What's the Potential Danger Behind the collisions of Hash Functions

Xiaoyun Wang[1,2]

[1] School of Mathematics & System Science, Shandong University

[2] Center for Advanced Study, Tsinghua University

06/21/2005

Outline

- Cryptology and Information Security
- Hash Functions and Cryptology
- Introduction to Hash Function
- Application of Hash Function
- Dedicated Hash Functions
- Cryptanalysis on Hash Functions
- Colliding X.509 Certificates
- The Meaningful Collision Attack for Hash Functions
- The Second-Preimage Attack for Weak Message and MAC

Cryptology and Information Security

- Information Security in Computer Network

  Privacy, Integrity, legality, Efficiency, No disavowing
- Cryptology is the key technique in Information Security

  Privacy: Encryption: Public Key or Symmetric Ciphers

  Integrity：Hash Functions

  Legality: Digital signature and Authentication (Based on hash function and hard mathematics problems such as factorization, discrete logarithm etc.

  Efficiency:

Hash Functions and Cryptology

- Hash Function is an important technique in Information Security.
- Hash function is a fundamental tool in cryptology.

1 To guarantee the data integrity in the message transfer.

2 To guarantee the security of digital signatures(no forgery).

3 Used to design many cryptographic algorithms and protocols.

For example, digital signature , group signature, threshold signature, e-cash, e-vote, bit-commitment, many other provable-security cryptosystems.

Introduction to Hash Function

- **Hash Function: a compress function Y**=H(M) which hash any message with arbitrary length into a fixed length output:

  $H(M): M \in \{0,1\}^* \rightarrow \{0,1\}^l$

- One-way property： Given any Y=H(M), it is infeasible to get any substantial information of M. 2^l The ideal strength computations.
- Second-Preimage Resistance: Given any message M_1 2^l, it is difficult to find another message M_2 such that : $H(M_1)=H(M_2)$,
- Free-Collision: It is difficult to find two different messages(M_1, M_2) with the same hash value:

  $H(M_1)=H(M_2)$。 Birthday attack: $2^{\frac{l}{2}}$

Application of Hash Function

--Hash Function and Signature-1

H(M): hash function

S(M): signature algorithm

Signing process:

- Compute the fingerprint (or digest) of message:

$$M \xrightarrow{H} H(M)$$

- Signing the fingerprint H(M): s=S(H(M))
- If the fingerprint of M_1 is the same as another different M_2

