Breaking a New Hash Function Design Strategy Called SMASH

Norbert Pramstaller

Conference Hash Functions - Krakow

2005/06/24

Institute for Applied Information Processing and Communications (IAIK) - Krypto Group

Faculty of Computer Science
Graz University of Technology

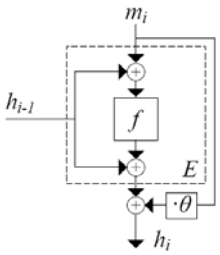
The work described in this paper has been supported by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT.

Outline

- The SMASH design strategy
- Observation on the design method
  - Forward prediction property
  - Pattern construction property
- Breaking simple instances of SMASH design strategy
  - *SMASH-ORD3* and *SMASH-ORDy*
- The proposed hash functions *SMASH-256 and SMASH-512*
- Breaking *SMASH-256*
- Conclusion

The SMASH design strategy

- Hash value computation



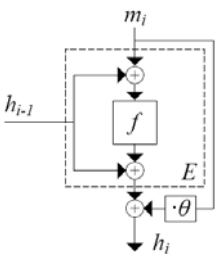
$$h_0 = f(\mathrm{iv}) + \mathrm{iv}$$
$$h_i = f(h_{i-1} + m_i) + h_{i-1} + \theta m_i \quad \text{for } i = 1, \ldots, t$$
$$h_{t+1} = f(h_t) + h_t \ .$$

  - Nonlinear compression function f based on bijective n-bit mapping
  - θ is an arbitrary field element in $GF(2^n)$ with $\theta \neq \{0, 1\}$

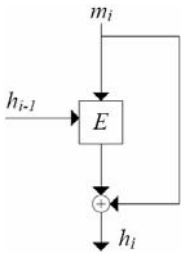
$+$ … addition in $GF(2^n)$

The SMASH design strategy

- Comparing SMASH with block cipher based hash function
  - Block cipher E following the Even-Mansour construction
  - Matyas-Meyer-Oseas operation mode



SMASH scheme



Matyas-Meyer-Oseas scheme

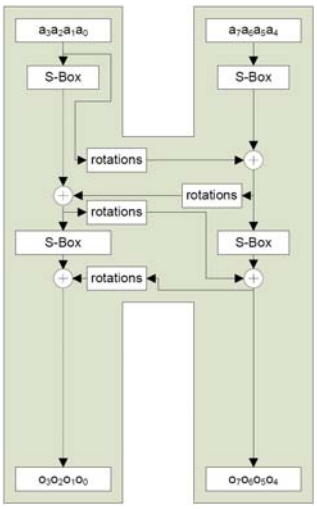
  - Only difference is multiplication with θ

The core function *f*

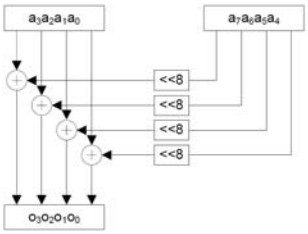
- Consists of
  - H-rounds: S-Boxes and diffusion layers
  - L-rounds: shift operations

$$f = \ldots H_3 \circ L \circ H_3 \circ H_2 \circ H_1$$

H-round

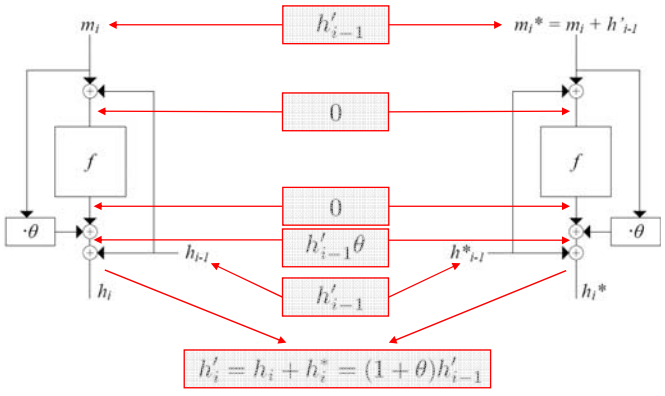


L-round



The forward prediction property

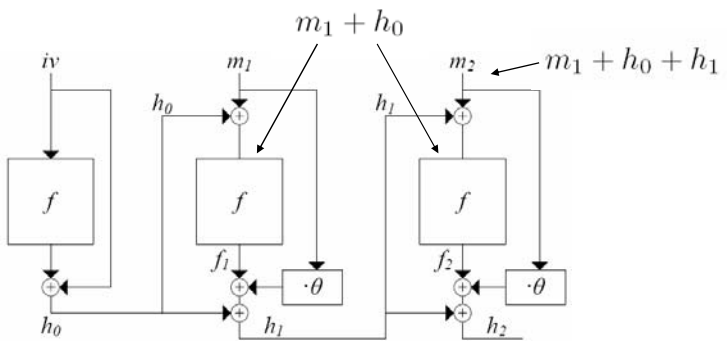
- Given intermediate hash values h_{i-1}, h^*_{i-1} with difference $h'_{i-1} = h_{i-1} + h^*_{i-1}$
- Choose m_i and compute $m^*_i = m_i + h'_{i-1}$



