Key Recovery Attack on HMAC/NMAC based on Reduced Whirlpool

Jian Guo

joint on-going work with *Lei Wang* and *Shuang Wu*

Tsinghua University, China, 28 Nov 2012

Outline

1. Introduction
   - HMAC and NMAC
   - The Whirlpool Hash Function
2. Key Recovery Attacks
   - Attack Overview
   - Key Recovery for 5-round HMAC-Whirlpool
   - Key recovery for 5-round NMAC-Whirlpool
   - Key recovery for 6-round HMAC-Whirlpool in related-key setting
   - Other PGV modes and MAC modes
3. Conclusion

Outline

1 Introduction
- HMAC and NMAC
- The Whirlpool Hash Function

2 Key Recovery Attacks
- Attack Overview
- Key Recovery for 5-round HMAC-Whirlpool
- Key recovery for 5-round NMAC-Whirlpool
- Key recovery for 6-round HMAC-Whirlpool in related-key setting
- Other PGV modes and MAC modes

3 Conclusion

HMAC and NMAC

- Designed by Mihir Bellare, Ran Canetti and Hugo Krawczyk in Crypto 1996
- Standarized by ANSI, IETF, ISO, NIST from 1997
- The most widely deployed hash-based MAC construction.

HMAC and NMAC

- Designed by Mihir Bellare, Ran Canetti and Hugo Krawczyk in Crypto 1996
- Standarized by ANSI, IETF, ISO, NIST from 1997
- The most widely deployed hash-based MAC construction.

The key is inserted in the way of key-as-IV, *i.e.*, $H(K, M)$ means hashing message M with IV replaced by K.

HMAC and NMAC

- Designed by Mihir Bellare, Ran Canetti and Hugo Krawczyk in Crypto 1996
- Standarized by ANSI, IETF, ISO, NIST from 1997
- The most widely deployed hash-based MAC construction.

The key is inserted in the way of key-as-IV, *i.e.*, $H(K, M)$ means hashing message M with IV replaced by K.

$$\mathtt{NMAC}(K_0, K_1, M) = H(K_1, H(K_0, M))$$

HMAC and NMAC

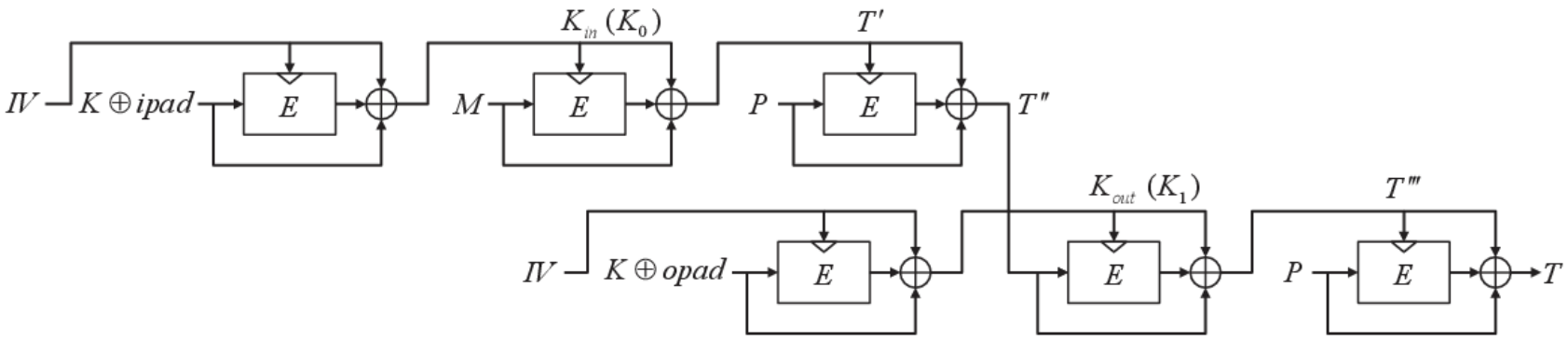
- Designed by Mihir Bellare, Ran Canetti and Hugo Krawczyk in Crypto 1996
- Standarized by ANSI, IETF, ISO, NIST from 1997
- The most widely deployed hash-based MAC construction.

The key is inserted in the way of key-as-IV, *i.e.*, $H(K, M)$ means hashing message M with IV replaced by K.

$\mathtt{NMAC}(K_0, K_1, M) = H(K_1, H(K_0, M))$

$\mathtt{HMAC}(K, M) = H(IV, K \oplus ipad \parallel H(IV, K \oplus opad \parallel M))$, ipad and opad are predefined constants.

