Improved Single-Key Distinguisher on HMAC-MD5 and Key Recovery Attacks on Sandwich-MAC-MD5

Yu Sasaki[1] and Lei Wang[2]

[1]NTT Secure Platform Laboratories

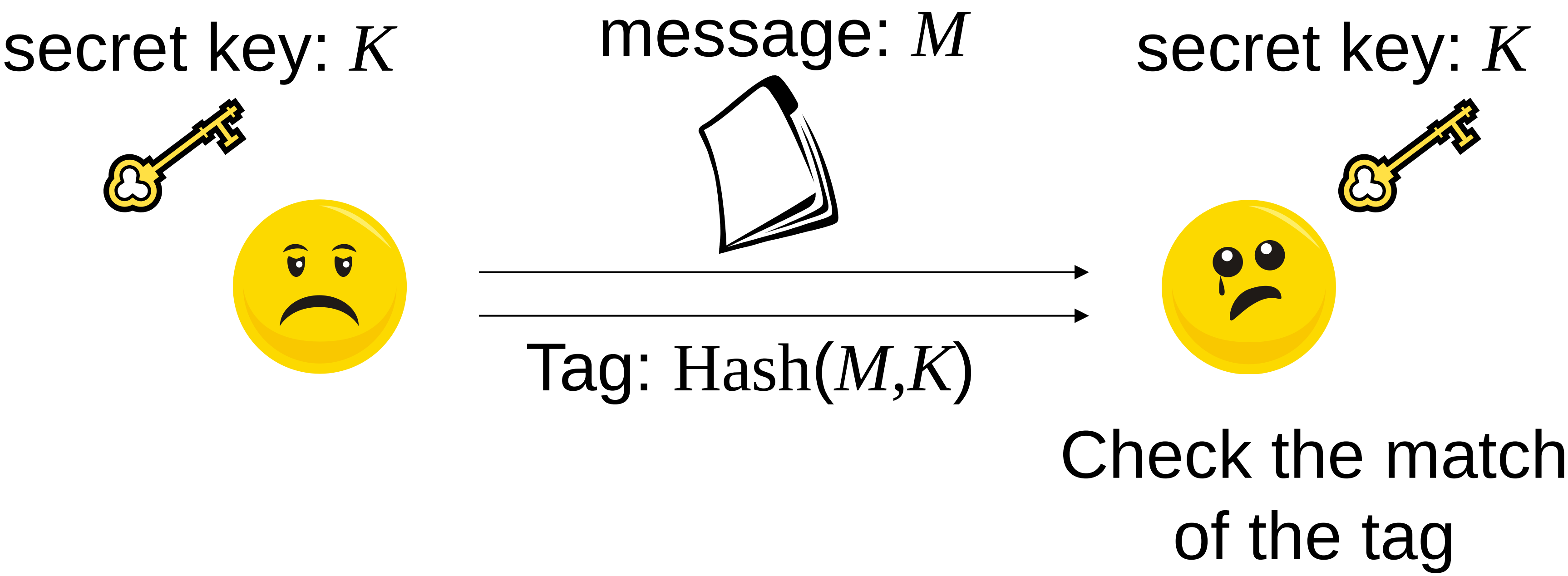
[2]Nanyang Technological University, Singapore

SAC 2013 (16/August/2013)

NTT

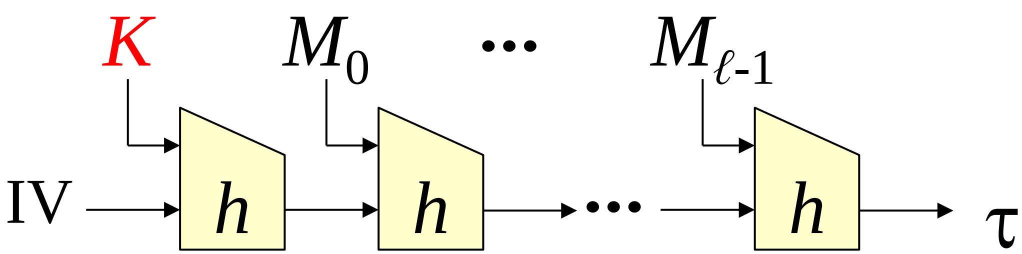
Hash Function Based MAC

- Message Authentication Codes (MAC) provide the integrity and authenticity.



