Recent Advances in Meet-in-the-Middle Attacks

Jian Guo

ASK 2012, 27 August 2012

Talk Overview

Introduction

Problem

Given functions $f : D_1 \longrightarrow R$ and $g : D_2 \longrightarrow R$, find $x \in D_1$ and $y \in D_2$ so that $f(x) = g(y)$.

Introduction

Problem

Given functions $f : D_1 \longrightarrow R$ and $g : D_2 \longrightarrow R$, find $x \in D_1$ and $y \in D_2$ so that $f(x) = g(y)$.

MITM attacks

Randomly pick a list L_1 of x's from D_1 and a list L_2 of y's from D_2, and compute the sets $R_1 = \{f(x) \mid x \in L_1\}$ and $R_2 = \{g(y) \mid y \in L_2\}$. When $|R_1|x|R_2| \geq |R|$, the chance that there is at least one common element in R_1 and R_2 becomes non-negligible.

Introduction

Problem

Given functions $f : D_1 \longrightarrow R$ and $g : D_2 \longrightarrow R$, find $x \in D_1$ and $y \in D_2$ so that $f(x) = g(y)$.

MITM attacks

Randomly pick a list L_1 of x's from D_1 and a list L_2 of y's from D_2, and compute the sets $R_1 = \{f(x) \mid x \in L_1\}$ and $R_2 = \{g(y) \mid y \in L_2\}$. When $|R_1|x|R_2| \geq |R|$, the chance that there is at least one common element in R_1 and R_2 becomes non-negligible.

Locating such collision is usually done by sorting the elements of R_1 and/or R_2 in lookup tables, the minimum memory requirement is $\min(|R_1|, |R_2|)$, and $(|R_1| + |R_2|)$ computations.

Birthday Attack and its Memory Requirement

Birthday Attack

Given function $f : D \longrightarrow R$, find $x, y \in D$ and $x \neq y$ such that $f(x) = f(y)$. Randomly pick x from D, compute $f(x)$ and store the pair $(x, f(x))$ in a table, repeat until a collision on $f(x)$ is hit. With probability $1/2$, it is expected to repeat $1.18 \times |R|^{1/2}$ times before hitting a collision, and hence memory requirement is in the order of $|R|^{1/2}$.

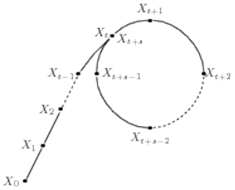
Birthday Attack and its Memory Requirement

Birthday Attack

Given function $f : D \longrightarrow R$, find $x, y \in D$ and $x \neq y$ such that $f(x) = f(y)$. Randomly pick x from D, compute $f(x)$ and store the pair $(x, f(x))$ in a table, repeat until a collision on $f(x)$ is hit. With probability $1/2$, it is expected to repeat $1.18 \times |R|^{1/2}$ times before hitting a collision, and hence memory requirement is in the order of $|R|^{1/2}$.

Floyd's cycle-finding algorithm [8] in 1967

When $D = R$, randomly choose $X_0 \in D$, define $X_i = f(X_{i-1})$ (i.e., $X_i = f^i(X_0)$) for $i = 1, 2, 3, \ldots$, there exist $s, t \in \mathbb{Z}^+$ such that $X_t = X_{t+s}$.
There exists j such that $f^{2j}(X_0) = f^j(X_0)$ (tortoise and the hare).



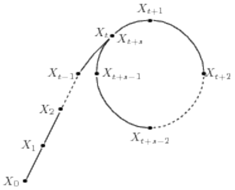
Birthday Attack and its Memory Requirement

Birthday Attack

Given function $f : D \longrightarrow R$, find $x, y \in D$ and $x \neq y$ such that $f(x) = f(y)$. Randomly pick x from D, compute $f(x)$ and store the pair $(x, f(x))$ in a table, repeat until a collision on $f(x)$ is hit. With probability $1/2$, it is expected to repeat $1.18 \times |R|^{1/2}$ times before hitting a collision, and hence memory requirement is in the order of $|R|^{1/2}$.

