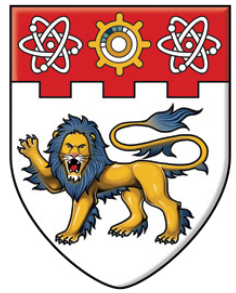
Key Recovery Attack against HMAC/NMAC with Reduced Whirlpool

Jian Guo



NANYANG
TECHNOLOGICAL
UNIVERSITY

Dagstuhl Seminar — Symmetric Cryptography.
Germany, 07 Jan 2014

Based on works:

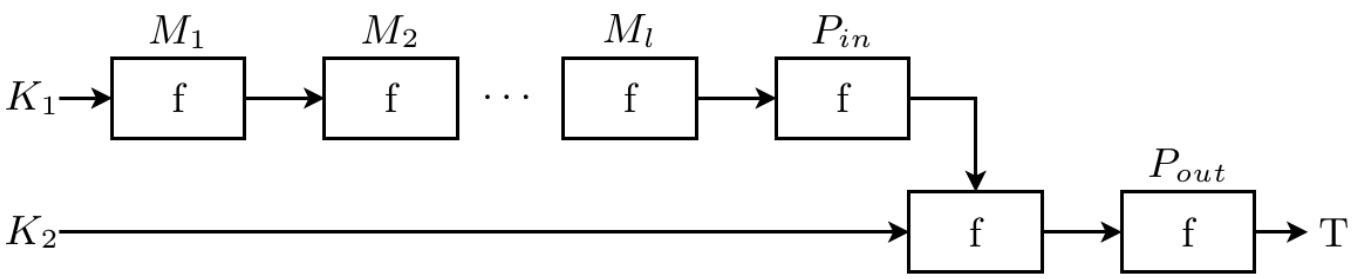
1. Jian Guo, Yu Sasaki, Lei Wang, Shuang Wu, *Cryptanalysis of HMAC/NMAC-Whirlpool*, ASIACRYPT 2013
2. Jian Guo, Yu Sasaki, Lei Wang, Meiqin Wang, Long Wen, *Equivalent Key Recovery Attacks against HMAC and NMAC with Whirlpool Reduced to 7 Rounds*.

Talk Overview

1. Introduction
   - HMAC and NMAC
   - The Whirlpool Hash Function
   - Motivation
2. Key Recovery Attacks
   - The Attack Framework
   - 6-Round Original Key Recovery Attack
   - 7-Round Equivalent Key Recovery Attack
3. Conclusion

HMAC and NMAC

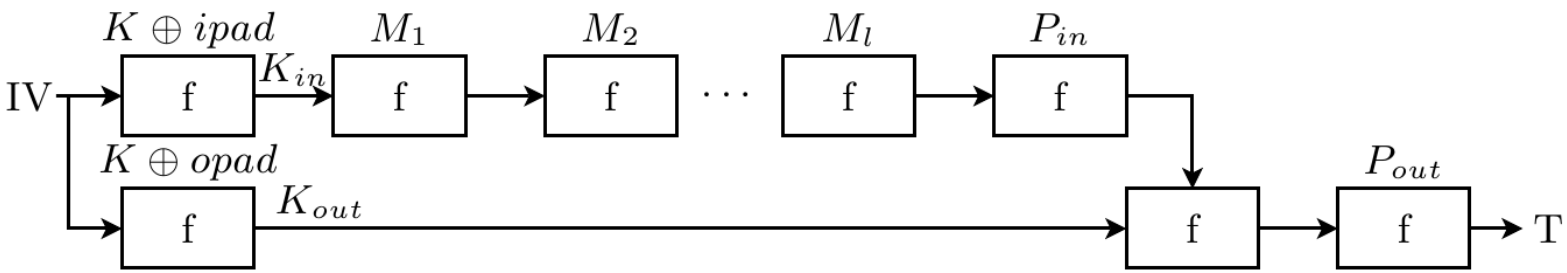
- Designed by Mihir Bellare, Ran Canetti and Hugo Krawczyk in Crypto 1996
- Standarized by ANSI, IETF, ISO, NIST from 1997
- **The** most widely deployed hash-based MAC construction.