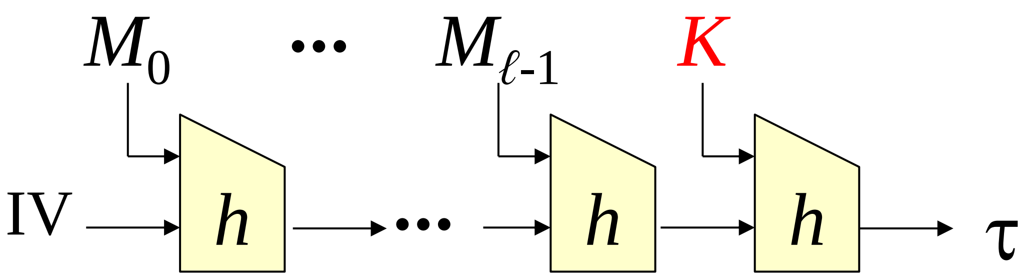
Classical MAC Constructions

- Prefix



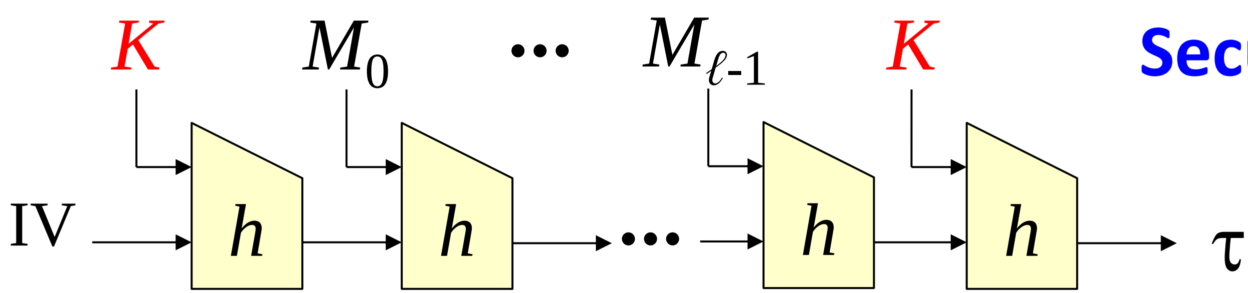
Length extension attack

- Suffix



Collision attack

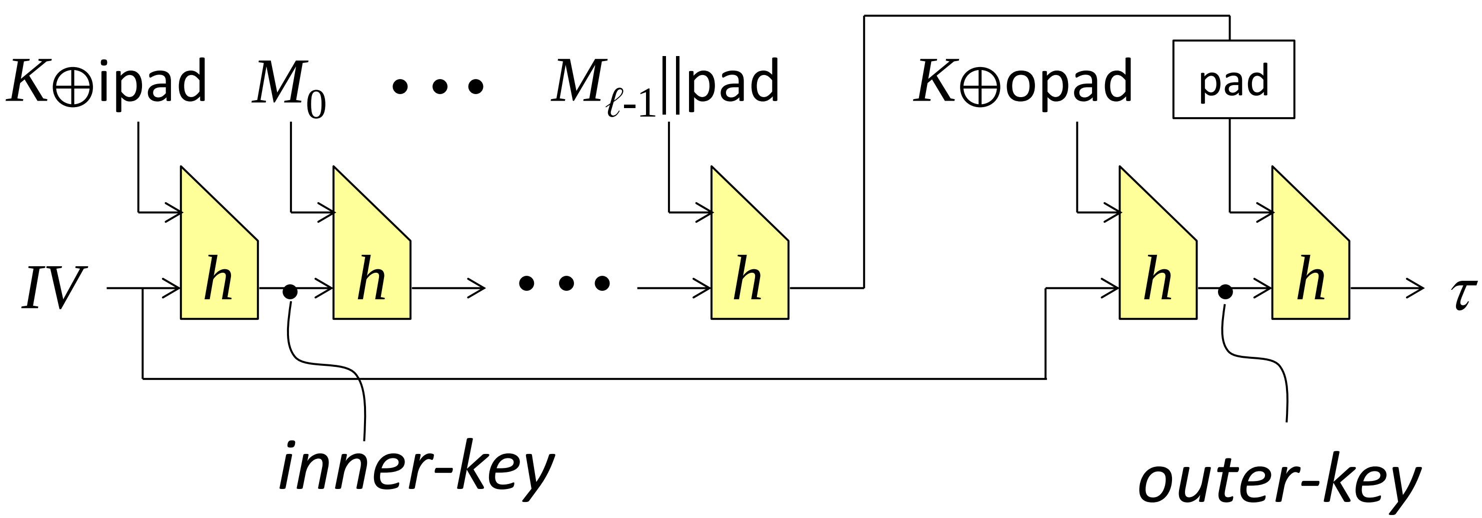
- Hybrid



Secure !!

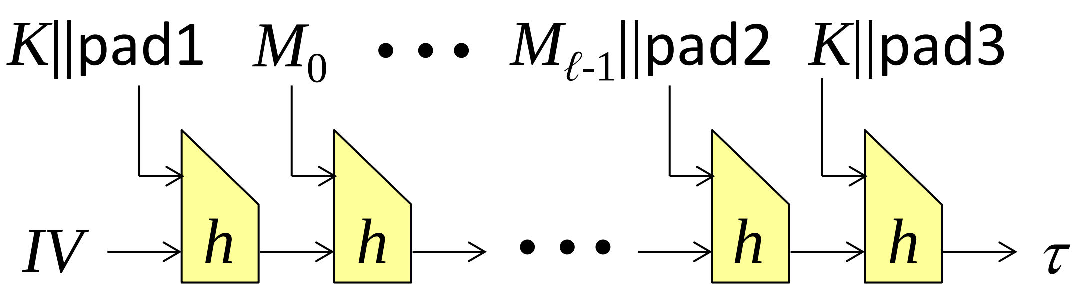
HMAC

- The most widely used hash-based MAC
  - Requires 2 keys for inner and outer functions
  - Requires 2 hash function calls
  - 3 additional blocks for converting hash into MAC; non-negligible overhead for short messages



Sandwich-MAC

- Several MACs improve HMAC
- Sandwich-MAC [Yasuda ACISP 2007] has advantages on performance.
  - Requires 1 key
  - Requires 1 hash function call
  - 2 additional blocks for converting hash into MAC ; small overhead, suitable for short messages



Motivation

- HMAC and Sandwich-MAC have the same provable security: secure PRF up to $O(2^{n/2})$.
- Need more comparison

- We investigate attacks when a weak hash function (MD5) is instantiated.
- Then, extract features which can be applied in generic.