The attack strategy to break SMASH-256 and SMASH-512

The pattern construction property

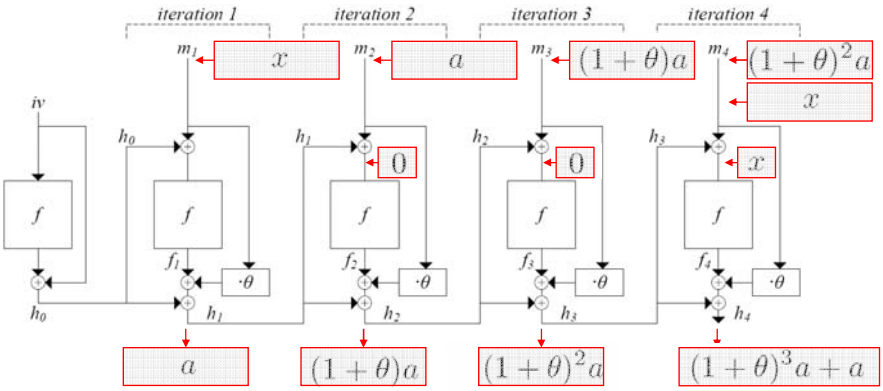
- Input of f must be the same for both iterations



$$m_2 = m_1 + f_1 + \theta m_1 \Rightarrow f_2 = f_1$$

The simple instance SMASH-ORD3

- Assume we can choose a θ such that $(1+\theta)^3 = 1$



The simple instance SMASH-ORDy

- Generalization of the attack on SMASH-ORD3
- Assume we can choose a θ such that $ord(1+\theta) = y$
- For an attack $y+1$ message blocks are required
- This attack strategy does not work for SMASH-256 and SMASH-512 due to number of maximum message blocks

SMASH-256

- Specific instance of SMASH design method
- SMASH-256 specified by
  - n = 256
  - GF(2^{256}) defined by $q(\theta) = \theta^{256} + \theta^{16} + \theta^3 + \theta + 1$
  - compression function *f*

$$f = H_1 \circ H_3 \circ H_2 \circ L \circ H_1 \circ H_2 \circ H_3 \circ L \circ H_2 \circ H_1 \circ H_3 \circ L \circ H_3 \circ H_2 \circ H_1$$

- The finite field element θ is defined as root of $q(\theta)$
- Due to chosen padding method, SMASH-256 can process messages with bit length less than 2^{128} bits (after padding)

Applying previous attack on SMASH-256

- Since θ is defined as root of $q(\theta)$ we get

$$ord(1+\theta) = ((2^{256} - 1)/5)$$

- For attack $((2^{256} - 1)/5) + 1$ message blocks required
- SMASH-256 can hash up to $2^{120} - 1$ message blocks
  - still possible to construct colliding messages but
  - these messages are no longer valid inputs according to SMASH-256 specification
- Same holds for SMASH-512

Extending attack method to break SMASH-256

- So far we extended forward prediction property
  - Introduce difference x once at the beginning and once at the end
- Extension of attack
  - Introduce non-zero difference x three or more times
  - If input difference to f is non zero we want to have same absolute values as in first iteration
  - In this cases output difference is always

$$(a + \theta x)$$

  - We construct a difference of the form

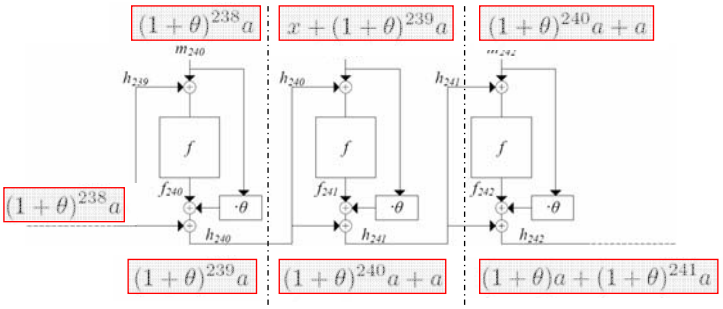
$$h'_t = a \cdot q(\theta) = a \cdot 0$$