NMAC

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.



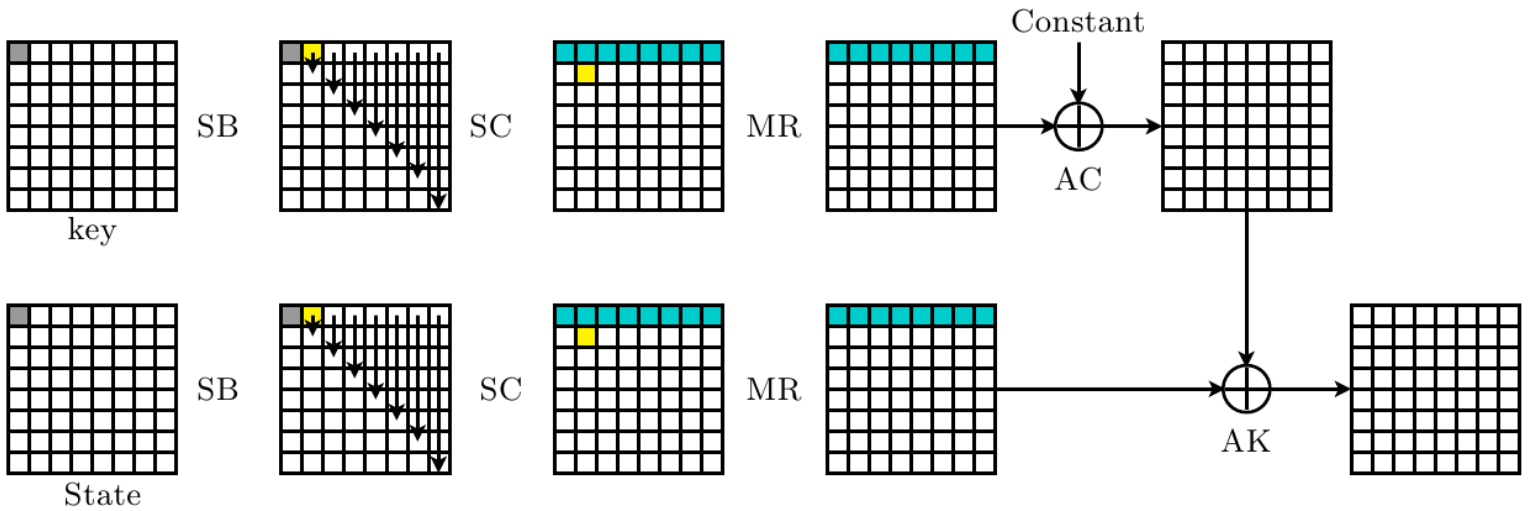
HMAC

Whirlpool

- designed by Barreto and Rijmen in 2000 with 512-bit digest
- standarized by ISO/IEC, approved by NESSIE (New European Schemes for Signatures, Integrity, and Encryption).
- follows Merkle-Damgård strengthening, and Miyaguchi-Preneel mode, *i.e.*, $f(H, M) = E_H(M) \oplus H \oplus M$
- both state and key follow the AES-like process, with 10 rounds.

Whirlpool

- designed by Barreto and Rijmen in 2000 with 512-bit digest
- standarized by ISO/IEC, approved by NESSIE (New European Schemes for Signatures, Integrity, and Encryption).
- follows Merkle-Damgård strengthening, and Miyaguchi-Preneel mode, *i.e.*, $f(H, M) = E_H(M) \oplus H \oplus M$
- both state and key follow the AES-like process, with 10 rounds.