Our Contributions

1. Improve the internal state recovery attack on HMAC-MD5 both in adaptive and non-adaptive settings.

2. By using the above, propose a key-recovery attack on Sandwich-MAC-MD5.
   – First key recovery attack on hybrid-type MACs
   – conditional key distribution technique

3. Improve the attack on MD5-MAC$_{K0,K1,K2}$.
   – Improve the complexity to recover K_1.
   – Propose the first key recovery attack for K_2.

Attack Results

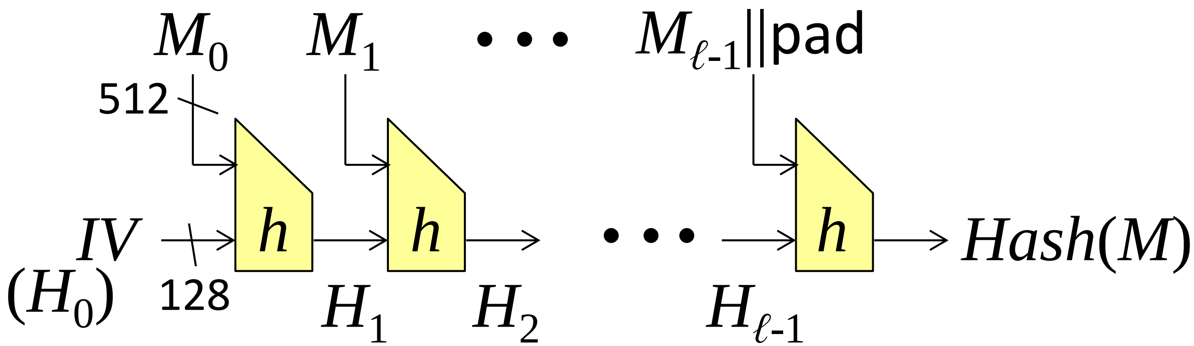
Target	Model	Attack goal	Data	Time	Memory	Ref.
HMAC-MD5	Adaptive	Dist-H/ISR	2^{97}	2^{97}	2^{89}	[32]
	Adaptive	Dist-H/ISR	$2^{89.09}$	2^{89}	2^{89}	Ours
	Non-adaptive	Dist-H/ISR	2^{113}	2^{113}	2^{66}	[32]
	Non-adaptive	Dist-H/ISR	2^{113-x}	2^{113-x}	2^{66+x}	Ours
MD5-MAC		K_1-recovery	2^{97}	2^{97}	2^{89}	[32]
		K_1-recovery	$2^{89.09}$	2^{89}	2^{89}	Ours
		(K_1, K_2)-recovery	$2^{89.04}$	2^{89}	2^{89}	Ours
Sandwich-MAC-MD5	Basic	Key recovery	$2^{89.04}$	2^{89}	2^{89}	Ours
	Variant B	Key recovery	$2^{89.04}$	2^{89}	2^{89}	Ours
	Extended B	Key recovery	$2^{89.04}$	2^{89}	2^{89}	Ours

