Hack Lab Presentation

Surveying And Reproducing MD5 Fast Collision Attack Algorithms

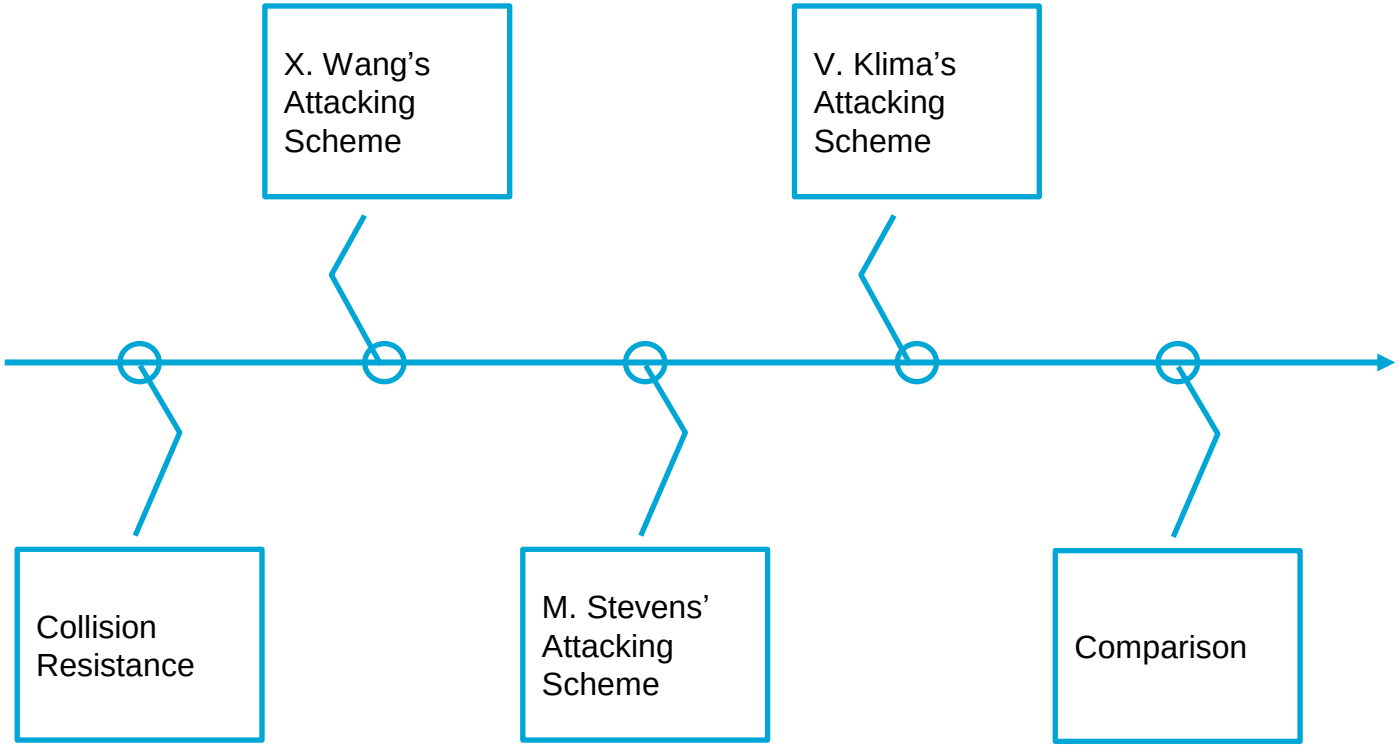
Supervisor: Kaitai Liang

Meng Zheng
Electrical Engineering
M.Zheng-3@student.tudelft.nl

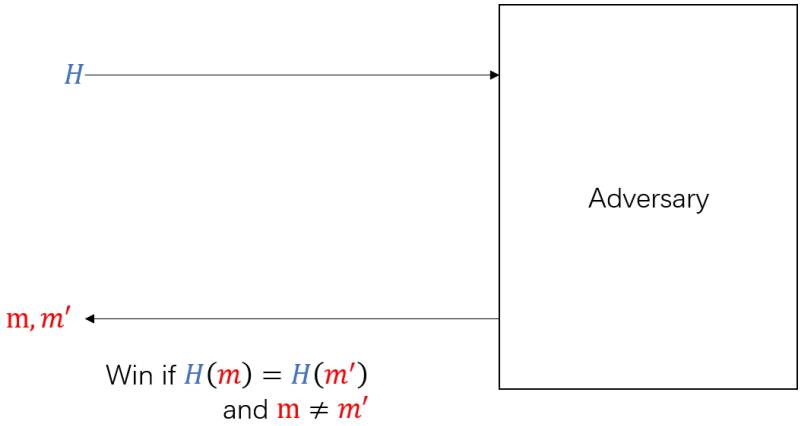
Yuhang Tian
Embedded System
Y.Tian-13@student.tudelft.nl