HMAC and NMAC



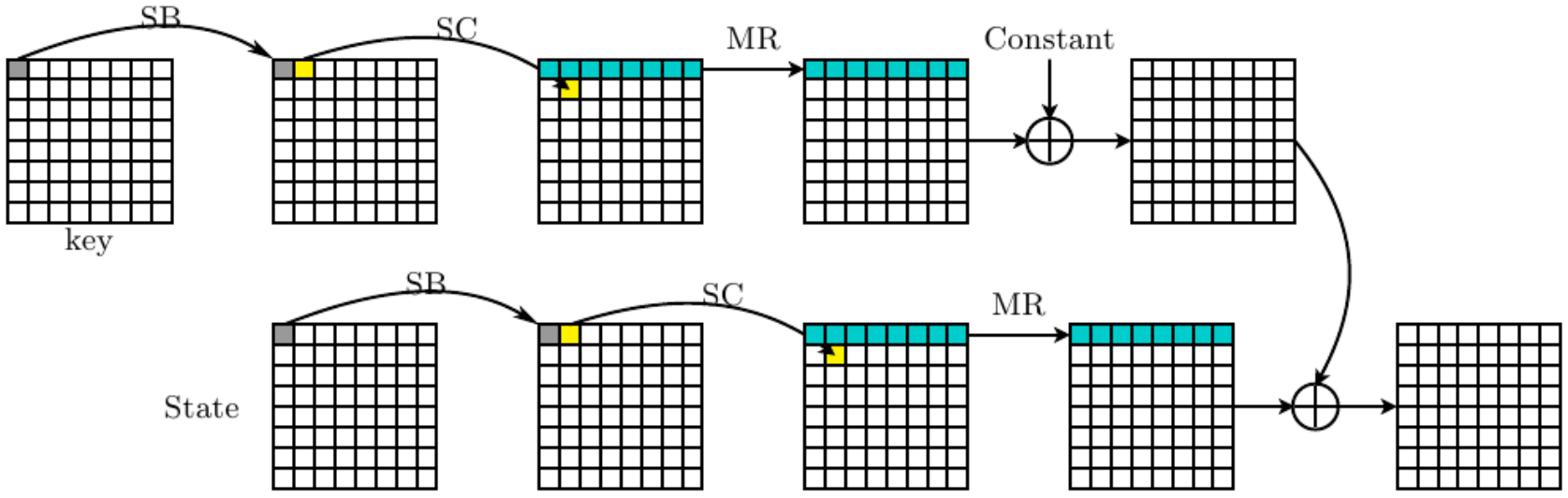
$\texttt{NMAC}(K_0, K_1, M) = H(K_1, H(K_0, M))$, with K_0 as K_{in}, and K_1 as K_{out}.

Whirlpool

- designed by Barreto and Rijmen in 2000 with 512-bit digest
- standarized by ISO/IEC
- follows Merkle-Damgård strengthening, and Miyaguchi-Preneel mode, *i.e.*, $C(H, M) = E_H(M) \oplus H \oplus M$
- both state and key follow the AES-like process

Whirlpool

- designed by Barreto and Rijmen in 2000 with 512-bit digest
- standarized by ISO/IEC
- follows Merkle-Damgård strengthening, and Miyaguchi-Preneel mode, *i.e.*, $C(H, M) = E_H(M) \oplus H \oplus M$
- both state and key follow the AES-like process



Key $AC \circ MR \circ SC \circ SB$

State $AK \circ MR \circ SC \circ SB$

Outline

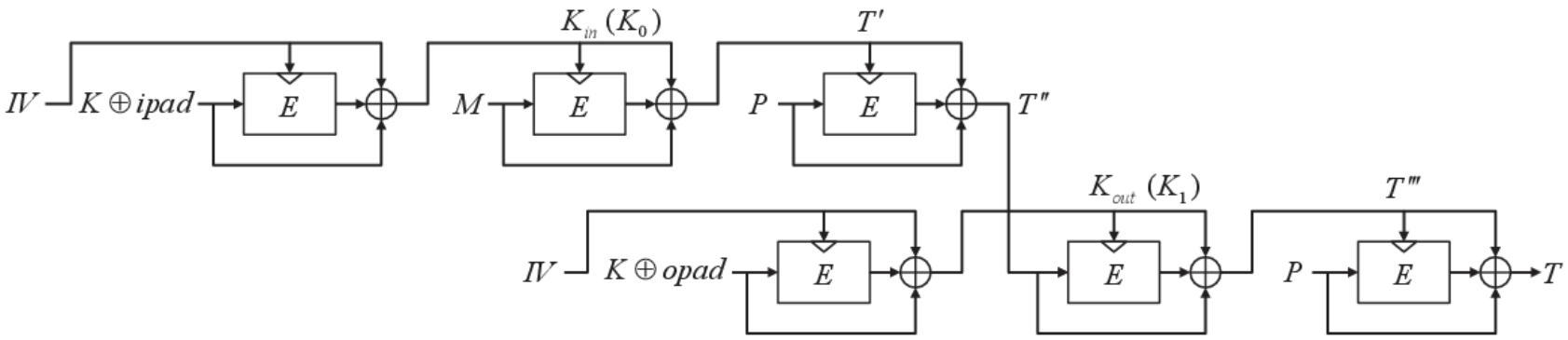
1 Introduction
- HMAC and NMAC
- The Whirlpool Hash Function

2 Key Recovery Attacks
- Attack Overview
- Key Recovery for 5-round HMAC-Whirlpool
- Key recovery for 5-round NMAC-Whirlpool
- Key recovery for 6-round HMAC-Whirlpool in related-key setting
- Other PGV modes and MAC modes