Improved Single-key Attacks against HMAC-MD5

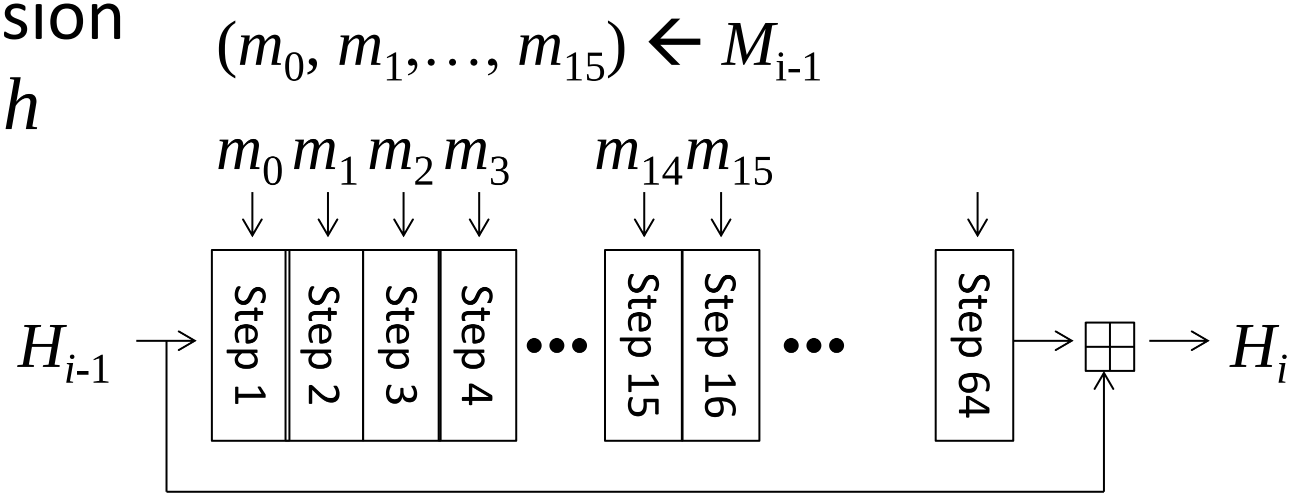
MD5

- Widely known to be broken but still widely used

Merkle-Damgård structure

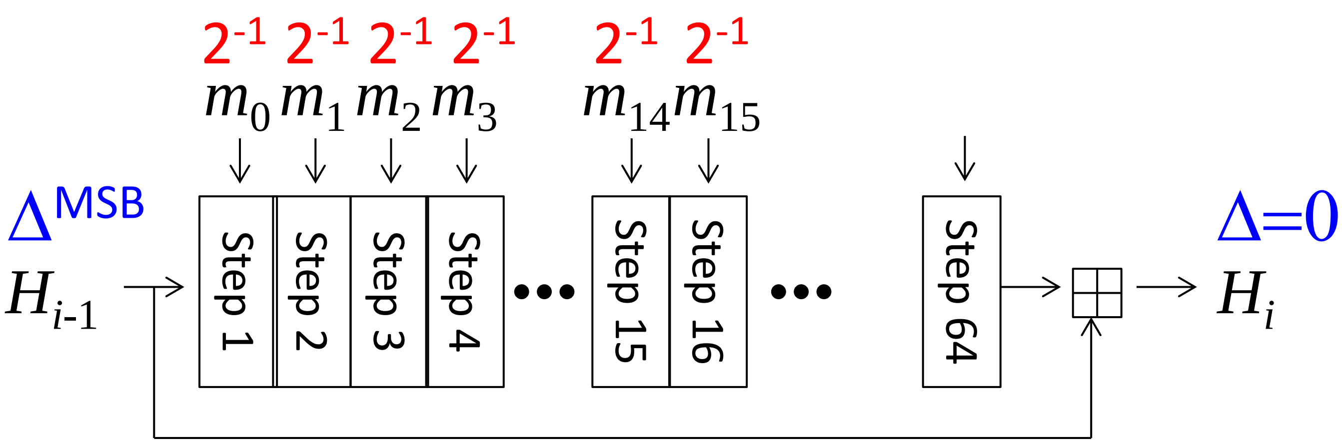


Compression function h



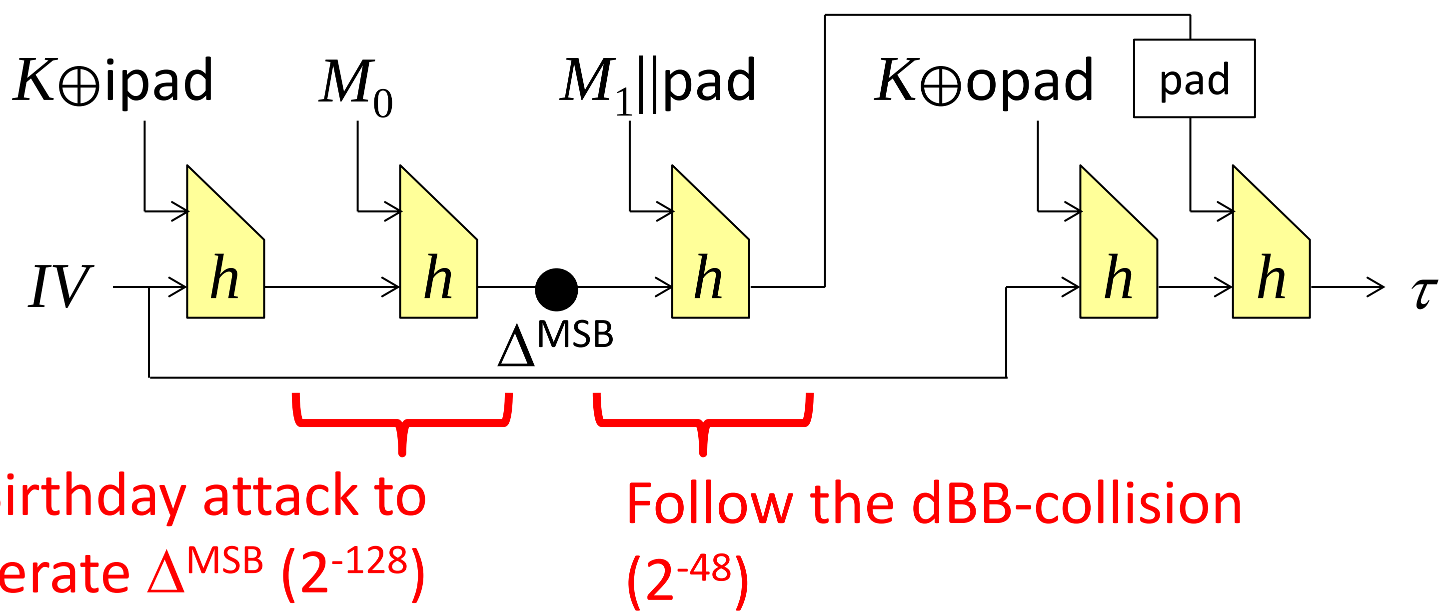
dBB-collision

- The compression function h generates a collision with probability 2^{-48} for (H_{i-1}, M_{i-1}) and (H_{i-1}', M_{i-1}) when $H_{i-1} \oplus H_{i-1}'$ has a special difference called Δ^{MSB}.
- In the dBB-collision, each of the first 16 steps has the differential characteristic with $Pr.=2^{-1}$.