April 2021

Overview


X. Wang's Attacking Scheme
V. Klima's Attacking Scheme
Collision Resistance
M. Stevens' Attacking Scheme
Comparison

Collision Resistance & Collision Attack



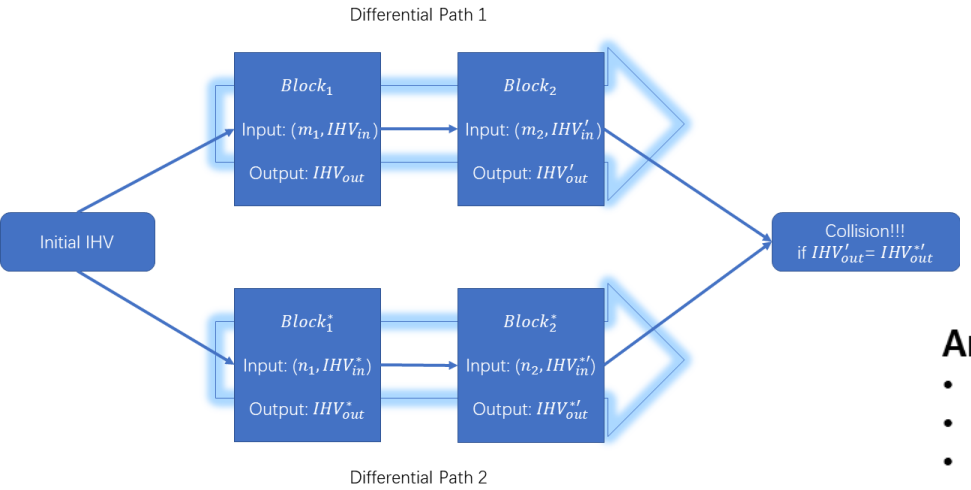
A hash function should have the collision resistance – it is **infeasible** to find two messages that have the identical hash values

General attacks – Birthday Paradox
For a given hash function with output size of N bits, this general algorithm succeeds after approximately

$$\sqrt{\frac{\pi}{2}} \times 2^{\frac{N}{2}}$$

For MD5, $\approx 2^{64.3}$

Wang's Collision Attack –
Differential Path & Message Modification



Message Modification:
When a certain condition in the second round fails, one can use message modification. This is a substitution formula specially made for this condition on the message block B. In the case that this condition does not hold applying this substitution has the effect that this condition now does hold without interfering with other previous conditions.

Any Improvement?
- C → Java
- rearrange the structure of the second block
- the original code calculates x[i]s during the testing, we put them afterwards so as to avoid redundant calculations
- prune our code according to the feedback by SonarQube

In general, it can improve about 15% of the searching speed and significantly promote efficiency

Marc Stevens' Fastcoll Attack

MD5 backward & optimized sufficient requirements

- Collision happens deterministically
- Random Q fullfill conditions
- Calculate Message using MD5 backward:

$$m_t = RR(Q_{t+1} - Q_t, RC_t) - f_t(Q_t, Q_{t-1}, Q_{t-2}) - Q_{t-3} - AC_t \text{ for } 0 \leq t \leq 15.$$

- Judging other optimized conditions
- Without Tunnels

Any Improvement?

- C ⟶ Python
- Enable overflow
- Enable same Bin & Hex structure as Java
- Still not efficient enough as without Tunnels & addition operating

Klima's Collision Attack –

Point of Verification (PoV) & Tunnels

- The Deterministic and Probabilistic Tunnels
- The modification is more flexible
- Tunneling enables to fast collision searching and in some sense replace present multi-message modification methods considerably.

Any Improvement?

- C → Java
- Pre-calculate masking bits to avoid using *"mask_bit"* function to generate them during looping

However, the improvement is not significant, since the speed is originally fast.

Comparison

Table 4.1: Comparison of MD5 Collision Schemes

	Wang's Collision	Stevens' Collision (IPA)	Klima's Collision	Stevens' Collision (CPA)
Core Idea	differential path message modification early stop	differential path message modification early stop	differential path point of verification	differential path message modification early stop near collision
Execution Time	$\approx$ 30 min	$\approx$ 30 min	$\approx$ 1 min	$>$ 1 hour
Feature	insufficient conditions large executing variance	sufficient conditions samll executing variance	tunnel implementation fastest execution	prefixes can be different

- Due to the insufficient conditions, Wang's searching sometimes meaninglessly traverses Q[20], which adversely increases the worst-case execution time.
- The searching time of Stevens' attack tends to be more stable.
- The fast algorithm to find a collision pair of MD5 is the one provided by Klima equipped with tunnels.
- Stevens' versions are fancy that can choose the prefix or even use a single block to obtain the collision, but the trade-off is spending more time.