3 Conclusion

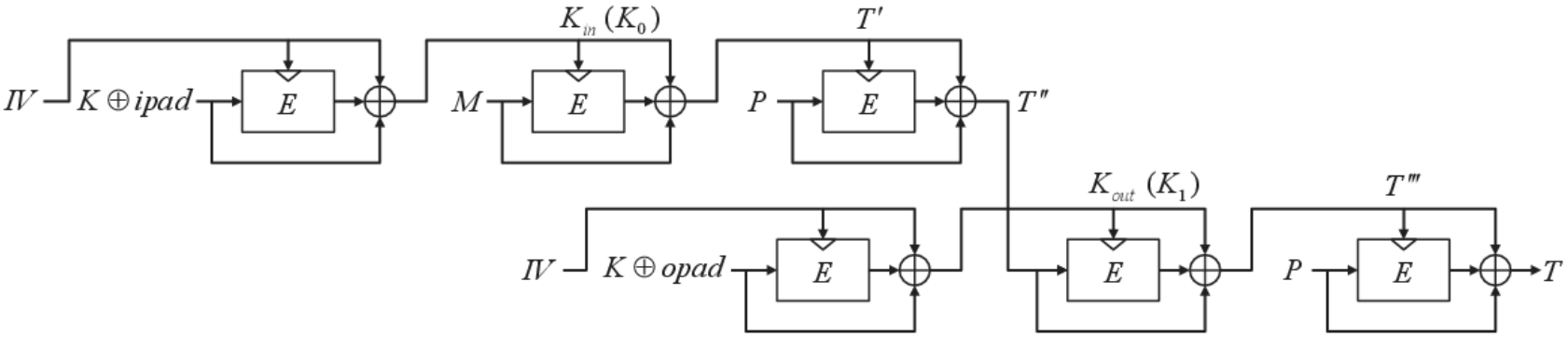
Attack Overview

1. Generate structured internal collision on T'
2. Recovering K_{in}
3. Recovering the real key from K_{in}.



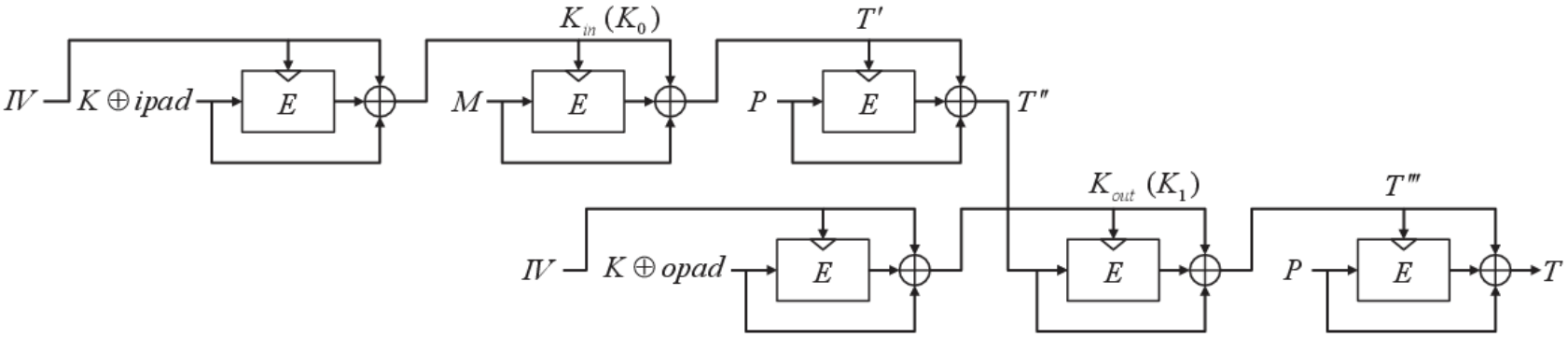
Generate structured internal collision

Finding structured internal collision on T'



Generate structured internal collision

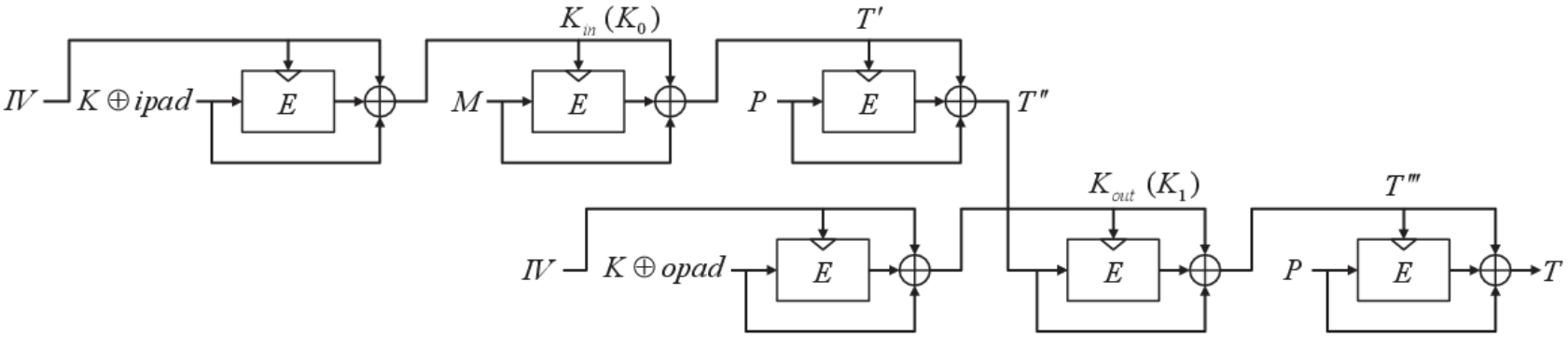
Finding structured internal collision on T'

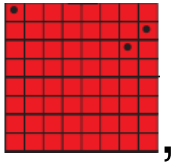


- query many M to get corresponding T, find collision of T, verify (by querying $M||M'$ for abitrary M') if it is also collision of T'.