Key: $AC \circ MR \circ SC \circ SB$; State: $AK \circ MR \circ SC \circ SB$

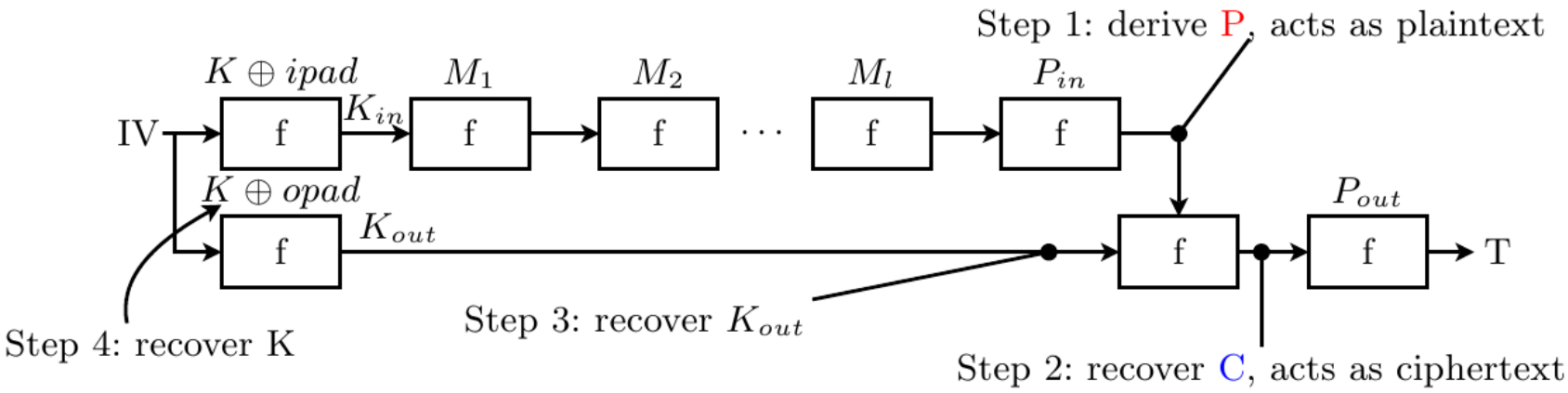
Motivation

AES,1998	Whirlpool, 2000
⇓	⇓
First Cryptanalysis	
Ferguson et al. 2000, etc.	Mendel et al. 2009, etc.
⇓	⇓
Analysis on MAC Applications	
follows naturally	**Ours**

Collision/Preimage attacks against hash function **do not** lead directly to attack on MAC applications.