$$H(M_1)=H(M_2)$$

Then M_1 and M_2 have the same signatures

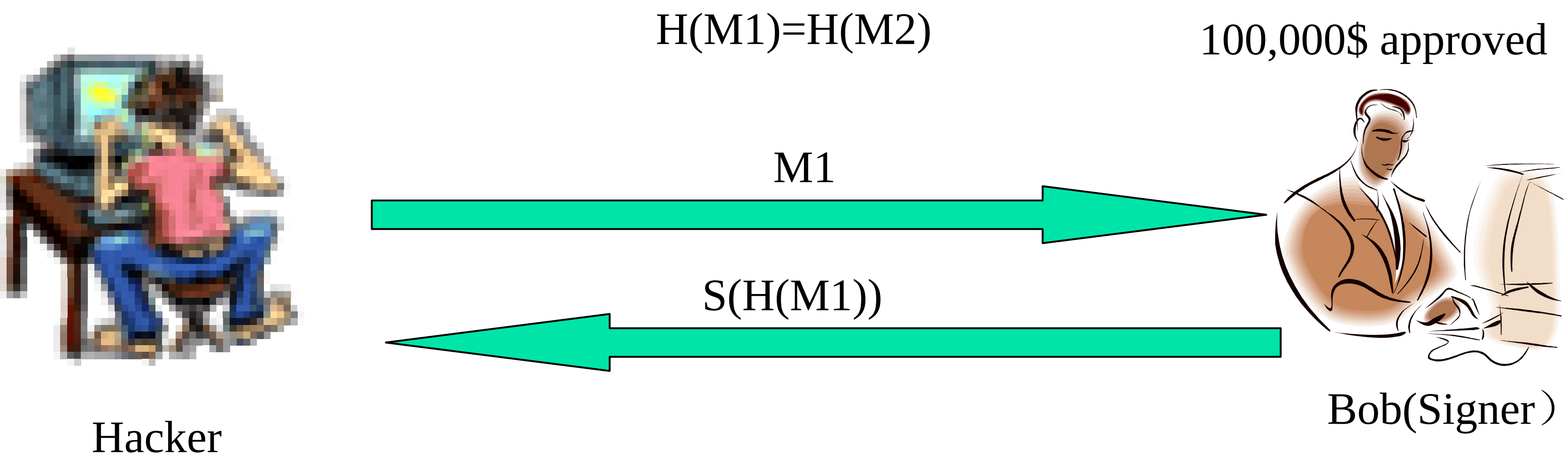
$$S(H(M_1))=S(H(M_2))$$

Application of Hash Function

--Hash Function and Signature-2

M_1=(project application 1+apllication fund100,000$)

M_2=(project application 2+application fund1000,000$)



H(M1)=H(M2)

100,000$ approved

M1

S(H(M1))

Hacker

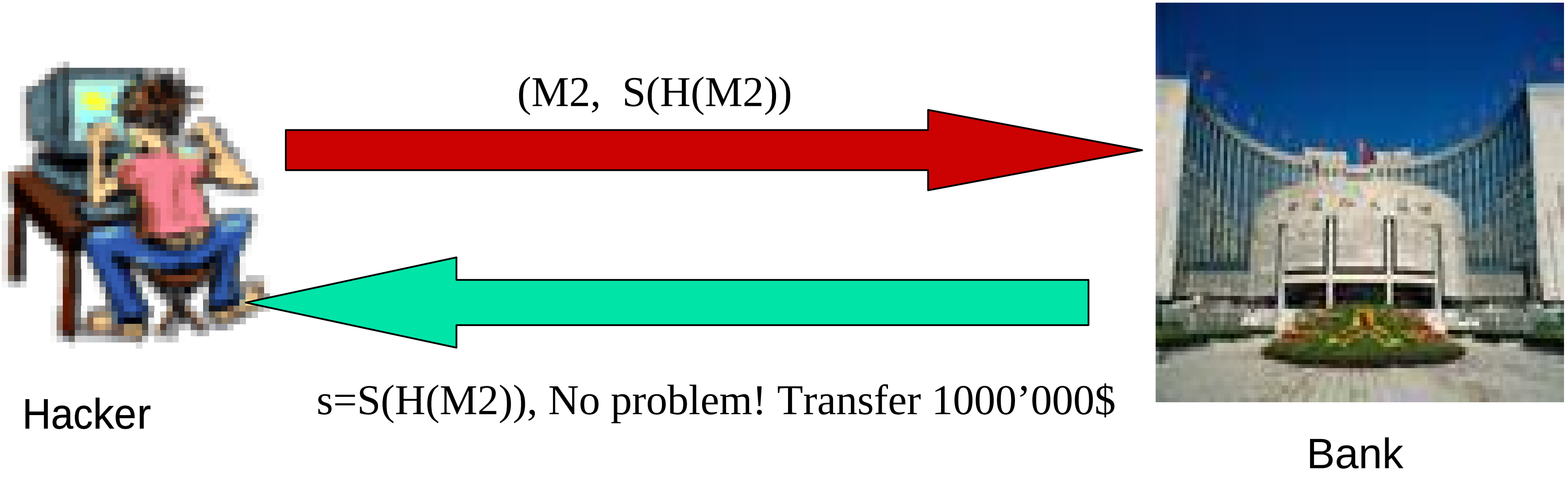
Bob(Signer）

Bob has signed both messages M1 and M2 because of S(H(M1)) ＝S(H(M2))