Generate structured internal collision

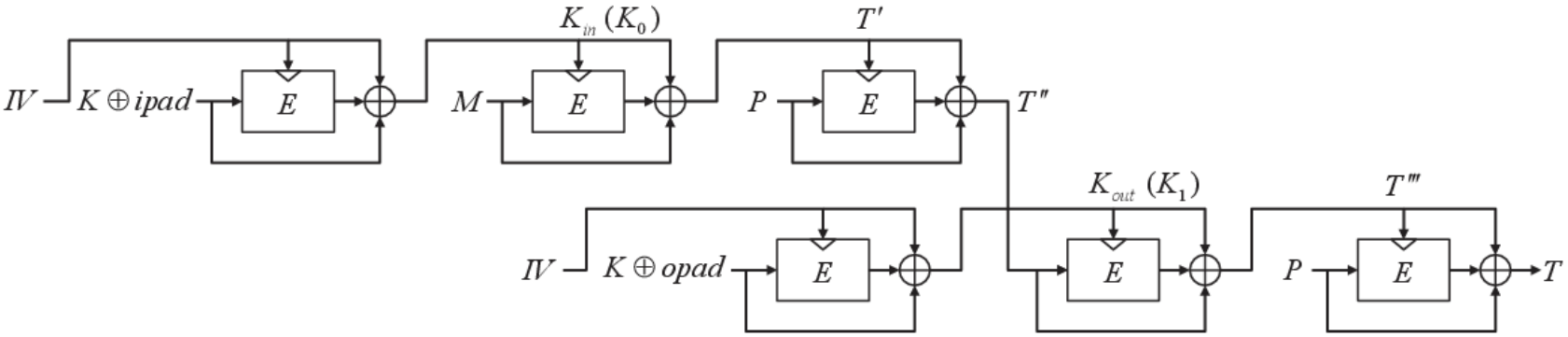
Finding structured internal collision on T'

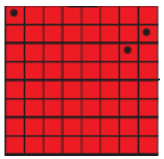


- query many M to get corresponding T, find collision of T, verify (by querying $M||M'$ for abitrary M') if it is also collision of T'.
- allow difference in 3-byte of M only, like  , finding collision of T' costs 2^{488} with 2^{24} memory.

Generate structured internal collision

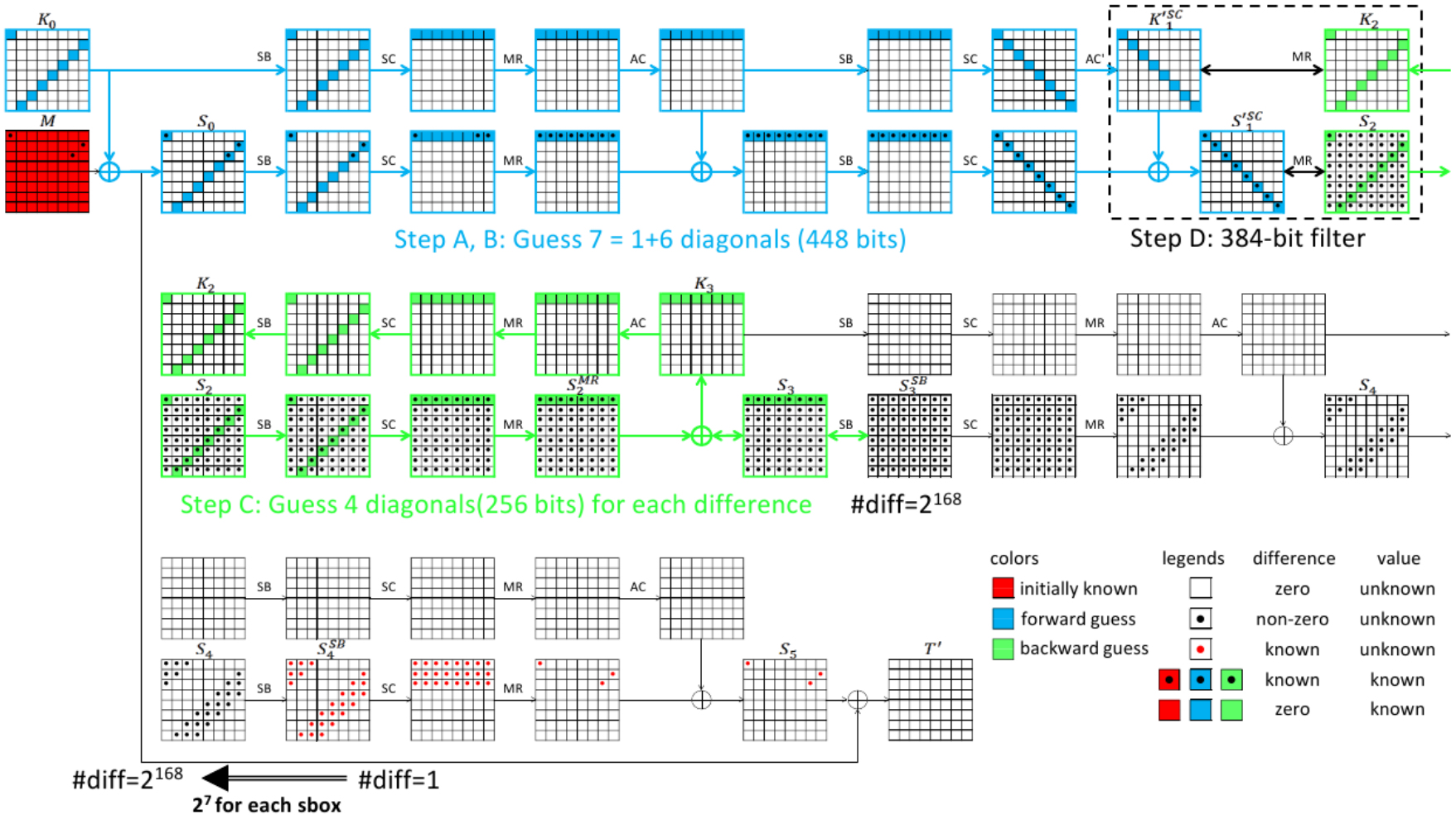
Finding structured internal collision on T'



- query many M to get corresponding T, find collision of T, verify (by querying $M||M'$ for abitrary M') if it is also collision of T'.
- allow difference in 3-byte of M only, like  , finding collision of T' costs 2^{488} with 2^{24} memory.

