Symmetric States and their Structure

Improved Analysis Of CubeHash

Niels Ferguson, Stefan Lucks, and Kerry A. McKay

Outline

- Two main parts to this talk
- Symmetry structure
  - Builds on structure presented in [Aum09]
- Attacks
  - Multicollision
  - Preimage
    - Small improvements over preimage attacks in [Aum09]

Symmetry in CubeHash Round

- Symmetric states described in [Aum09]
  - 32-bit words equal
  - Defined 15 distinct symmetry classes
  - Any symmetry class in CubeHash can be described by these 15, or intersections of these 15

C_1	AABBCCDD	EEFFGGHH	IIJJKKLL	MMNNOOPP
C_2	ABABCDCD	EFEFGHGH	IJIJKLKL	MNMNOPOP
C_3	ABBACDDC	EFFEGHHG	IJJIKLLK	MNNMOPPO
C_4	ABCDABCD	EFGHEFGH	IJKLIJKL	MNOPMNOP
C_5	ABCDBADC	EFGHFEHG	IJKLJILK	MNOPNMPO
C_6	ABCDCDAB	EFGHGHEF	IJKLKLIJ	MNOPOPMN
C_7	ABCDDCBA	EFGHHGFE	IJKLLKJI	MNOPPONM
C_8	ABCDEFGH	ABCDEFGH	IJKLMNOP	IJKLMNOP
C_9	ABCDEFGH	BADCFEHG	IJKLMNOP	JILKNMPO
C_{10}	ABCDEFGH	CDABGHEF	IJKLMNOP	KLIJOPMN
C_{11}	ABCDEFGH	DCBAHGFE	IJKLMNOP	LKJIPONM
C_{12}	ABCDEFGH	EFGHABCD	IJKLMNOP	MNOPIJKL
C_{13}	ABCDEFGH	FEHGBADC	IJKLMNOP	NMPOJILK
C_{14}	ABCDEFGH	GHEFCDAB	IJKLMNOP	OPMNKLIJ
C_{15}	ABCDEFGH	HGFEDCBA	IJKLMNOP	PONMLKJI

Table 1. Symmetry Classes [2]

- We show additional structure to these symmetry classes

Hierarchy

- Let S be the state in 32-bit words
- Let V be the set of all 4-bit vectors, D a linear subspace of V, d in D
  - Symmetry when S[i] = S[i⊕d], S[16+i] = S[16+(i⊕d)]

D	Free words	Subspaces	Values
D=V	2	1	2^{64}
D is 3-d linear subspace	4	15	2^{128}
D is 2-d linear subspace	8	35	2^{256}
D is 1-d linear subspace	16	15	2^{512}

15 classes in [Aum09]

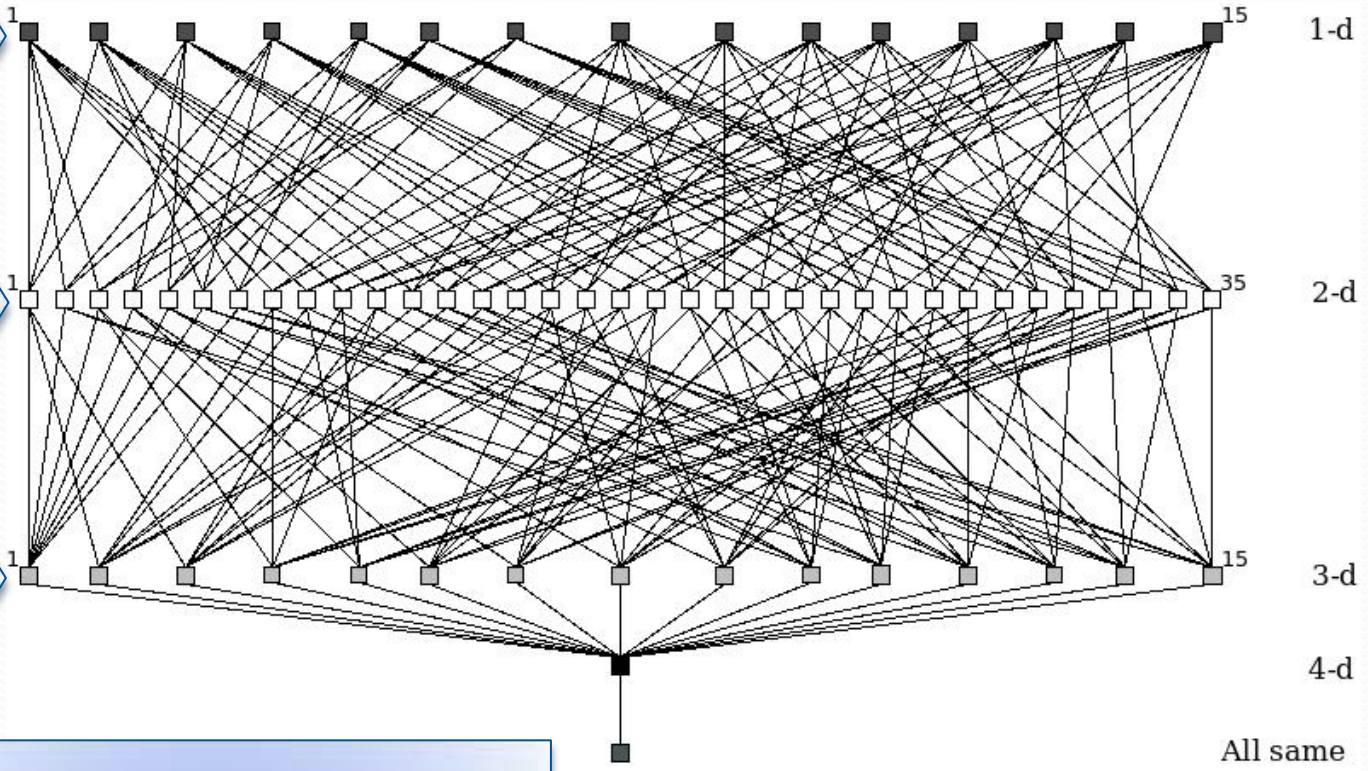
- One additional state with all words equal