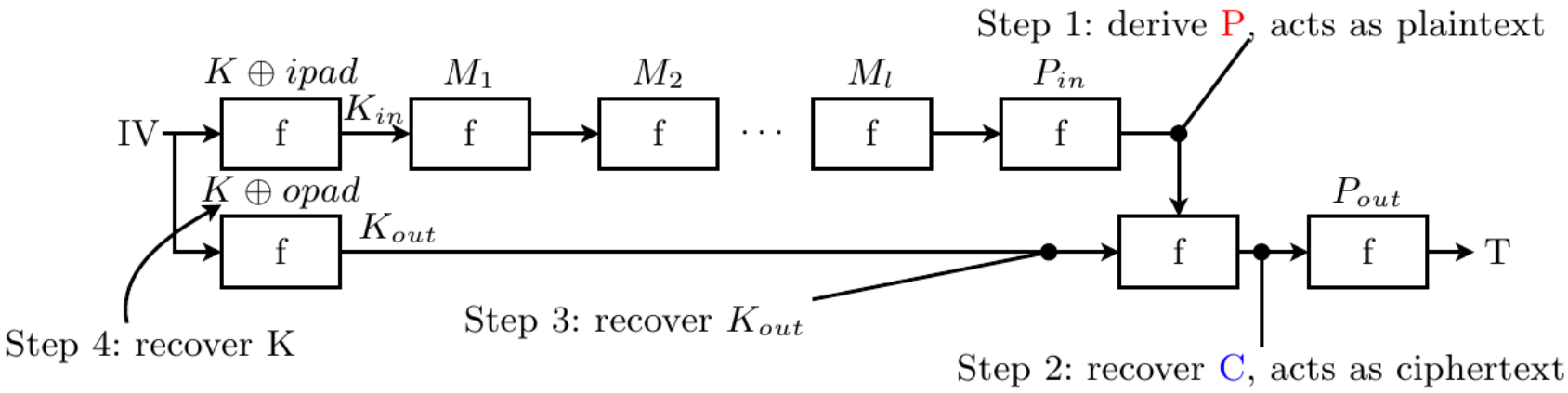
Attack Overview

1. Derive many P



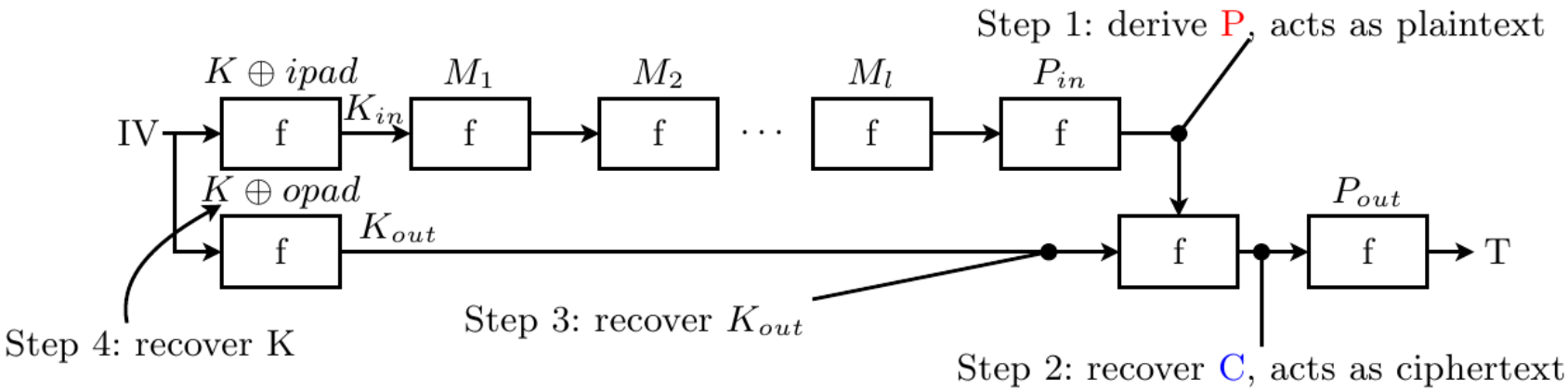
Attack Overview

1. Derive many P
2. Derive corresponding C



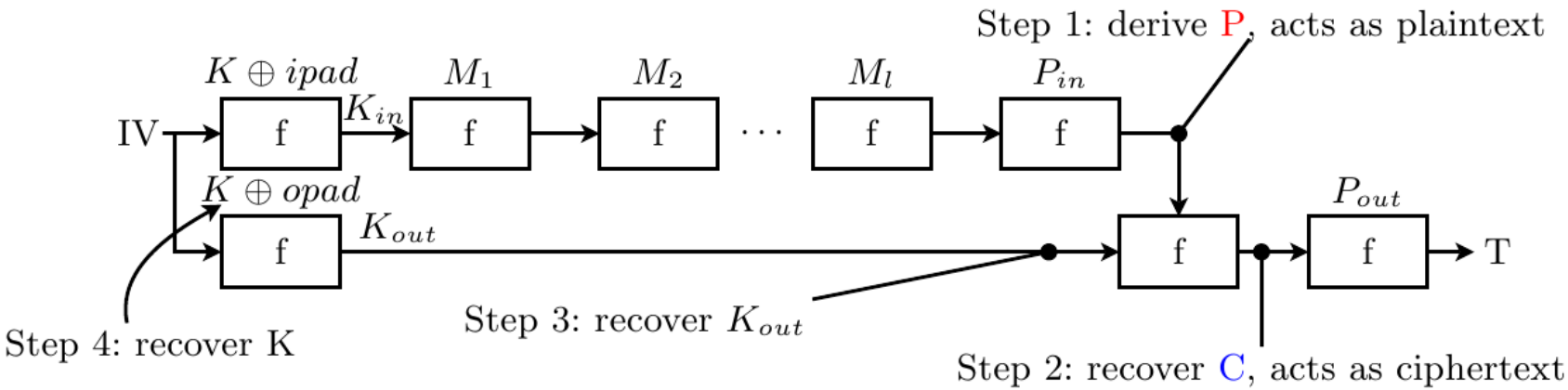
Attack Overview

1. Derive many P