Hacker prepares two application versions for a project in advance

Application of Hash Function

--Hash Function and Signature-3



(M2, S(H(M2))

s=S(H(M2)), No problem! Transfer 1000’000$

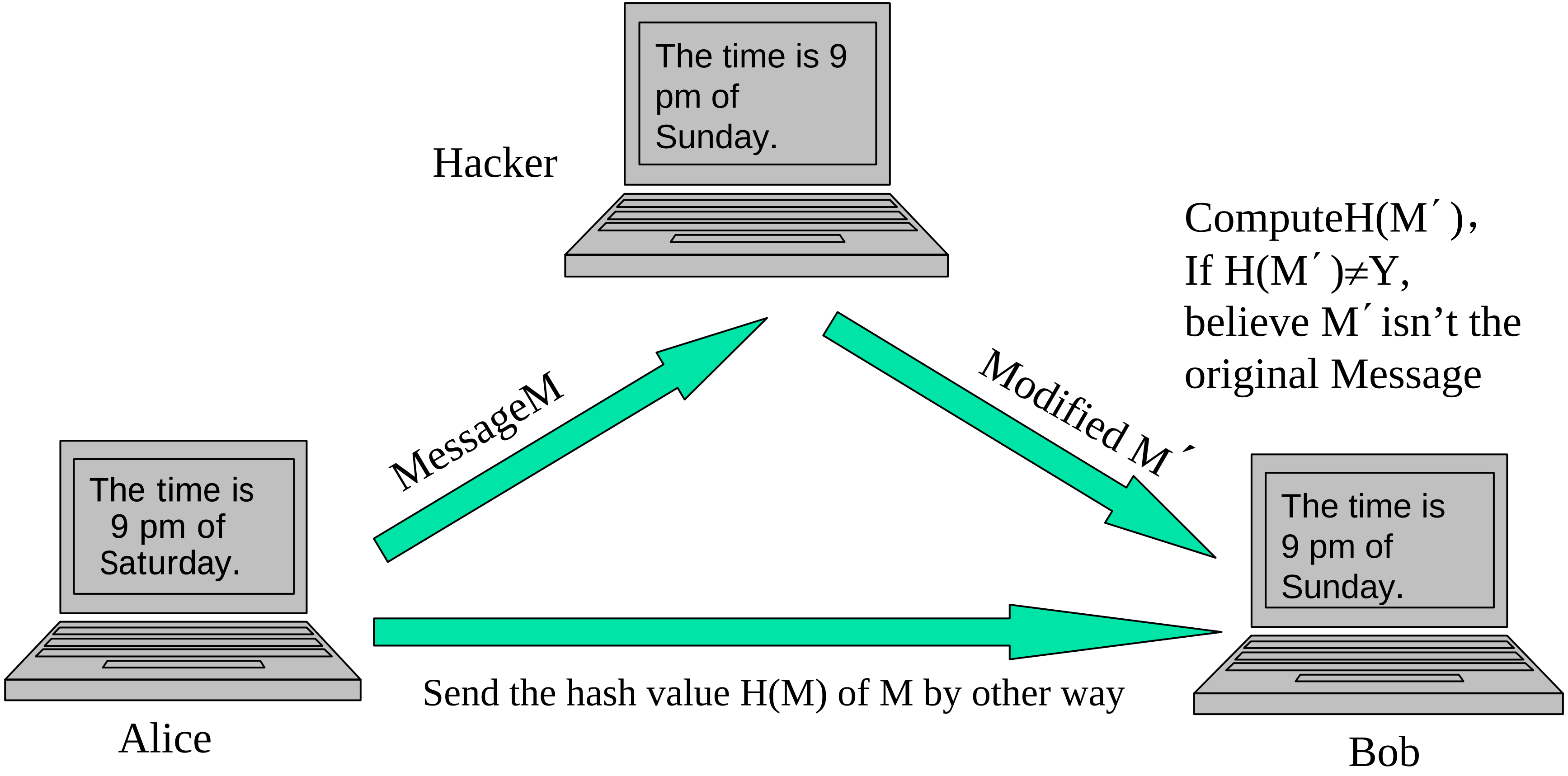
Hacker

Bank

Hacker withdraws 1000,000$
with the forged signature

Application of Hash Function

--Hash Function and Data Integrity


Hacker
The time is 9 pm of Sunday.
ComputeH(M´)，
If H(M´)≠Y,
believe M´ isn't the original Message
MessageM
Modified M´
The time is 9 pm of Saturday.
The time is 9 pm of Sunday.
Send the hash value H(M) of M by other way
Alice
Bob

Application of Hash Function

--Knowledge Proof Based on **Hash Function**

Prover P: Know a secret, for example:

$y=g^x \bmod p$, y: public; x: secret

Verifier V: To verify Prover that P knows the secret x, but cannot get the substantial information about x.

H(M): One-way hash function

$c=H(y*g*g^s y^c) \quad (**)$

P: computes (c,s) satisfies the equation (**), send (c, s) to Verifier.

V: believe that P knows the secret x if the equation ** holds.

Application of Hash Function

--Knowledge Proof Based on **Hash Function(Cont.)**

- Utilizing the knowledge proof based on hash function, many cryptographic algorithm and protocols are constructed:

  Signature, group signature, threshold signature,

  e-cash, bit commitment etc.

Dedicated Hash Functions

- Before 1990: Hash functions based on block ciphers
  Since 1990: Dedicated hash functions (constructed directly)
- Two kinds of dedicated hash functions
- MDx (Rivest)： MD4, MD5, HAVAL, RIPEMD,
  RIPEMD-160.
- SHAx (NIST)： SHA-0, SHA-1, SHA-256, 384, 512

Two widely used hash functions in the world: MD5, SHA-1。

Cryptanalysis on Hash Functions
---Earlier Work on MDx

1 1993: Boer and Bosselaers found one message with two different sets of initial values.