Floyd's cycle-finding algorithm [8] in 1967

When $D = R$, randomly choose $X_0 \in D$, define $X_i = f(X_{i-1})$ (i.e., $X_i = f^i(X_0)$) for $i = 1, 2, 3, \ldots$, there exist $s, t \in \mathbb{Z}^+$ such that $X_t = X_{t+s}$.
There exists j such that $f^{2j}(X_0) = f^j(X_0)$ (tortoise and the hare).



Price: Time Complexity: $3 \cdot |R|^{1/2}$, more Time-Memory trade-off in [21].

Parallel Birthday Attack with Distinguished Points

Distinguished Point [22] 1999

values with distinguished properties, e.g., last z bits are '0's.

$$D(y) = \begin{cases} 1 & \text{if } y \text{ is distinguished point, with probability } 2^{-z} \\ 0 & \text{otherwise} \end{cases}$$

Parallel Birthday Attack with Distinguished Points

Distinguished Point [22] 1999

values with distinguished properties, e.g., last z bits are '0's.

$$D(y) = \begin{cases} 1 & \text{if } y \text{ is distinguished point, with probability } 2^{-z} \\ 0 & \text{otherwise} \end{cases}$$

Parallel Attack

1. Randomly choose x_0, and compute trail $x_{i+1} = f(x_i)$ for $i = 0, 1, 2, \cdots$ until a distinguished point x_j is hit ($D(x_j) = 1$), store only (x_0, x_j).

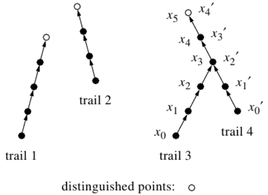


figure credit: [22]

Parallel Birthday Attack with Distinguished Points

Distinguished Point [22] 1999

values with distinguished properties, e.g., last z bits are '0's.

$$D(y) = \begin{cases} 1 & \text{if } y \text{ is distinguished point, with probability } 2^{-z} \\ 0 & \text{otherwise} \end{cases}$$

Parallel Attack

1. Randomly choose x_0, and compute trail $x_{i+1} = f(x_i)$ for $i = 0, 1, 2, \cdots$ until a distinguished point x_j is hit ($D(x_j) = 1$), store only (x_0, x_j).

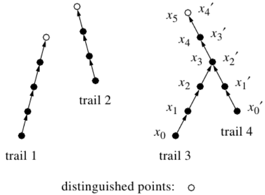


figure credit: [22]

Complexties: Memory $2^{-z} \cdot |R|^{1/2}$, p parallel nodes with $p < 2^{-z} \cdot |R|^{1/2}$, Time: linear speedup, i.e., $|R|^{1/2}/p$ for each node.

MITM Attacks: Memory and Parallelization

Morita-Ohta-Miyaguchi [19] 1991: $f, g : D \longrightarrow R$ with $D = R$, define a random function $s : D \longrightarrow \{0, 1\}$, and

$$T(x) = \begin{cases} f(x) & \text{if } s(x) = 0 \\ g(x) & \text{else } s(x) = 1 \end{cases}$$

Apply Floyd's cycle finding algorithm to T, with probability $1/2$, collision of T is collision of f and g.

MITM Attacks: Memory and Parallelization

Morita-Ohta-Miyaguchi [19] 1991: $f, g : D \longrightarrow R$ with $D = R$, define a random function $s : D \longrightarrow \{0, 1\}$, and

$$T(x) = \begin{cases} f(x) & \text{if } s(x) = 0 \\ g(x) & \text{else } s(x) = 1 \end{cases}$$

Apply Floyd's cycle finding algorithm to T, with probability $1/2$, collision of T is collision of f and g.

Parallel computation with distinguished points applies too.

MITM Attacks: Memory and Parallelization

Morita-Ohta-Miyaguchi [19] 1991: $f, g : D \longrightarrow R$ with $D = R$, define a random function $s : D \longrightarrow \{0, 1\}$, and

$$T(x) = \begin{cases} f(x) & \text{if } s(x) = 0 \\ g(x) & \text{else } s(x) = 1 \end{cases}$$

Apply Floyd's cycle finding algorithm to T, with probability $1/2$, collision of T is collision of f and g.