2. Derive corresponding C
3. Recover K_{out} from found (P, C) pairs



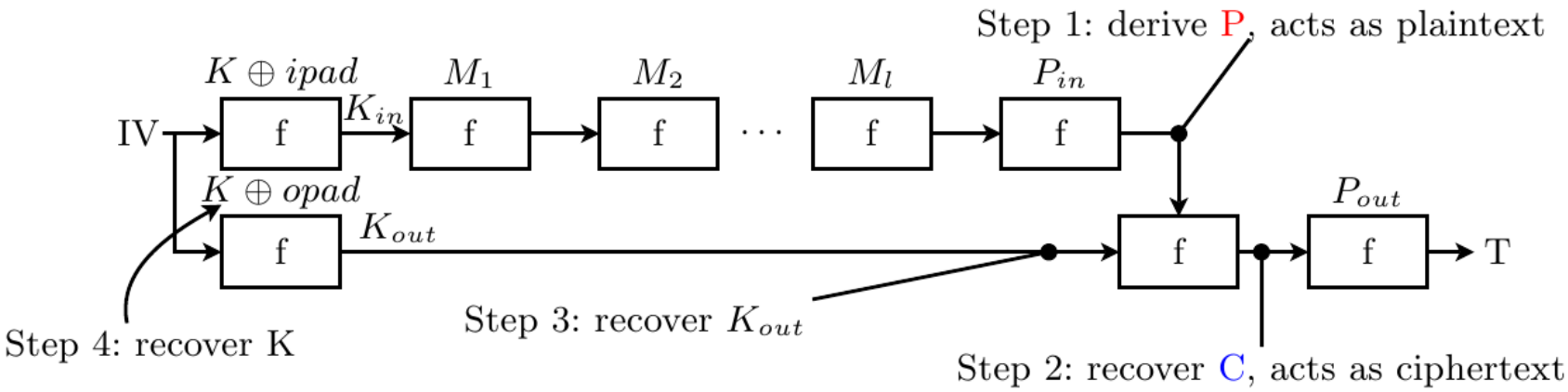
Attack Overview

1. Derive many P
2. Derive corresponding C
3. Recover K_{out} from found (P, C) pairs
4. Recover the original key K from K_{out}