2 1996: Dobbertin found a collision attack on MD4 with probability 2^{-22} (FSE’96).

3 1996: Dobbertin gave a psuodrandom collision example of MD5 which is two messages with another set of initial values (Eucrypt’96: Rump session).

4 2003: Rompay etc: collision attack with probability 2^{-29} (Asiacrypt’03).

Cryptanalysis on Hash Functions --- Wang etc Collision Attacks on MDx

In Crypto’04, Wang announced some collision examples on a series of hash functions.

1 MD5: Finding a collision with probability 2^{-37} (2004).

2 MD4: Finding a collision with probability 2^{-2}-2^{-6}.

3 RIPEME: Finding a collision with probability of 2^{-19}.

4 HAVEL-128: Finding a collision with probability of 2^{-7}.

Cryptanalysis on Hash Functions
---Earlier work on SHA-0

- 1997: Wang gave an algebraic method attack to find collision with probability 2^{-58} .

  Circulated in China, wrote in Chinese.
- 1998: Chabaud and Joux found a collision attack with probability: 2^{-61}.
- 1998: Improved to about 2^{-45} by message modification

Cryptanalysis on Hash Functions
--- Latest Work on SHA-0 and SHA-1

- Joux: A four-block message collision was found by Joux in August which took about 80,000 hours of CPU time equivalent to the complexity 2^{51} (Crypt'04 Rump session and Eurocrypt'05).
- Biham and Chen: Found real collisions of SHA-1 up to 40 steps, and estimated that collisions of SHA-1 can be found up to 53-round reduced SHA-1 with complexity 2^{48}, where the reduction is to the last 53 rounds of SHA-1. (Crypt'04 Rump session and Eurocrypt'05).
- Wang, Yin and Yu (Feb of 2005): Find a collision of SHA-1 with probability 2^{-69}. This is the first attack faster than the birthday attack 2^{-80} (To appear in Crypt'05).
- Wang, Yu and Yin (Feb of 2005): Find a collision of SHA-0 with probability 2^{-39} (To appear in Crypt'05) .