Parallel computation with distinguished points applies too.

Open Question: what if $D \neq R$?

Application to DES

double DES with different keys (*i.e.,* $C = DES_{K2}(DES_{K1}(P))$).

Application to DES

double DES with different keys (*i.e.*, $C = DES_{K2}(DES_{K1}(P))$).

Diffie-Hellman [5] 1977: one can carry out MITM attack on the functions $DES_{K1}(\cdot)$ and $DES^{-1}_{K2}(\cdot)$.

Application to DES

double DES with different keys (*i.e.*, $C = DES_{K2}(DES_{K1}(P))$).

Diffie-Hellman [5] 1977: one can carry out MITM attack on the functions $DES_{K1}(\cdot)$ and $DES^{-1}_{K2}(\cdot)$.

Merkle-Hellman [18] 1981: application to triple-DES $C = DES_{K1}(DES_{K2}(DES_{K1}(P)))$, MITM on functions $DES^{-1}_{K1}(ENC(DES^{-1}_{K1}(\cdot)))$ and $DES_{K2}(\cdot)$

Application to DES

double DES with different keys (*i.e.,* $C = DES_{K2}(DES_{K1}(P))$).

Diffie-Hellman [5] 1977: one can carry out MITM attack on the functions $DES_{K1}(\cdot)$ and $DES^{-1}_{K2}(\cdot)$.

Merkle-Hellman [18] 1981: application to triple-DES $C = DES_{K1}(DES_{K2}(DES_{K1}(P)))$, MITM on functions $DES^{-1}_{K1}(ENC(DES^{-1}_{K1}(\cdot)))$ and $DES_{K2}(\cdot)$

— all in mode level, i.e., regardless of the internal details of the cipher.

Application to Reduced DES/AES

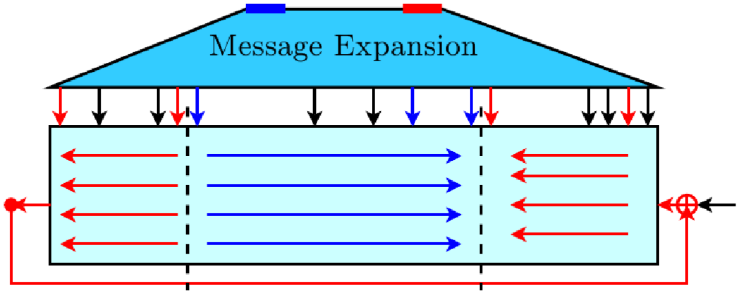
Attacking the details of the cipher:

- Chaum-Evertse [4] 1985: Application to 6-7 rounds of DES.
- Dunkelman-Sekar-Preneel [7] 2007 :Improved Results with similar rounds.

- Many attacks against AES, e.g., Dunkelman-Keller-Shamir [6] 2010: 7-round AES-128

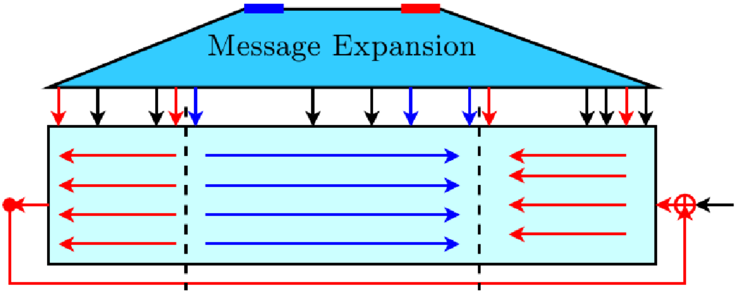
Basic Attack against Compression Functions

When the compression function follows Davies-Meyer, i.e., $H' = E_M(H) \oplus H$.