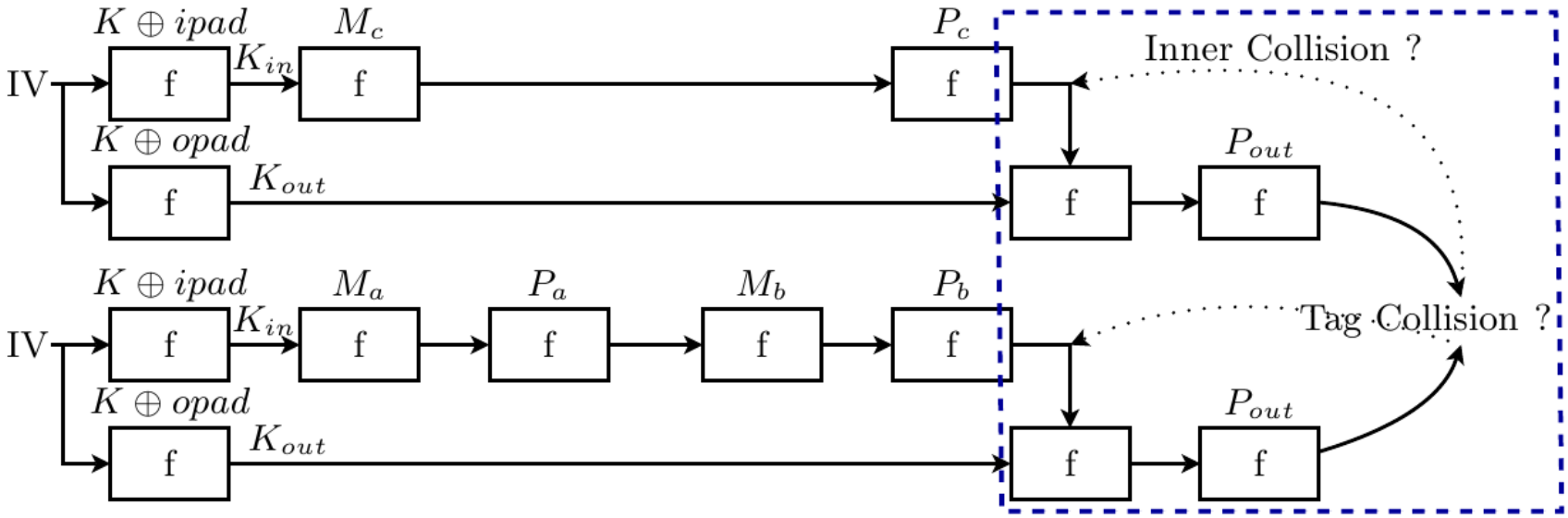
Attack Overview

1. Derive many P
2. Derive corresponding C
3. Recover K_{out} from found (P, C) pairs
4. Recover the original key K from K_{out}
5. Recover K_{in} (or K_1) for NMAC only.