Colliding Valid X.509 Certificates

- A. Lenstra, X.Y. Wang, B. Weger

  http://eprint.iacr.org/2005/067.pdf
- Constructing a pair of valid X.509 certificates in which the “to be signed parts” is a collision for MD5.
- Two certificates are different public keys for an owner.
- The issuing Certificates Authority cannot prove the right key possession.

Meaningful Collisions for MD5

- Stefan Lucks and Magnus Daum (Rump Session in Eurocrypt'05)
  http://th.informatik.unimannheim.de/people/lucks/HashCollisions/
  http://www.cits.rub.de/MD5Collisions/

  - M_1: A Recommendation Letter for Alice from the Boss Caesar
  - M_2: A Order Letter for Alice's privilege from the Boss Caesar
  - Two letters have the same signature because of

$$H(M_1)=H(M_2)$$

The Second-Preimage Attack of Weak Messages

Wang, Lai etc results in Eurocrypt’05:

- Any message is a weak message of MD4 with probability 2^{-122}, and for a weak message it only need one-MD4 computation to find its second-preimage.
- Any message M can be modified with the basic message modification techniques. The resulting message M_0 is a weak message with probability 2^{-23}. M and M_0 are close and the Hamming weight of the difference for two messages is 50 on average.
- Under the advanced message modification, any message M can be modified into M_0 which is a weak message with probability 2^{-2} to 2^{-6}. However, the Hamming weight of the their difference grows quickly up to 110.

The Second-Preimage Attack of Weak Messages

Yu and Wang etc (Recent work):

- Any message is a weak message with probability 2^{-56} by a new collision differential path (See Table 1 and Table 2) .
- By message modifications techniques, any message can be converted into a weak message with 2^{27} MD4 computations, the Hamming weight for their difference is 44

Constructing MAC based-MD4

Three basic proposals to construct a MAC based-MD4

- Secret prefix: $MAC(M)=MD_4(K_1||M)$
- Secret suffix: $MAC(M)=MD_4(M||K_2)$
- Envelope: $MAC(M)=MD_4(K_1||M||K_2)$

Key Recovery from Hash Function Collisions
--- Existing Attack for Key Recovery of Envelope Method

Bart Preneel and Paul C.van Oorschat :

- It needs $2^{n/2}$ known text-MAC pairs and 2^{k1} offline compression function operations to recovery the key $K_{1,}$, and 2^{k2} computations (exhaustive search) for recovery of K_2.

- Choosing $K_1 \neq K_2$ does not offer additional security property. So they suggested

$$K_1=K_{2.}$$

Key Recovery from Hash Functions Collisions --- Effective Key Recovery of Envelope MAC Based on MD4

Wang and Yu (recent work):

Suppose $K_1=K_2=K$

Case 1 K is 128-bit.

Case 2 If K is a complete block, we deduce the 128-bit secret IV instead of finding K.

So we suppose K has 128-bit length.

Key Recovery from Hash Functions Collisions
-- Effective key recovery of envelope MAC based on MD4 (Cont.))

