1)} \neq G_{\theta(i)}, G_{\theta(i)}^{-1}$.

Outline

- Introduction
- LPS Hash Function
- **Tillich-Zémor Collisions Attack**
- Extension to a Preimage Attack
- Extension to Morgenstern Hashes
- Conclusion

Collisions for LPS Hash [ZT2008]

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

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

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

Collisions for LPS Hash [ZT2008]

- 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 of (actually all) $m \in \Omega$ 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

- Choose

$$\Omega = \left\{ \begin{pmatrix} a+bi & c+di \\ -c+di & a-bi \end{pmatrix} \middle| (a,b,c,d) \in E_e \text{ for some } e > 0 \right\}$$

where E_e is the set of 4-tuples $(a,b,c,d) \in \mathbb{Z}^4$ such that

$$\begin{cases} a^2+b^2+c^2+d^2 = l^e \\ a > 0, a \equiv 1 \bmod 2 \\ b \equiv c \equiv d \equiv 0 \bmod 2. \end{cases}$$

The lifted set Ω

- Choose

$$\Omega = \left\{ \begin{pmatrix} a+bi & c+di \\ -c+di & a-bi \end{pmatrix} | (a,b,c,d) \in E_e \text{ for some } e>0 \right\}$$

where E_e is the set of 4-tuples $(a,b,c,d) \in \mathbb{Z}^4$ such that

$$\begin{cases} a^2+b^2+c^2+d^2 = l^e \\ a>0, a \equiv 1 \mod 2 \\ b \equiv c \equiv d \equiv 0 \mod 2. \end{cases}$$

- Up to a unit, $m \in \Omega$ has unique factorization [LPS1988]
Here, we may forget the unit [TZ2008]

Lifting to Ω

- Lifting the identity to Ω amounts to solve

$$\begin{cases} a^2 + b^2 + c^2 + d^2 = l^e \\ a > 0, a \equiv 1 \bmod 2 \\ b \equiv c \equiv d \equiv 0 \bmod 2 \\ a - \lambda \equiv b \equiv c \equiv d \equiv 0 \bmod p \end{cases}$$

Lifting to Ω

- Lifting the identity to Ω amounts to solve

$$\begin{cases} a^2+b^2+c^2+d^2 = l^e \\ a>0, a\equiv 1 \bmod 2 \\ b\equiv c\equiv d\equiv 0 \bmod 2 \\ a-\lambda\equiv b\equiv c\equiv d\equiv 0 \bmod p \end{cases}$$

or

$$\begin{cases} (\lambda+wp)^2+4(xp)^2+4(yp)^2+4(zp)^2 = l^e \\ \lambda+wp>0 \\ \lambda+wp\equiv 1 \bmod 2 \end{cases}$$

Tillich-Zémor collision attack

- To find (w, x, y, z) and λ such that

$$\begin{cases} (\lambda + wp)^2 + 4(xp)^2 + 4(yp)^2 + 4(zp)^2 = l^e \\ \lambda + wp > 0 \\ \lambda + wp \equiv 1 \bmod 2 \end{cases}$$

  - take e even: $e = 2k$
  - choose $\lambda + wp = l^k - 2mp^2$ for $m = 1$ or 2
    (so the equation is **satisfied modulo 4mp²**)
  - "simplify" by $4mp^2$: we get $x^2 + y^2 + z^2 = n := m(l^k - mp^2)$

Tillich-Zémor collision attack

- To find (w, x, y, z) and λ such that

$$\begin{cases} (\lambda + wp)^2 + 4(xp)^2 + 4(yp)^2 + 4(zp)^2 = l^e \\ \lambda + wp > 0 \\ \lambda + wp \equiv 1 \bmod 2 \end{cases}$$

  - take e even: $e = 2k$
  - choose $\lambda + wp = l^k - 2mp^2$ for $m = 1$ or 2
    (so the equation is **satisfied modulo 4mp²**)
  - "simplify" by $4mp^2$: we get $x^2 + y^2 + z^2 = n := m(l^k - mp^2)$
  - pick random x until $\mathbf{n - x^2}$ is a **sum of two squares**
  - find y and z with **Euclidean algorithm**: we are done !

Outline

- Introduction
- LPS Hash Function
- Tillich-Zémor Collisions Attack
- **Extension to a Preimage Attack**
- Extension to Morgenstern Hashes
- Conclusion

Preimages for LPS Hash

- Lift again to Ω: given $M = \begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} \in PSL(2, \mathbb{F}_p)$...

Preimages for LPS Hash

- Lift again to Ω: given $M = \begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} \in PSL(2, \mathbb{F}_p)$...
- ... first write $M = \begin{pmatrix} A+B\mathbf{i} & C+D\mathbf{i} \\ -C+D\mathbf{i} & A-B\mathbf{i} \end{pmatrix}$.