Basic Attack against Compression Functions

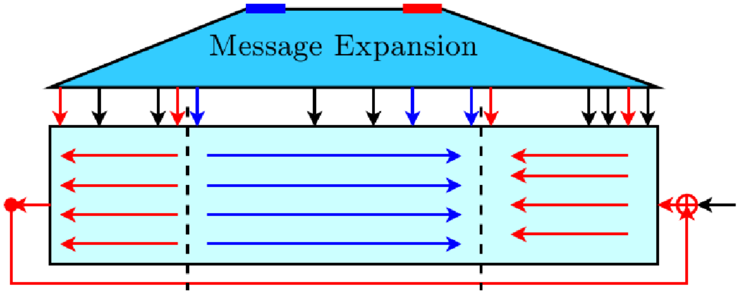
When the compression function follows Davies-Meyer, i.e., $H' = E_M(H) \oplus H$.



Complexities: with l neutral bits, Time 2^{n-l} & Memory 2^l.

Basic Attack against Compression Functions

When the compression function follows Davies-Meyer, i.e., $H' = E_M(H) \oplus H$.

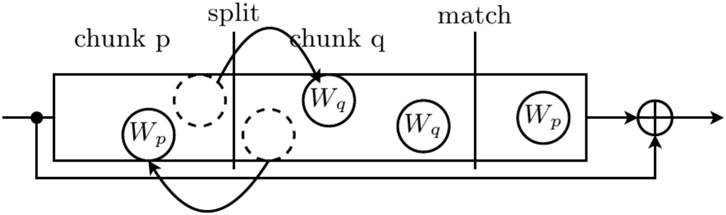


Complexities: with l neutral bits, Time 2^{n-l} & Memory 2^l.

Limitations: the number of steps that can be attacked is very limited.

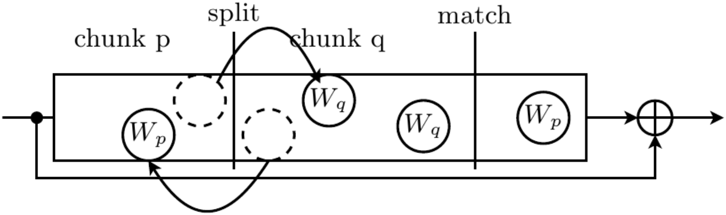
How to Attack More Steps

Initial Structure (Biclique?)

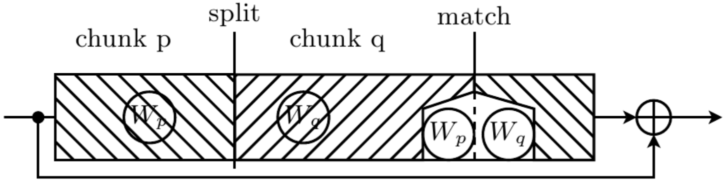


How to Attack More Steps

Initial Structure (Biclique?)



Partial Matching and its Variants



Techniques Developed

- Splice-and-Cut
- Initial Structure, Probabilistic Initial Structure, Bicliques
- Partial Matching, Indirect Partial Matching, Probabilistic Partial Matching, Partial Matching with Differential View (Fuzzy Matching)

Initial Structure

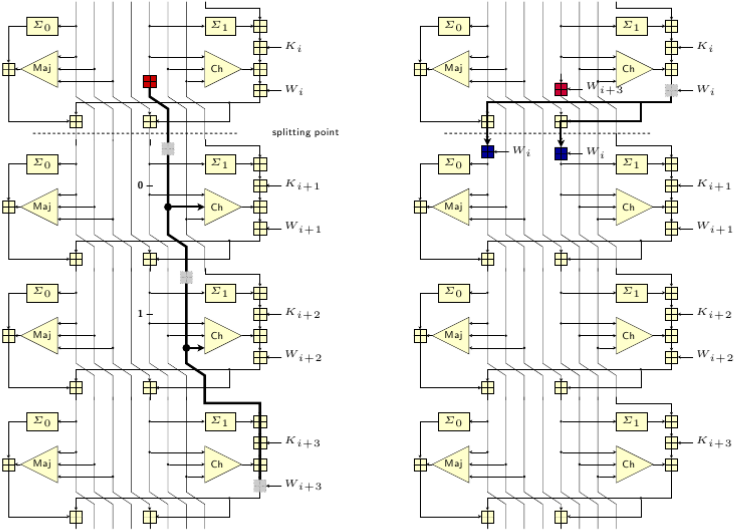


Figure: 4-step Initial Structure - An example of SHA-2

Initial Structure: Trade-off between Neutral Bits and Steps

The case of SHA-2

- 2 steps: 32 bits
- 3 steps: 16 bits
- 4 steps: 11 bits [9]
- 6 steps: 3 bits (biclique) [14]