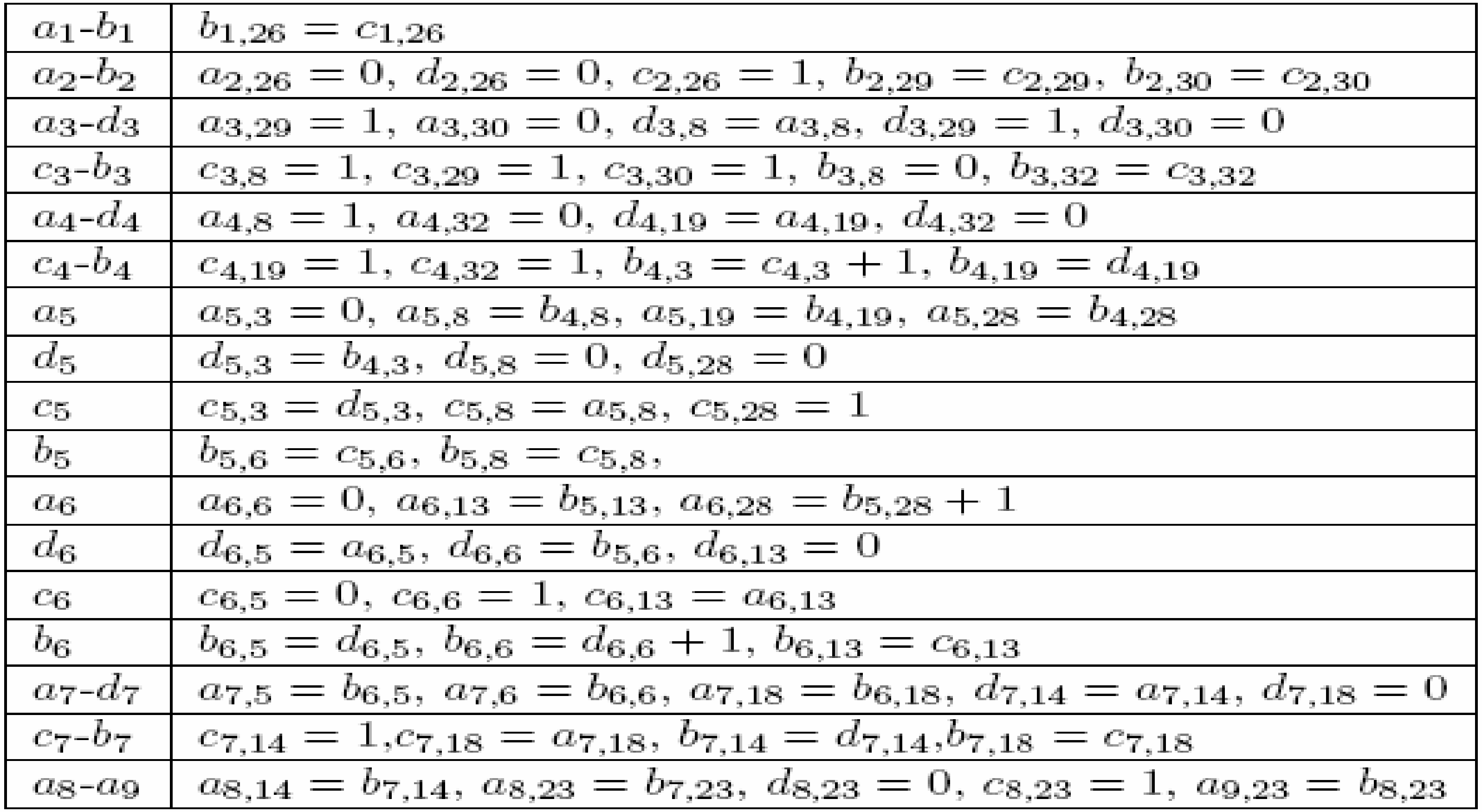
- Determine one bit condition $b_{1,26}= c_{1,26}$ in Table 2 with one computations and 2^{62} MACs of random 384-bit messages M and their corresponding chosen 384-bit messages' MAC(M') , where difference is:

  (K||M)-(K|| M')=(0, 0, 0, 0, 2^{22}, 0,……, 0) (See Table 1)

- Determine other conditions $b_{1,i+4}= c_{1,i+4}$ by one computation and the same number of MAC pairs with the similar collision differential path determined by the difference:

  (K||M)-(K|| M')=(0, 0, 0, 0, 2^{i}, 0,……, 0) ($i\neq22$).