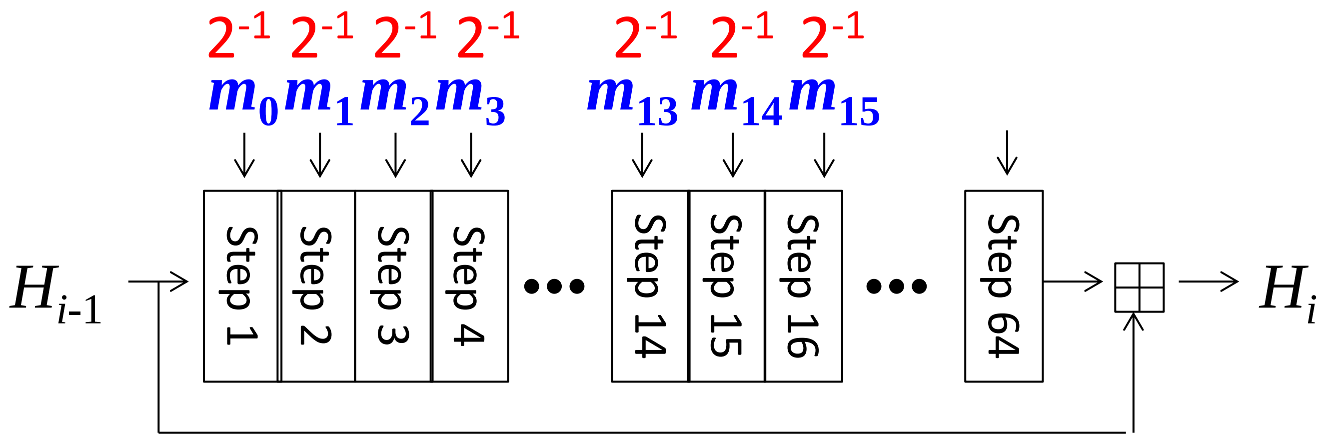
Previous Attack against HMAC-MD5

1. Generate $2^{128} \times 2^{48} = 2^{176}$ pairs by changing M_0.
   - One pair satisfies the dBB-collision.
   - We have other $2^{176-128} = 2^{48}$ collisions. (noise)
2. For each 2^{48} collisions, change M_1 2^{48} times.
   - If another collision is found, it is a dBB-collision.

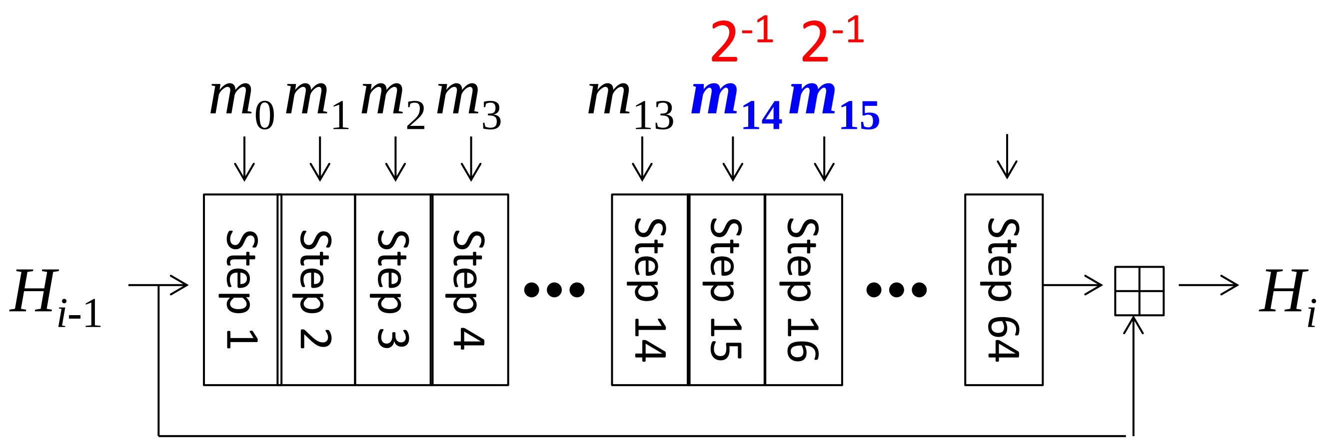


Improving ISR against HMAC-MD5

Previous work: retake all messages → $Pr = 2^{-48}$.



Ours: Reuse the messages for the first 14 steps so that the characteristic remains satisfied. → $Pr = 2^{-34}$.