Breaking SMASH-256

- For a collision we need

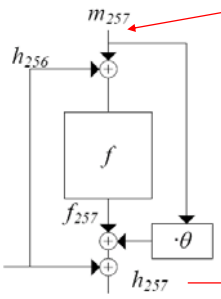
$$a \cdot q(\theta) = (1+\theta)^{256}a + (1+\theta)^{16}a + (1+\theta)^{3}a + (1+\theta)^{2}a + a$$

- Constructing the polynomial



Breaking SMASH-256 cont'd

- Introduce non-zero difference x in $i = 1, 241, 254, 255, 257$
  - 257 message blocks needed
  - 4 message blocks determined by attack
  - 253 message blocks can be chosen arbitrarily



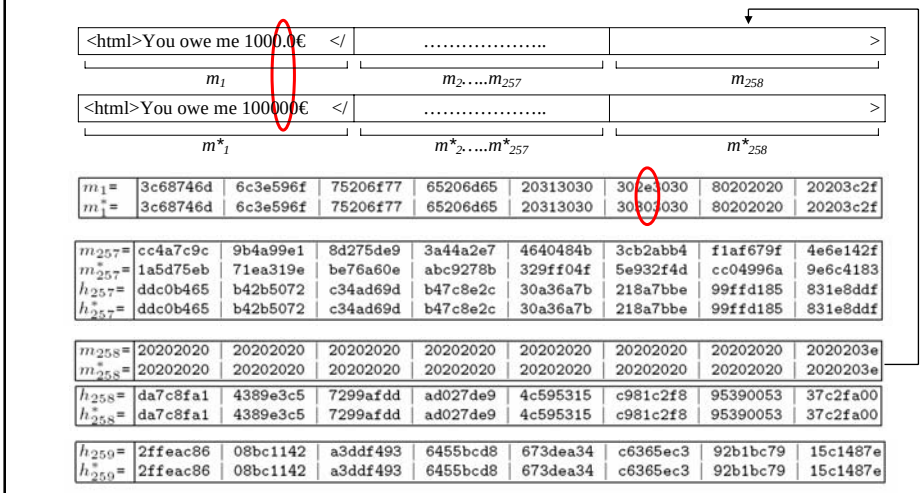
$$x + (1+\theta)^{255}a + (1+\theta)^{15}a + (1+\theta)^{2}a + (1+\theta)a$$

$$\theta^{256} + \theta^{16} + \theta^{3} + \theta + 1$$
$$= (1+\theta)^{256} + (1+\theta)^{16} + (1+\theta)^{3} + (1+\theta)^{2} + 1$$

$$(1+\theta)^{256}a + (1+\theta)^{16}a + (1+\theta)^{3}a + (1+\theta)^{2}a + a$$

=> collision after iteration 257

Sample collision



m_1=	3c68746d	6c3e596f	75206f77	65206d65	20313030	30e30030	80202020	20203c2f
m^*_1=	3c68746d	6c3e596f	75206f77	65206d65	20313030	30307030	80202020	20203c2f
m_{257}=	cc4a7c9c	9b4a99e1	8d275de9	3a44a2e7	4640484b	3cb2abb4	f1af679f	4e6e142f
m^*_{257}=	1a5d75eb	71ea319e	be76a60e	abc9278b	329ff04f	5e932f4d	cc04996a	9e6c4183
h_{257}=	ddc0b465	b42b5072	c34ad69d	b47c8e2c	30a36a7b	218a7bbe	99ffd185	831e8ddf
h^*_{257}=	ddc0b465	b42b5072	c34ad69d	b47c8e2c	30a36a7b	218a7bbe	99ffd185	831e8ddf
m_{258}=	20202020	20202020	20202020	20202020	20202020	20202020	20202020	2020203e
m^*_{258}=	20202020	20202020	20202020	20202020	20202020	20202020	20202020	2020203e
h_{258}=	da7c8fa1	4389e3c5	7299afdd	ad027de9	4c595315	c981c2f8	95390053	37c2fa00
h^*_{258}=	da7c8fa1	4389e3c5	7299afdd	ad027de9	4c595315	c981c2f8	95390053	37c2fa00
h_{259}=	2ffeac86	08bc1142	a3ddf493	6455bcd8	673dea34	c6365ec3	92b1bc79	15c1487e
h^*_{259}=	2ffeac86	08bc1142	a3ddf493	6455bcd8	673dea34	c6365ec3	92b1bc79	15c1487e

Conclusion

- Attack on the SMASH design strategy
  - SMASH-256 and SMASH-512
  - All hash functions that follow the design strategy
- Two observations that make attack possible
  - Forward prediction property
  - Pattern construction property
- Ad-hoc fix (?)
  - Add counter value => compression function is not constant over iterations