- Totally determine 32 conditions $b_{1,i+4}= c_{1,i+4}$ by 2^{68} MAC pairs and 32 computations.



a_1-b_1	$b_{1,26} = c_{1,26}$
a_2-b_2	$a_{2,26} = 0,\ d_{2,26} = 0,\ c_{2,26} = 1,\ b_{2,29} = c_{2,29},\ b_{2,30} = c_{2,30}$
a_3-d_3	$a_{3,29} = 1,\ a_{3,30} = 0,\ d_{3,8} = a_{3,8},\ d_{3,29} = 1,\ d_{3,30} = 0$
c_3-b_3	$c_{3,8} = 1,\ c_{3,29} = 1,\ c_{3,30} = 1,\ b_{3,8} = 0,\ b_{3,32} = c_{3,32}$
a_4-d_4	$a_{4,8} = 1,\ a_{4,32} = 0,\ d_{4,19} = a_{4,19},\ d_{4,32} = 0$
c_4-b_4	$c_{4,19} = 1,\ c_{4,32} = 1,\ b_{4,3} = c_{4,3} + 1,\ b_{4,19} = d_{4,19}$
a_5	$a_{5,3} = 0,\ a_{5,8} = b_{4,8},\ a_{5,19} = b_{4,19},\ a_{5,28} = b_{4,28}$
d_5	$d_{5,3} = b_{4,3},\ d_{5,8} = 0,\ d_{5,28} = 0$
c_5	$c_{5,3} = d_{5,3},\ c_{5,8} = a_{5,8},\ c_{5,28} = 1$
b_5	$b_{5,6} = c_{5,6},\ b_{5,8} = c_{5,8},$
a_6	$a_{6,6} = 0,\ a_{6,13} = b_{5,13},\ a_{6,28} = b_{5,28} + 1$
d_6	$d_{6,5} = a_{6,5},\ d_{6,6} = b_{5,6},\ d_{6,13} = 0$
c_6	$c_{6,5} = 0,\ c_{6,6} = 1,\ c_{6,13} = a_{6,13}$
b_6	$b_{6,5} = d_{6,5},\ b_{6,6} = d_{6,6} + 1,\ b_{6,13} = c_{6,13}$
a_7-d_7	$a_{7,5} = b_{6,5},\ a_{7,6} = b_{6,6},\ a_{7,18} = b_{6,18},\ d_{7,14} = a_{7,14},\ d_{7,18} = 0$
c_7-b_7	$c_{7,14} = 1, c_{7,18} = a_{7,18},\ b_{7,14} = d_{7,14}, b_{7,18} = c_{7,18}$
a_8-a_9	$a_{8,14} = b_{7,14},\ a_{8,23} = b_{7,23},\ d_{8,23} = 0,\ c_{8,23} = 1,\ a_{9,23} = b_{8,23}$

Table 2: A set of sufficient conditions for the MD4 differential path.

Key Recovery from Hash Functions Collisions
-- Effective key recovery of envelope MAC based on MD4 (Cont.)

- It is possible to determine more bit conditions of a_1, d_1 , c_1 and b_1 by other collision differential paths.

- Provided that we have found s ($s \geq 32$) conditions for a_1, d_1 , c_1 and b_1 , we search for other 128-s bits, and then compute 128-bit K.

Key Recovery from Hash Functions Collisions
-- Effective key recovery of envelope MAC based on MD4 (Cont.)

Conclusion:

- Determine the secret key K with about 2^{96} computations and 2^{68} MAC pairs with 32 collision differential paths determined by the difference:

  (K||M)-(K|| M')=(0, 0, 0, 0, 2^i, 0,……, 0)

- The above result can be improved to determine the secret key K with about 2^{96-r} computations with more collision differential paths and more MAC pairs.

Thanks!