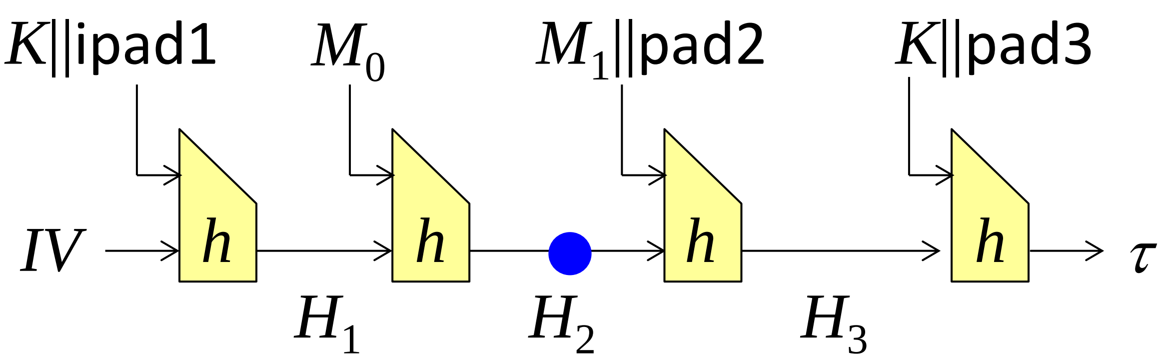


NTT

Key Recovery Attacks against Sandwich-MAC-MD5

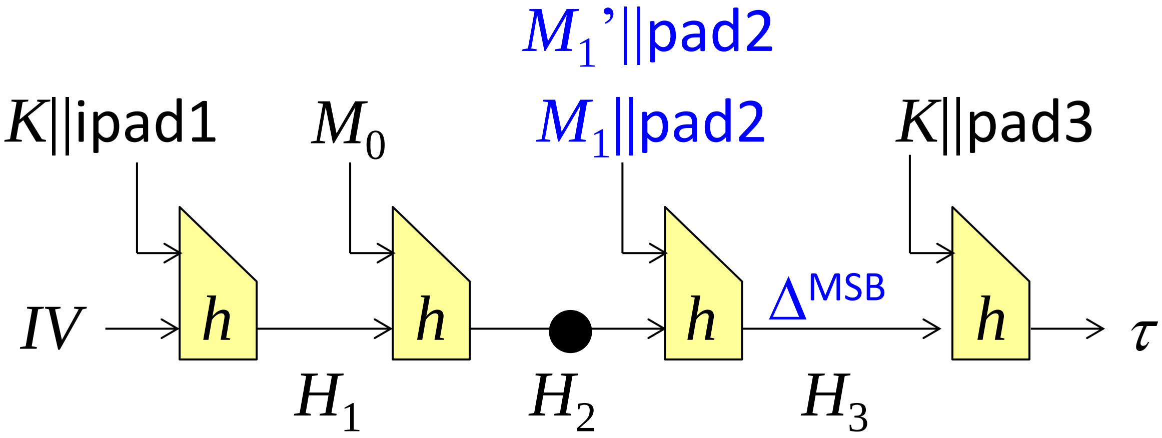
Phase 1: Internal State Recovery

- Recover the internal state value H_2, similarly with the internal state recovery on HMAC-MD5.



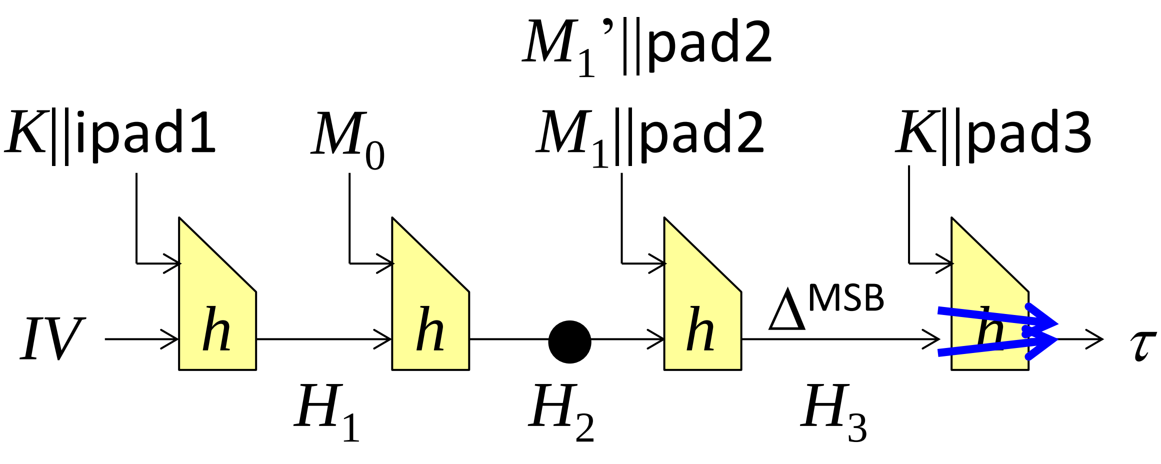
Phase 2: IV Bridge

- From the recovered H_2, find (M_1, M_1') which generates Δ^{MSB} at H_3.
- This can be done by a variant of collision attack called IV Bridge with a complexity of 2^{10} [Tao+ ePrint].



Phase 3: Collecting dBB-near-collisions

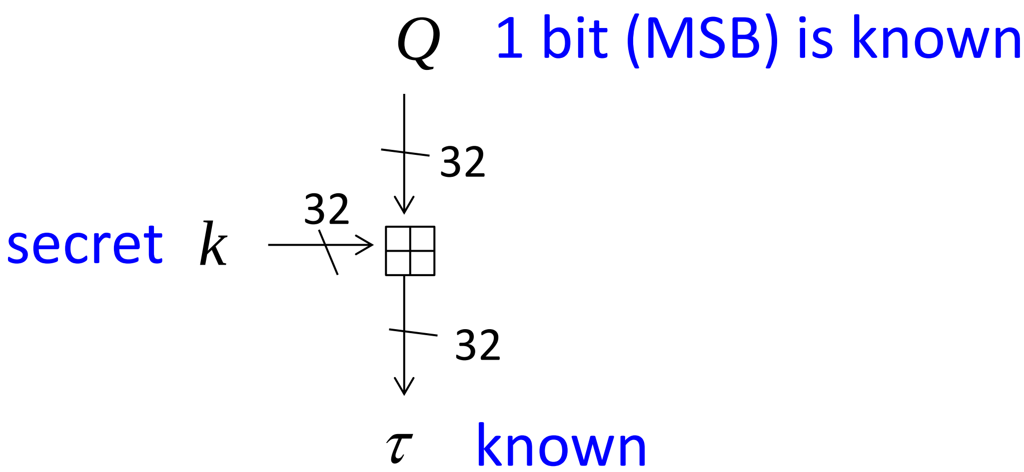
- By querying 2^{48} IV bridges, one tag collision is obtained. To be precise, 2^{47} IV bridges to obtain dBB-near-collisions enough.
- For the dBB-near-collision, 1 bit of internal state is recovered because the characteristic is satisfied.