Preimages for LPS Hash

- Lift again to Ω: given $M = \begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} \in PSL(2, \mathbb{F}_p)$...
- ... first write $M = \begin{pmatrix} A+B\mathbf{i} & C+D\mathbf{i} \\ -C+D\mathbf{i} & A-B\mathbf{i} \end{pmatrix}$.
- We look for (a, b, c, d) such that

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

(plus some congruence conditions)

Preimages for LPS Hash

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

Preimages for LPS Hash

- $(A\lambda + wp)^2 + (B\lambda + xp)^2 + (C\lambda + yp)^2 + (D\lambda + zp)^2 = l^{2k}$
- Trivial extension does not work:
  - Fixing $A\lambda + wp$ to satisfy the equation modulo p...

Preimages for LPS Hash

- $(A\lambda + wp)^2 + (B\lambda + xp)^2 + (C\lambda + yp)^2 + (D\lambda + zp)^2 = l^{2k}$
- 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 $2p(wA + xB + yC + zD)\lambda$.

Preimages for LPS Hash

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

Preimages for LPS Hash

- $(A\lambda + wp)^2 + (B\lambda + xp)^2 + (C\lambda + yp)^2 + (D\lambda + zp)^2 = l^{2k}$
- 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 would most likely have no solution.

Preimages for LPS Hash

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

Preimages for LPS Hash

- Solution for diagonal matrices :

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

Preimages for LPS Hash

- Solution for diagonal matrices :

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

  - Fix λ to satisfy the equation modulo p

Preimages for LPS Hash

- Solution for diagonal matrices :

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

  - Fix λ to satisfy the equation modulo p
  - Pick random w, x that satisfy the equation modulo p^2 and "simplify by p^2" ...

  ... until the resulting equation $\mathbf{y^2} + \mathbf{z^2} = \mathbf{n}$ has solution

Preimages for LPS Hash

- Solution for diagonal matrices :

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

  - Fix λ to satisfy the equation modulo p
  - Pick random w, x that satisfy the equation modulo p^2 and "simplify by p^2"...

    ... until the resulting equation $\mathbf{y^2} + \mathbf{z^2} = \mathbf{n}$ has solution
  - Use Euclidean algorithm: we are done with diagonal case!

Preimages for LPS Hash

- Decompose any matrix as a product of *diagonal matrices* and *graphs generators*:
$$\begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \lambda \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix}$$
    - Find λ and squares α, ω, β_1, β_2 such that

$$\begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \lambda \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} f_1 & f_2 \\ f_3 & f_4 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix} = \lambda \begin{pmatrix} f_1 & \omega f_2 \\ \alpha f_3 & \alpha\omega f_4 \end{pmatrix}$$

and

$$\begin{pmatrix} f_1 & f_2 \\ f_3 & f_4 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}$$

Preimages for LPS Hash

- Decompose any matrix as a product of *diagonal matrices* and *graphs generators*:
$\begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \lambda \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix}$
  - Find λ and squares α, ω, β_1, β_2 such that

$$\begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \lambda \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} f_1 & f_2 \\ f_3 & f_4 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix} = \lambda \begin{pmatrix} f_1 & \omega f_2 \\ \alpha f_3 & \alpha\omega f_4 \end{pmatrix}$$

    and

$$\begin{pmatrix} f_1 & f_2 \\ f_3 & f_4 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}$$

  - Pick random square β_1, solve for β_2 then α and ω until α, ω, β_2 are squares

Outline

Cryptanalysis of Morgenstern Hash

- LPS graphs for odd primes $l \longrightarrow$ Morgenstern graphs for l^k, including $l = 2$ [M1994]
  Morgenstern hashes use $l = 2$ [PLQ2007]

Cryptanalysis of Morgenstern Hash

- LPS graphs for odd primes $l \rightarrow$ Morgenstern graphs for l^k, including $l = 2$ [M1994]
  Morgenstern hashes use $l = 2$ [PLQ2007]
- 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 equation differs, but can be solved with the same techniques extended to polynomials
- See the paper for details

Outline

- Introduction
- LPS Hash Function
- Tillich-Zémor Collisions Attack
- Extension to a Preimage Attack
- Extension to Morgenstern Hashes
- **Conclusion**

Conclusion

- Collisions and Preimages for LPS and Morgenstern hashes
  - Rough runtime analysis: probabilistic polynomial time
  - 1024-bit parameters in less than 2min

Conclusion

- Collisions and Preimages for LPS and Morgenstern hashes
  - Rough runtime analysis: probabilistic polynomial time
  - 1024-bit parameters in less than 2min
- Our algorithms may be useful elsewhere:
  Graph Theory, Computer Science, attacking ZT hash (?), ...

Conclusion

- Collisions and Preimages for LPS and Morgenstern hashes
  - Rough runtime analysis: probabilistic polynomial time
  - 1024-bit parameters in less than 2min
- Our algorithms may be useful elsewhere:
  Graph Theory, Computer Science, attacking ZT hash (?), ...

- The attacks use extra structure given by those graphs
  - Both hash functions can be modified in a safe way
- We do **not** recommend to give up Expander Hashes
  - Other instances like ZT and Pizer are still safe