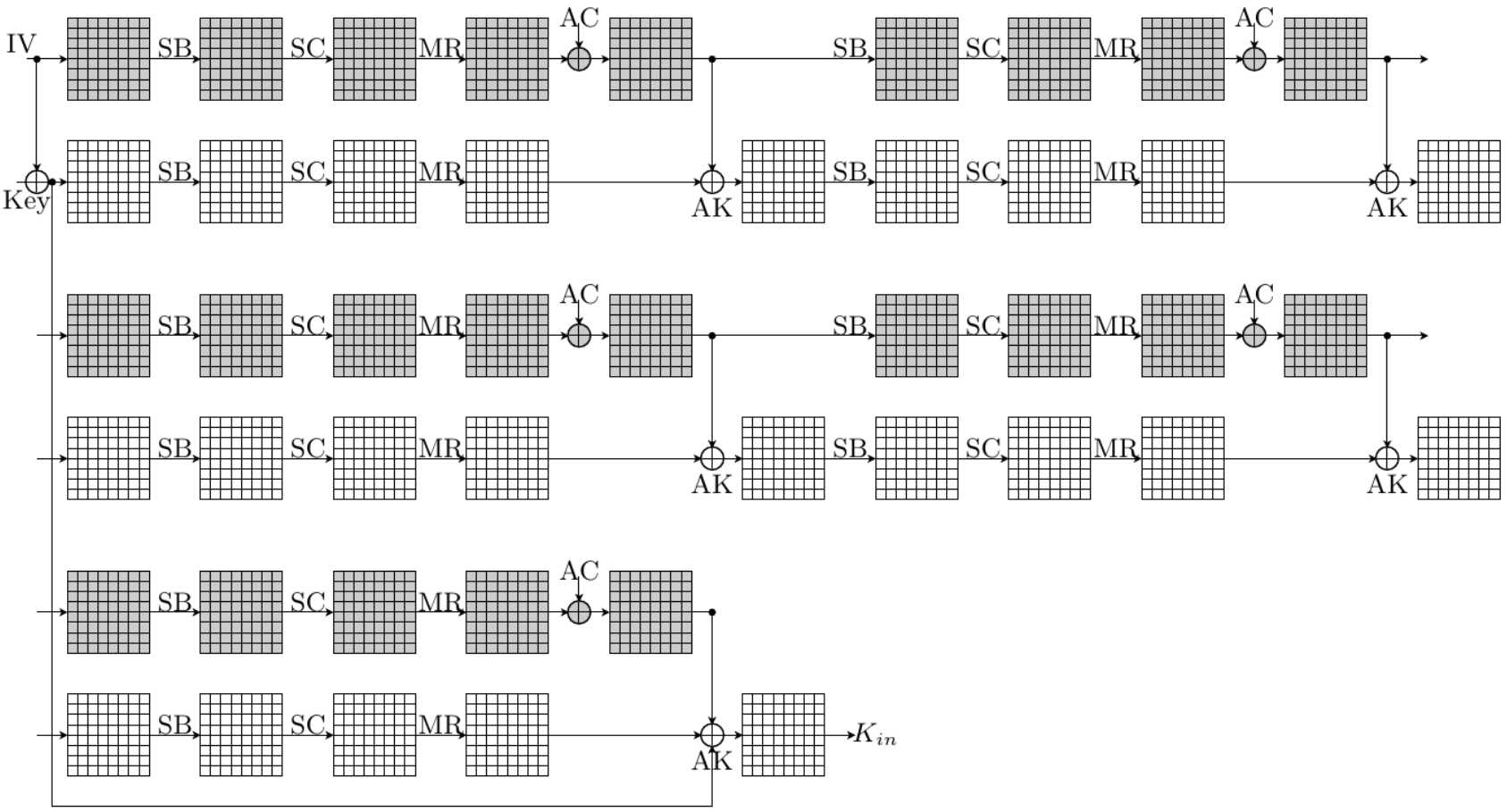
We know collision on T', but not the actual value, and the message pair (M_1, M_2) with structured difference producing T' collision, we want to recover K_{in}

Recovering K_{in}



known value and difference in M, and no difference in T'
recovery K_{in} with 2^{488} time and 2^{384} memory.