Key Recovery with Conditional Key Distributions

- Due to the structure of the MD5 compression function, 32 bits of the tag τ are computed by

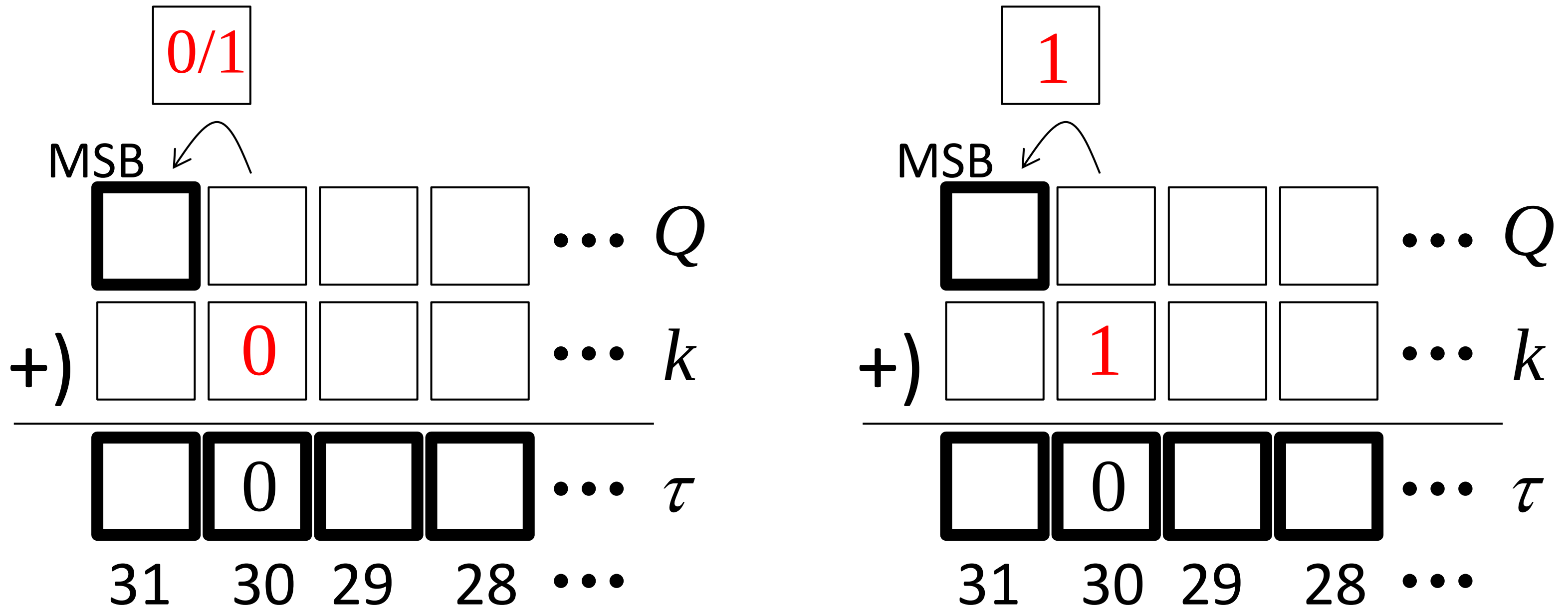
  (internal state Q) $\boxplus$ (a part of secret key k)



- By collecting 2^{32} pairs of such (Q, τ), the secret key k can be recovered.

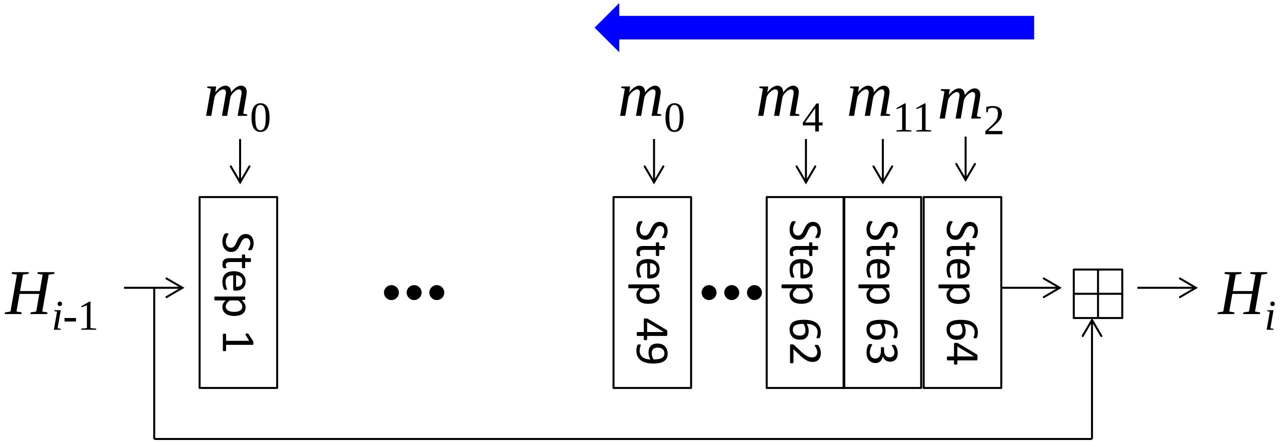
Conditional Key Distributions: Overview

- Collect pairs in which the 30th bit of τ is 0.
  1. If the 30th bit of k is 0: two possible carry patterns
  2. If the 30th bit of k is 1: one possible carry pattern
- Behavior of the addition depends on the key value. This eventually reveals the 30th and 31st bits of k.