More steps ⇔ Less Neutral bits ⇔ Higher Time Complexity & Less Memory Requirement

Initial Structure: Trade-off between Neutral Bits and Steps

The case of SHA-2

- 2 steps: 32 bits
- 3 steps: 16 bits
- 4 steps: 11 bits [9]
- 6 steps: 3 bits (biclique) [14]

More steps ⇔ Less Neutral bits ⇔ Higher Time Complexity & Less Memory Requirement

Difference between Initial Structure and Biclique?

> a single key, and in our opinion they have smaller potential. Indeed, even a single operation for each key implies a lower bound on the complexity which is not far from exhaustive search. Also from the technical point of view, the use of bicliques in those settings is not much different from earlier use of initial structures.

From [12]

Probabilistic Initial Structure

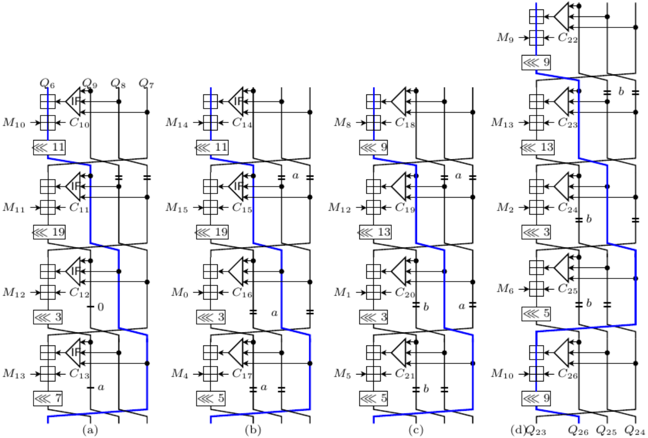


Figure: 17-step Initial Structure with Prob. 2^{-8} - An example of MD4

Tradeoff between Time Complexity and Attacked Steps.

(Indirect) Partial Matching

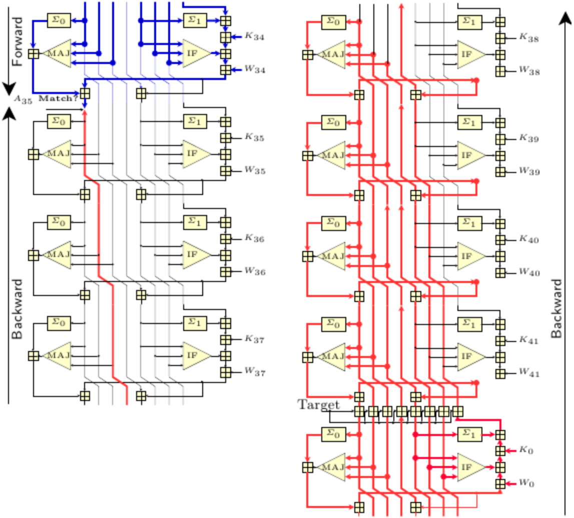


Figure: 9-step Indirect Partial Matching for SHA-2 with 32 matching bits

Partial Matching: Trade-off

more matching steps
$\Leftrightarrow$ less matching bits
$\Leftrightarrow$ (maybe) higher time complexity

Partial Matching: Trade-off

more matching steps
$\Leftrightarrow$ less matching bits
$\Leftrightarrow$ (maybe) higher time complexity

Overall: it is natural that one can attack more steps (more steps for both initial structure and partial matching) with less neutral bits, which results in higher time complexity.

Mode of Operations

[20] 2011: first application of MITM attack using the state value alone, regardless of the key/key schedule.