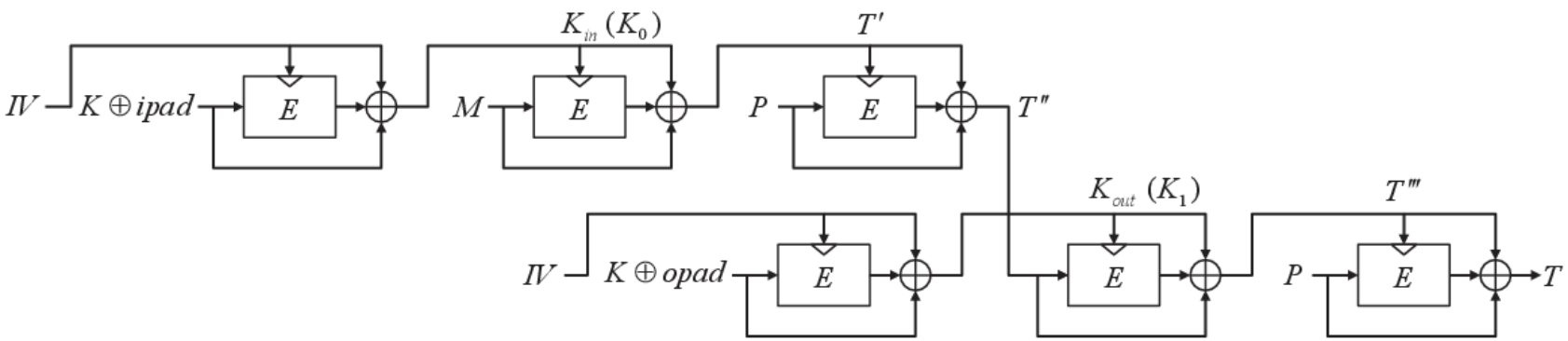
Recovering real key from K_{in}



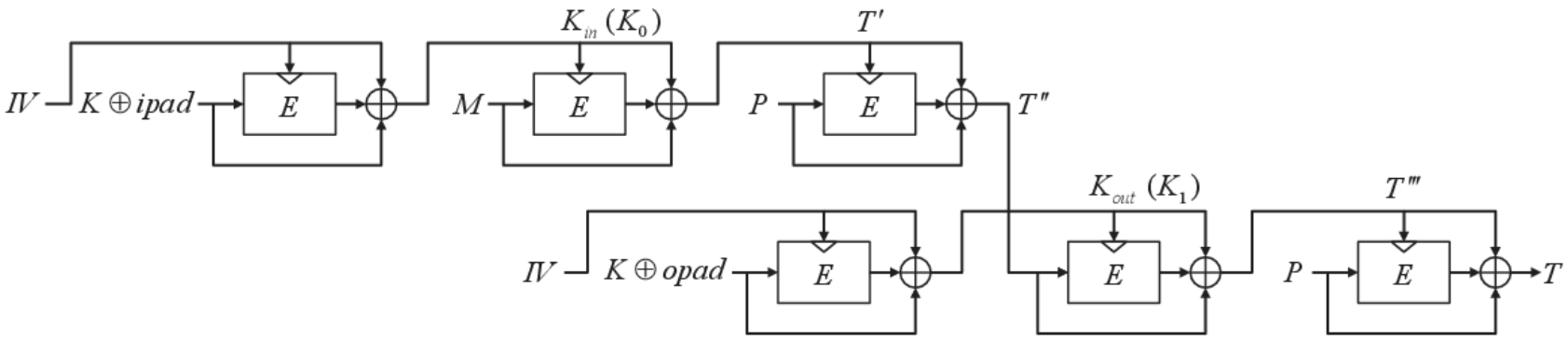
$C(IV, K \oplus ipad) = K_{in}$, recover K with known IV and K_{in}.
Meet-in-the-Middle Preimage attack applies, with 2^{448} time and 2^{64} memory [Wu et al 2012]

5-round NMAC-Whirlpool — Attack Overview

1. find inner collision on T' (done)
2. recover K_{in}, *i.e.*, K_0 for NMAC (done)
3. find special structured near-collision on $V = MR^{-1}(T'' \oplus T''')$.
4. recover K_{out}, *i.e.*, K_1 for NMAC