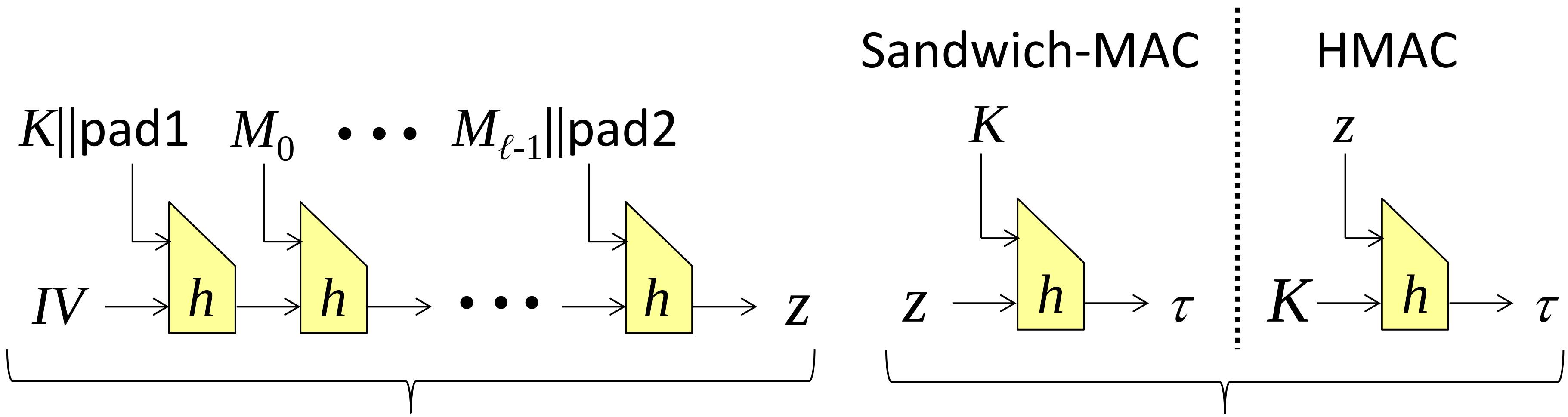
Phase 4: Rest of Attacks

- The key for the last step is recovered by using the conditional key distribution.
- Then, all keys are recovered step by step for the last 16 steps.



Discussion:
HMAC v.s. Sandwich-MAC

Comparison of HMAC and Sandwich-MAC

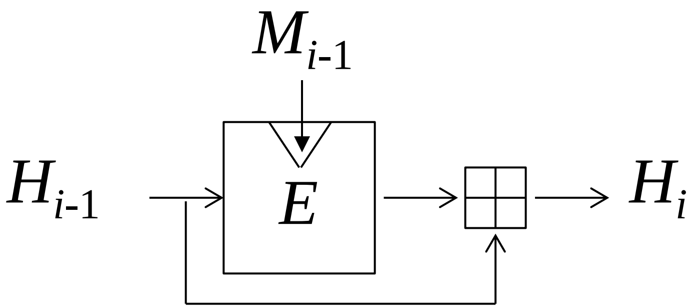


Message processing part is identical. Finalization is different.

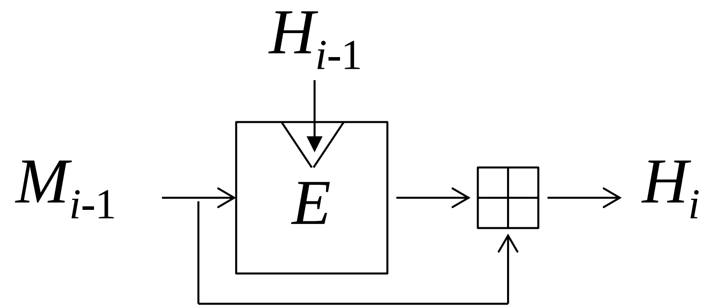
- Sandwich-MAC: A differential characteristic to recover the internal state is reused to recover K.
- HMAC: Two good characteristics are needed to recover K.

Comparison for Block-cipher Based Hash

Davies-Meyer mode



MMO mode



- In hybrid MACs, the MMO mode is the only choice for the finalization computation to resist side-channel analysis [Okeya ACISP 2006].
- Most of the currently used hash function adopts the Davies-Meyer mode.
- The HMAC construction is the most reasonable!!

Concluding Remarks

Attacks with MD5

- Improved internal state recovery attack on HMAC-MD5 in adaptive and non-adaptive settings.
- Key-recovery attack on Sandwich-MAC-MD5 with conditional key distribution techniques.
- Improve the attack on MD5-MAC.

Comparison with HMAC and Sandwich-MAC

- A certain type of differential characteristic can recover the key for Sandwich-MAC.
- From various viewpoints, HMAC is a solid design.

NTT

Thank you for your attention!!