Hierarchy (continued)

Each 1-d subspace is part of 7 2-d subspaces

Each 2-d subspace is part of 3 3-d subspaces, contains 3 1-d subspaces

Each 3-d subspace contains 7 2-d subspaces



From 3-d:
Have 15*7*3 = 315 hierarchies

Previous Attacks

- We revisit attacks from [Aum09]
  - Preimage and collision attacks using symmetric states
  - At the time, b=1
    - b is size of a message block, in bytes
- Noted issues
  - Attack that uses null messages to bridge two *different* symmetry classes does not work
    - Once in a symmetry class, one cannot leave it without injecting a nonzero message
  - Attacker is restricted to a fixed permutation via null messages
    - Even if two states, S and T, are in the same symmetry class, one cannot always get from S to T via null messages
      - Disjoint cycles

Our Attacks

- We look at b=32 and b=33
  - Current recommendation is b=32
- Use nonzero messages and stay in symmetry class
  - b=32: symmetry classes 1-7
    - 2^{128}-1 nonzero messages
  - b=33: any symmetry class
    - 2^{8}-1 nonzero messages
- As in [Aum09], our attacks rely on reaching a symmetric state

Multicollision

- Can find multicollisions [Joux04] once in a symmetric state
  - ceiling(lg k) × 2^{256} to find a k-collision
- b=32
  - $2^{381.2}$ to reach a symmetric state, C_1 to C_7
  - Total: $2^{381.2}$ + ceiling(lg k) × 2^{256}
  - Reaching symmetric state dominates
- b=33
  - 2^{253} to reach any symmetric state
  - Total: 2^{253} + ceiling(lg k) × 2^{256} ≈ ceiling(lg k) × 2^{256}
  - Multicollision dominates

Preimage

- Similar to preimage attack in [Aum09]
  - We show small improvements

- Attack structure
  1. Extend state by 1024-h bits and run finalization backwards to obtain state H_4
  2. Search for message prefix M_1 by injecting random blocks until any suitable symmetric class C_i is reached
     - $H_1 = H(H_0, M_1)$
  3. Search for postfix M_4 and state H_3 such that H_3 is in the same symmetry class as H_1, $H_4 = H(H_3, M_4)$
  4. Apply meet-in-the-middle to obtain a message $M = (M_1 \| M_2 \| M_3 \| M_4)$

Preimage (cont)

- Instead of targeting a particular class, target any reasonable class
  - Provides a little more flexibility and slightly reduces complexity
  - 2^{253} to reach a symmetric state, b=33
  - $2^{381.2}$ for b=32
- Once a symmetry state is reached in step 2, it fixes the class needed for step 3
  - Improvement: find two classes in step 2
    - Increases step 2 work to $2^{382.4}$
    - Reduces the work in step 3 to reach a compatible state from 2^{384} to 2^{383}
- For b=32
  - Dominated by 2nd and 3rd steps, with complexity about $2^{383.7}$
- For b=33
  - Can't get much better than 3×2^{256}

Conclusion

- For b=32, presented multicollision and preimage with complexity slightly less than 2^{384} hash operations
- For b=33, presented multicollision and preimage with complexity slightly more than 2^{256} hash operations
- Can work when b≥5
  - For smaller b, complexity increases
  - If b≤4, it seems impossible
- Provided hierarchical structure for symmetry classes

Thank you!

References

[Aum09] Aumasson, J.P., Brier, E., Meier, W., Naya-Plasencia, M., Peyrin, T.: Inside the hypercube. In Boyd, C., Nieto, J.M.G., eds.: ACISP. Volume 5594 of LNCS., Springer (2009) 202–213

[Bern09] Bernstein, D.J.: Cubehash specification (2.b.1). Submission to NIST (Round 2) (2009)

[Joux04] Joux, A.: Multicollisions in iterated hash functions. application to cascaded constructions. In Franklin, M.K., ed.: CRYPTO. Volume 3152 of LNCS., Springer (2004) 306–316