Mode of Operations

[20] 2011: first application of MITM attack using the state value alone, regardless of the key/key schedule.
[3] 2010: notation of 3-subset MITM attack

Mode of Operations

[20] 2011: first application of MITM attack using the state value alone, regardless of the key/key schedule.
[3] 2010: notation of 3-subset MITM attack

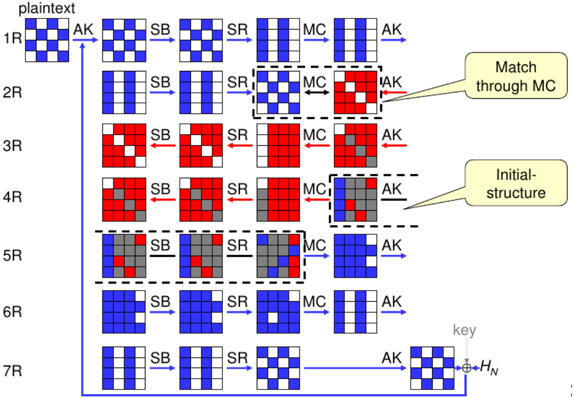
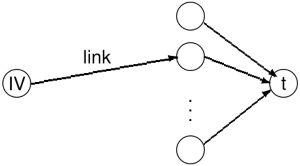


figure credit: Sasaki [20], results in preimage attack in DM mode, and second-preimage attack in MMO/MP modes.

Converting Pseudo-Preimages to Preimages

Large Precomputations	Storage 2^{n-l}, Time 2^{n-l}, Memory 2^l.

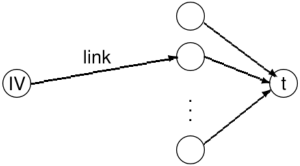
Converting Pseudo-Preimages to Preimages

Large Precomputations	Storage 2^{n-l}, Time 2^{n-l}, Memory 2^l.
Unbalanced MITM	 Time $2^{n-l/2+1}$, Memory 2^l

Converting Pseudo-Preimages to Preimages

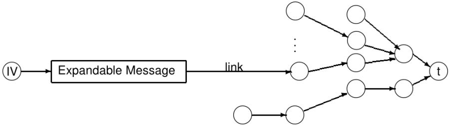
Large Precomputations — Storage 2^{n-l}, Time 2^{n-l}, Memory 2^l.

Unbalanced MITM



Time $2^{n-l/2+1}$, Memory 2^l

Tree Constructions



Time $[2^{n-l}, 2^{n-l/2+1}]$, Memory $< 2^{2l}$

Conversion: Large Computations [9]

Observation: with 2^{n-l} computations, a pseudo-preimage can be found, and the possible input chaining is limited to a set S of size 2^{n-l}.

One can find all linking messages (i.e., for all $h \in S$, find $H(IV, M) = h$) and store (M, h).

Conversion: Large Computations [9]

Observation: with 2^{n-l} computations, a pseudo-preimage can be found, and the possible input chaining is limited to a set S of size 2^{n-l}.
One can find all linking messages (i.e., for all $h \in S$, find $H(IV, M) = h$) and store (M, h).

Overall Complexities: storage 2^{n-l}, online computation 2^{n-l}, memory 2^l.

Conversion: Large Computations [9]

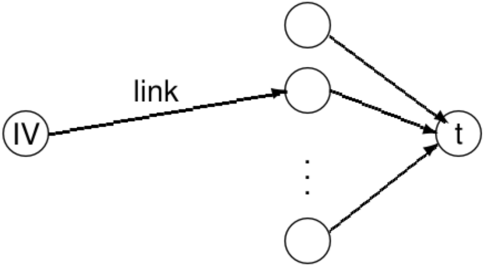
Observation: with 2^{n-l} computations, a pseudo-preimage can be found, and the possible input chaining is limited to a set S of size 2^{n-l}.

One can find all linking messages (i.e., for all $h \in S$, find $H(IV, M) = h$) and store (M, h).

Overall Complexities: storage 2^{n-l}, online computation 2^{n-l}, memory 2^{l}.

Problem: without other shortcuts, precomputation takes 2^{n}.

Conversion: Unbalanced MITM [17]



Inverting the compression function takes 2^{n-l}, and computing forward takes $2^0 = 1$, we are to meet in the middle on n bits. Best solution: repeat inversion $2^{l/2}$ times, and forward computation $2^{n-l/2}$, and overall computation results in $2^{n-l/2+1}$.