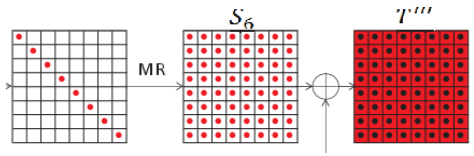


Find structured near-collision on $V = MR^{-1}(T'' \oplus T''')$



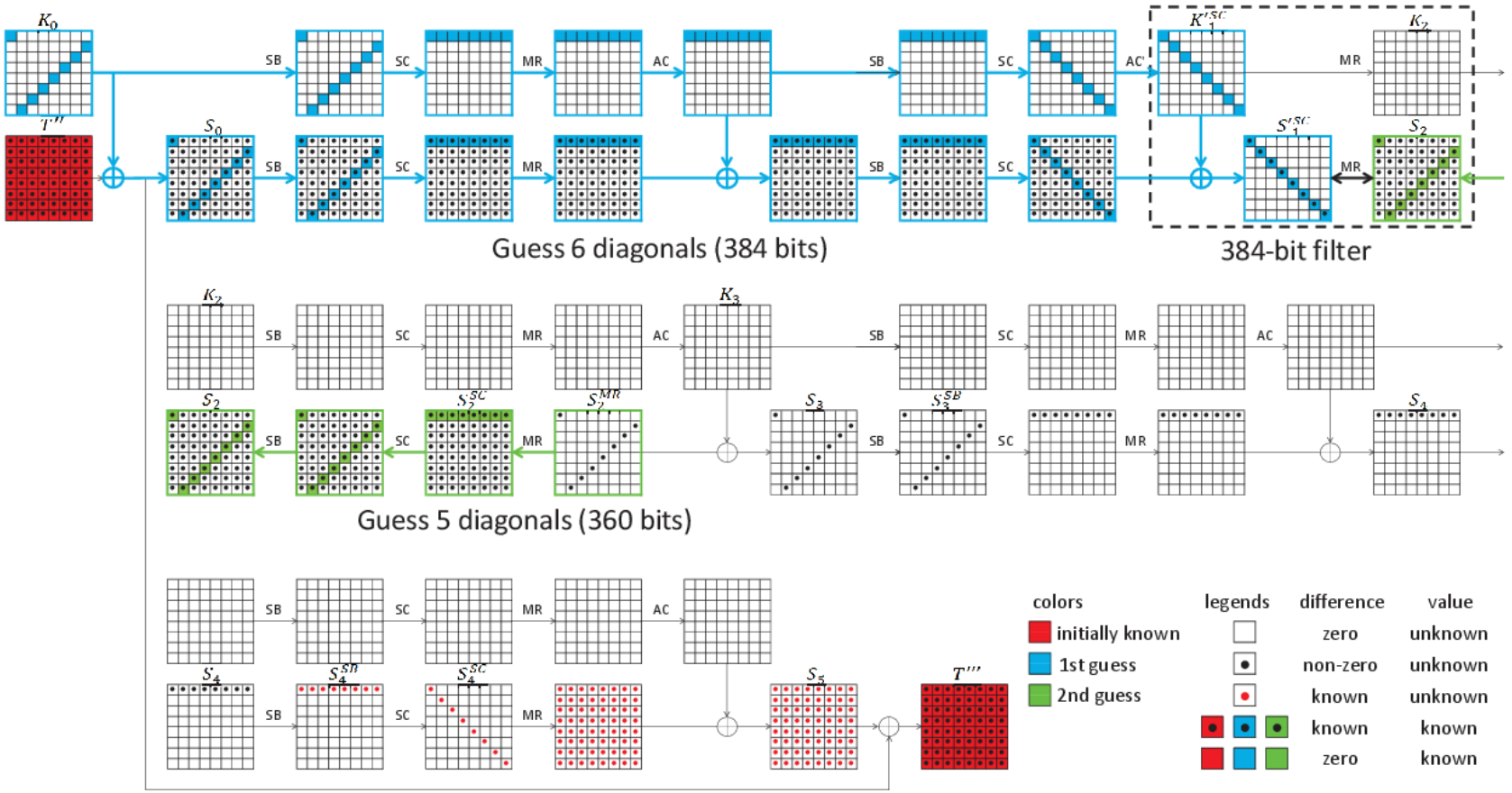
Recovering T''':

1. With know K_{in}, compute (M, T', T'', T) for many M.
2. compute (T''', T) for many randomly chosen T'''.
3. find a collision on T, and recover (M, T''', T'').
4. check if $V = MR^{-1}(T'' \oplus T''')$ is a near collision of the form



Recovering K_{out}

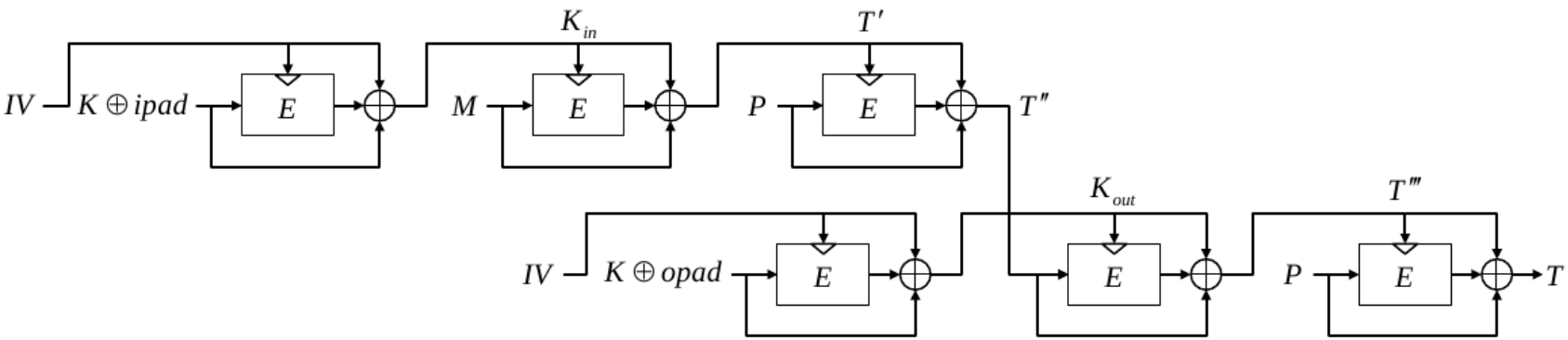
with known value and difference in T'' and T''' and difference of $V = MR^{-1}(T'' \oplus T''')$ in diagonal.



recover K_{out} in time 2^{424} and 2^{360} memory.