Comparison

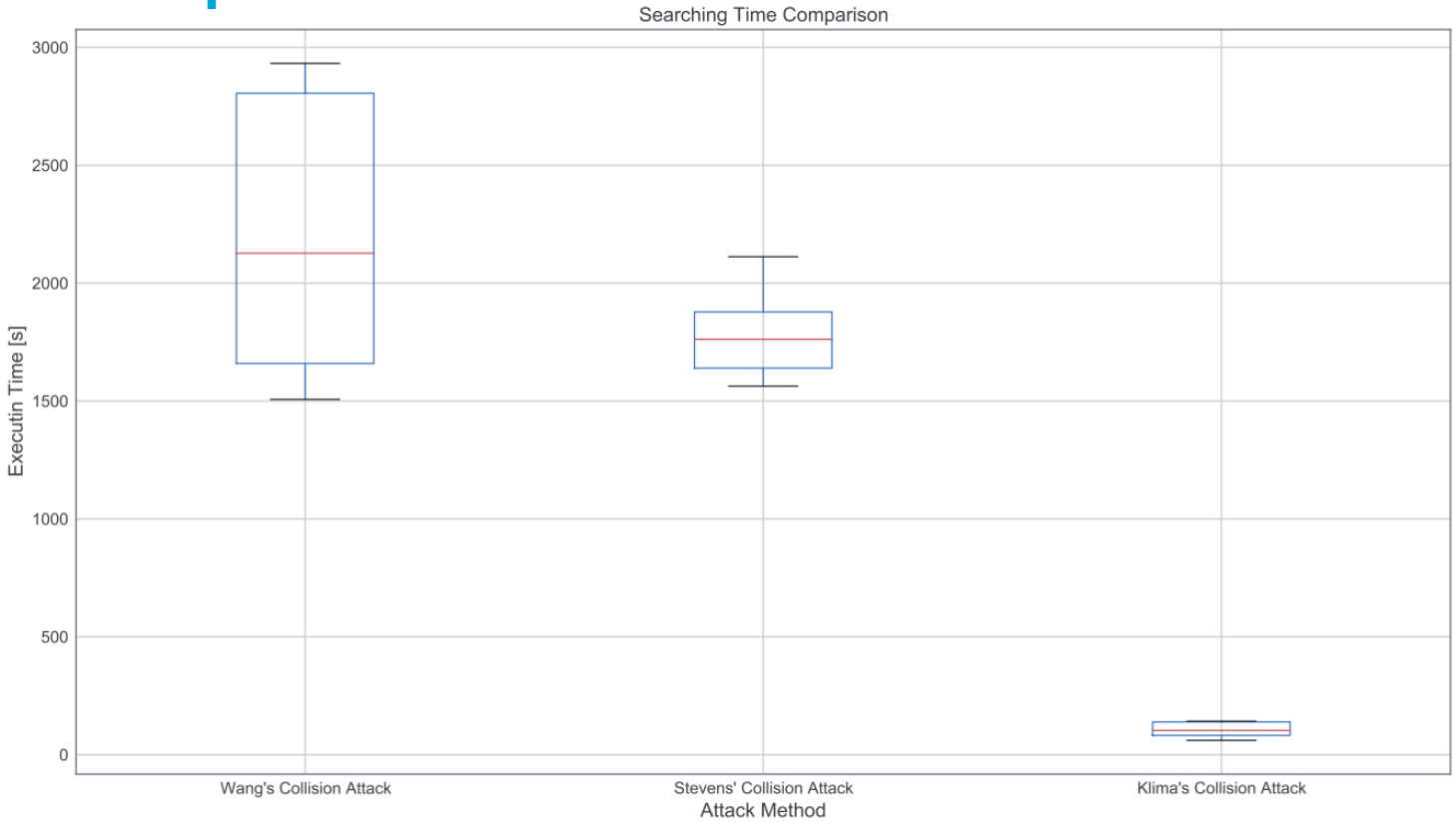


Figure 4.1: Box-plot Comparison

Thanks for your listening!

- Questions?

Project Website:
https://github.com/Timo9Madrid7/MD5-Collision

Reference

- Stevens, M., Lenstra, A. K., & De Weger, B. (2012). Chosen-prefix collisions for MD5 and applications. International Journal of Applied Cryptography, 2(4), 322-359.
- Stevens, M. (2012). Attacks on hash functions and applications. Mathematical Institute, Faculty of Science, Leiden University, 3.
- Stevens, M. (2006). Fast Collision Attack on MD5. IACR Cryptol. ePrint Arch., 2006, 104.
- Wang, X., & Yu, H. (2005, May). How to break MD5 and other hash functions. In Annual international conference on the theory and applications of cryptographic techniques (pp. 19-35). Springer, Berlin, Heidelberg.
- Klima, V. (2006). Tunnels in Hash Functions: MD5 Collisions Within a Minute. IACR Cryptol. ePrint Arch., 2006, 105.
- Klima, V. (2005). Finding MD5 Collisions on a Notebook PC Using Multi-message Modifications. IACR Cryptol. ePrint Arch., 2005, 102.
- Hawkes, P., Paddon, M., & Rose, G. G. (2004). Musings on the Wang et al. MD5 Collision. IACR Cryptol. ePrint Arch., 2004, 264.
- Smart, N. P., & Smart, N. P. (2016). Cryptography made simple. Springer.
- Mikle, O. (2004). Practical Attacks on Digital Signatures Using MD5 Message Digest. IACR Cryptol. ePrint Arch., 2004, 356.
- Kaminsky, D. (2005). MD5 to be considered harmful someday. In Aggressive Network Self-Defense (pp. 323-337). Syngress.
- Kashyap, N. D. (2006). A meaningful MD5 hash collision attack.