Conversion: Tree Construction [15, 9]

Observation: Multi-target pseudo-preimage attack usually works faster.

Conversion: Tree Construction [15, 9]

Observation: Multi-target pseudo-preimage attack usually works faster.

[15] 2008: MD4 pseudo-preimage attack speeds up linearly.

Conversion: Tree Construction [15, 9]

Observation: Multi-target pseudo-preimage attack usually works faster.

[15] 2008: MD4 pseudo-preimage attack speeds up linearly.

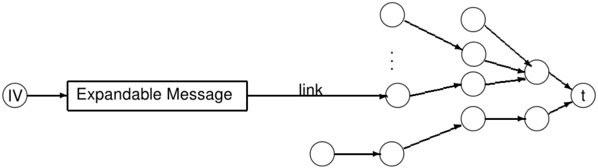
[9] 2010: Usually with 2^l targets, the attack speeds up by $2^{l/2}$.

Conversion: Tree Construction [15, 9]

Observation: Multi-target pseudo-preimage attack usually works faster.

[15] 2008: MD4 pseudo-preimage attack speeds up linearly.

[9] 2010: Usually with 2^l targets, the attack speeds up by $2^{l/2}$.



[9]: Time Complexity $3 \cdot 2^{n-2l/3}$ v.s. $2^{n-l/2}$ by unbalanced MITM approach.

Converting to Pseudo-Collision Attack [16]

When the matching point located at the end of the compression function, pseudo-collision can be found in $2^{n/2-l/2}$ v.s. $2^{n/2}$ in the ideal case.

Converting to Pseudo-Collision Attack [16]

When the matching point located at the end of the compression function, pseudo-collision can be found in $2^{n/2-l/2}$ v.s. $2^{n/2}$ in the ideal case.

The idea: preset matching l bits of the target to a constant C, with 2^l computations, one gets 2^l candidates with the target l bits set to C. Apply birthday attack to the remaining $n - l$ bits.

Converting to Pseudo-Collision Attack [16]

When the matching point located at the end of the compression function, pseudo-collision can be found in $2^{n/2-l/2}$ v.s. $2^{n/2}$ in the ideal case.

The idea: preset matching l bits of the target to a constant C, with 2^l computations, one gets 2^l candidates with the target l bits set to C. Apply birthday attack to the remaining $n - l$ bits.

Complexity: Time $2^{(n-l)/2}$, Memory 2^l.

New Applications to Block Ciphers

- KTANTAN [3, 23], 2010, 2011
- GOST [10], 2011
- 8-round AES-128 [2], 2011
- 7.5-round IDEA [13], 2012
- more ...

Details to be shown in next talk ...

How far should we go

- Time $2^{n-\epsilon}$, how big ϵ shall we consider it as "attack"? $2^n/n$?
- Should we consider the bruteforce on part of the cipher as "attack"? One can attack lots more steps by increasing the time complexity by a little bit.

Suggestion from Guo: work toward reducing the time complexity, and only on those who have the potential to be improved!

How far should we go

- Time $2^{n-\epsilon}$, how big ϵ shall we consider it as "attack"? $2^n/n$?
- Should we consider the bruteforce on part of the cipher as "attack"? One can attack lots more steps by increasing the time complexity by a little bit.

Suggestion from Guo: work toward reducing the time complexity, and only on those who have the potential to be improved!

Thank You!

References I

[1] M. Abe, editor.
Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings, volume 6477 of *LNCS*. Springer, 2010.

[2] A. Bogdanov, D. Khovratovich, and C. Rechberger.
Biclique Cryptanalysis of the Full AES.
In D. H. Lee and X. Wang, editors, *ASIACRYPT*, volume 7073 of *LNCS*, pages 344–371. Springer, 2011.

[3] A. Bogdanov and C. Rechberger.
A 3-Subset Meet-in-the-Middle Attack: Cryptanalysis of the Lightweight Block Cipher KTANTAN.
In A. Biryukov, G. Gong, and D. R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *LNCS*, pages 229–240. Springer, 2010.