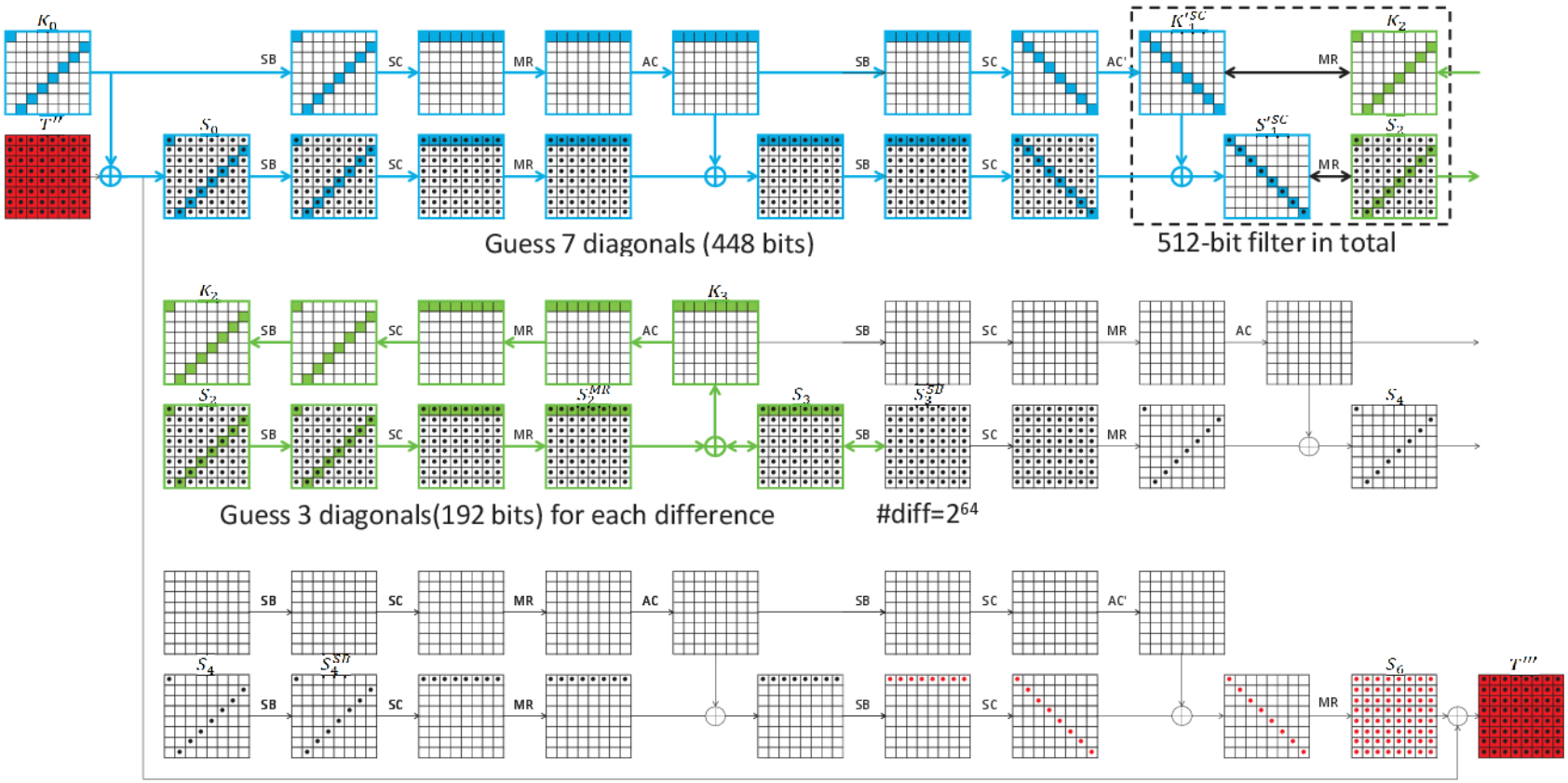
6-round HMAC-Whirlpool — Attack Overview

1. recover the internal value T', with key difference $ipad \oplus opad$, in 2^{256} [PSW2012]
2. recover T'' and T''' as before
3. find 3-collision on $V = MR^{-1}(T'' \oplus T''')$, in time 2^{406} time and memory
4. recover K_{out}
5. recover K from K_{out}



Key recovery for 6-round HMAC-Whirlpool

with known value and difference in T'', T''', and 3-collision with strctured difference in diagonal of $V = MR^{-1}(T'' \oplus T''')$.



recover K_{out} in 2^{448} time and memory, and recover K from K_{out} using preimage attack presented in [SWWW12].

Other PGV modes and MAC modes

- Same attack applies to whirlpool in 4 other PGV modes
- it applies to secret-suffix MAC, secret-prefix MAC, and envelope MAC with similar number of rounds.

Outline

1. Introduction
   - HMAC and NMAC
   - The Whirlpool Hash Function
2. Key Recovery Attacks
   - Attack Overview
   - Key Recovery for 5-round HMAC-Whirlpool
   - Key recovery for 5-round NMAC-Whirlpool
   - Key recovery for 6-round HMAC-Whirlpool in related-key setting
   - Other PGV modes and MAC modes
3. Conclusion

Conclusion

Result Summarization on HMAC/NMAC-Whirlpool

Target	#Rounds	Time	Memory	Data
HMAC-Whirlpool single-key	5	2^{448}	2^{384}	2^{488}
NMAC-Whirlpool single-key	5	2^{448}	2^{384}	2^{488}
HMAC-Whirlpool related-key	6	2^{496}	2^{406}	2^{488}

Comparison

Comparison with previous attacks

Targets	#Attacked Rounds	#Full Rounds	Percentage	Reference
MD5	64	64	100%	[WY05]
HMAC-MD5	No previous attack			
SHA-1 (collision)	80	80	100%	[WYY05]
SHA-1 (preimage)	57	80	71%	[KK12]
HMAC-SHA-1	34	80	43%	[CY06]
Whirlpool (collision)	5	10	50%	[LMRR09]
Whirlpool (preimage)	6	10	60%	[SWWW12]
HMAC-Whirlpool single-key	5	10	50%	Ours
HMAC-Whirlpool related-key	6	10	60%	Ours

End of Talk

Thank you!

Questions?