[4] D. Chaum and J.-H. Evertse.
Crytanalysis of DES with a Reduced Number of Rounds: Sequences of Linear Factors in Block Ciphers.
In H. C. Williams, editor, *CRYPTO*, volume 218 of *LNCS*, pages 192–211. Springer, 1985.

[5] W. Diffie and M. E. Hellman.
Exhaustive Cryptanalysis of the NBS Data Encryption Standard.
Computer, 10:74–84, June 1977.

[6] O. Dunkelman, N. Keller, and A. Shamir.
Improved Single-Key Attacks on 8-Round AES-192 and AES-256.
In Abe [1], pages 158–176.

[7] O. Dunkelman, G. Sekar, and B. Preneel.
Improved Meet-in-the-Middle Attacks on Reduced-Round DES.
In K. Srinathan, C. P. Rangan, and M. Yung, editors, *INDOCRYPT*, volume 4859 of *LNCS*, pages 86–100. Springer, 2007.

References II

[8] R. W. Floyd.
Nondeterministic Algorithms.
Journal of the ACM, 14(14):636 – 644, October 1967.

[9] J. Guo, S. Ling, C. Rechberger, and H. Wang.
Advanced Meet-in-the-Middle Preimage Attacks: First Results on Full Tiger, and Improved Results on MD4 and SHA-2.
In Abe [1], pages 56–75.

[10] T. Isobe.
A Single-Key Attack on the Full GOST Block Cipher.
In Joux [11], pages 290–305.

[11] A. Joux, editor.
Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers, volume 6733 of *LNCS*. Springer, 2011.

[12] D. Khovratovich.
Bicliques for permutations: collision and preimage attacks in stronger settings.
Cryptology ePrint Archive, Report 2012/141, 2012.
http://eprint.iacr.org/.

[13] D. Khovratovich, G. Leurent, and C. Rechberger.
Narrow-Bicliques: Cryptanalysis of Full IDEA.
In D. Pointcheval and T. Johansson, editors, *EUROCRYPT*, volume 7237 of *LNCS*, pages 392–410. Springer, 2012.

[14] D. Khovratovich, C. Rechberger, and A. Savelieva.
Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 family.
IACR Cryptology ePrint Archive, 2011:286, 2011.

References III

[15] G. Leurent.
MD4 is Not One-Way.
In K. Nyberg, editor, *FSE 2008*, volume 5086 of *LNCS*, pages 412–428. Springer, 2008.

[16] J. Li, T. Isobe, and K. Shibutani.
Converting meet-in-the-middle preimage attack into pseudo collision attack: Application to sha-2.
In A. Canteaut, editor, *FSE*, volume 7549 of *LNCS*. Springer, 2012.

[17] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone.
Handbook of Applied Cryptography.
CRC Press, 1996.

[18] R. C. Merkle and M. E. Hellman.
On the Security of Multiple Encryption.
Commun. ACM, 24(7):465–467, 1981.

[19] H. Morita, K. Ohta, and S. Miyaguchi.
A switching closure test to analyze cryptosystems.
In J. Feigenbaum, editor, *CRYPTO*, volume 576 of *LNCS*, pages 183–193. Springer, 1991.

[20] Y. Sasaki.
Meet-in-the-Middle Preimage Attacks on AES Hashing Modes and an Application to Whirlpool.
In Joux [11], pages 378–396.

[21] R. Sedgewick, T. G. Szymanski, and A. C. Yao.
The Complexity of Finding Cycles in Periodic Functions.
SIAM Journal on Computing, 11:376–390, 1982.

[22] P. C. van Oorschot and M. J. Wiener.
Parallel Collision Search with Cryptanalytic Applications.
Journal of Cryptology, 12(1):1–28, 1999.

References IV

[23] L. Wei, C. Rechberger, J. Guo, H. Wu, H. Wang, and S. Ling.
Improved Meet-in-the-Middle Cryptanalysis of KTANTAN (Poster).
In U. Parampalli and P. Hawkes, editors, *ACISP*, volume 6812 of *LNCS*, pages 433–438. Springer, 2011.