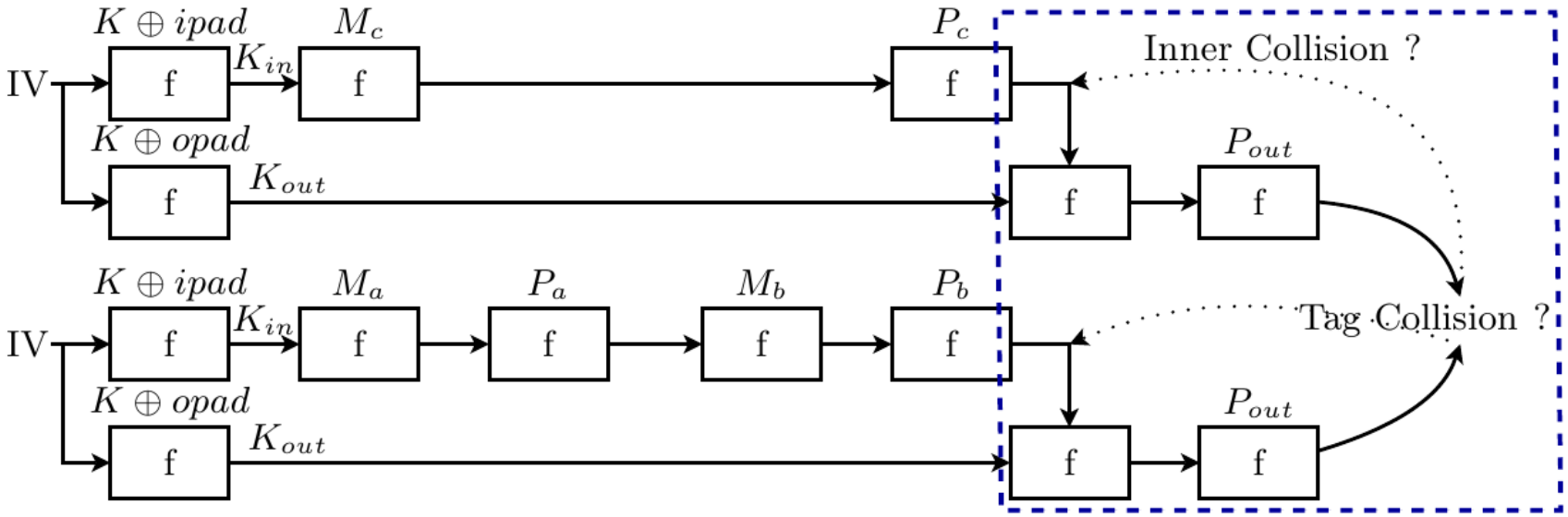
Step 1: Derive P

1. Gaëtan-Peyrin-Wang'13 showed how to derive $h = H(K \oplus ipad \| M_a)$ for some long message M_a of around $2^{n/2}$ blocks.



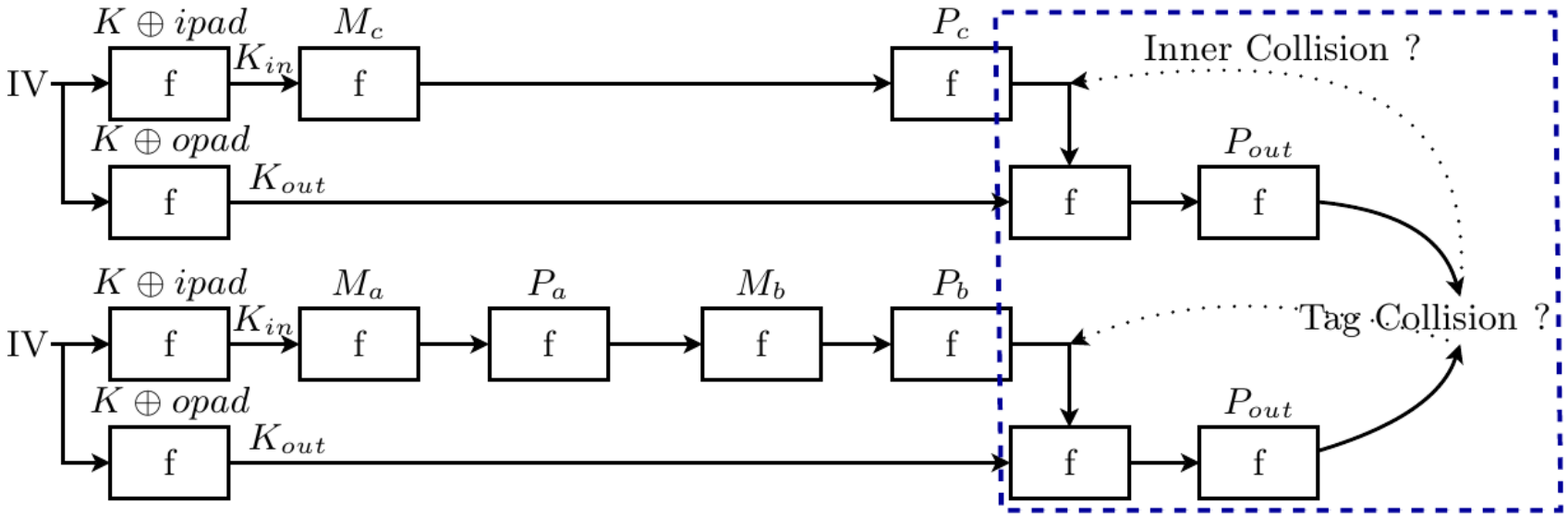
Step 1: Derive P

1. Gaëtan-Peyrin-Wang'13 showed how to derive $h = H(K \oplus ipad \| M_a)$ for some long message M_a of around $2^{n/2}$ blocks.
2. Using unbalanced Meet-in-the-Middle, one can recover $P = H(K \oplus ipad \| M_c \| P_c \| M_d) = f(f(h', P_c), M_d)$, for some 1/2-block M_c, and any M_d with padding satisfied.



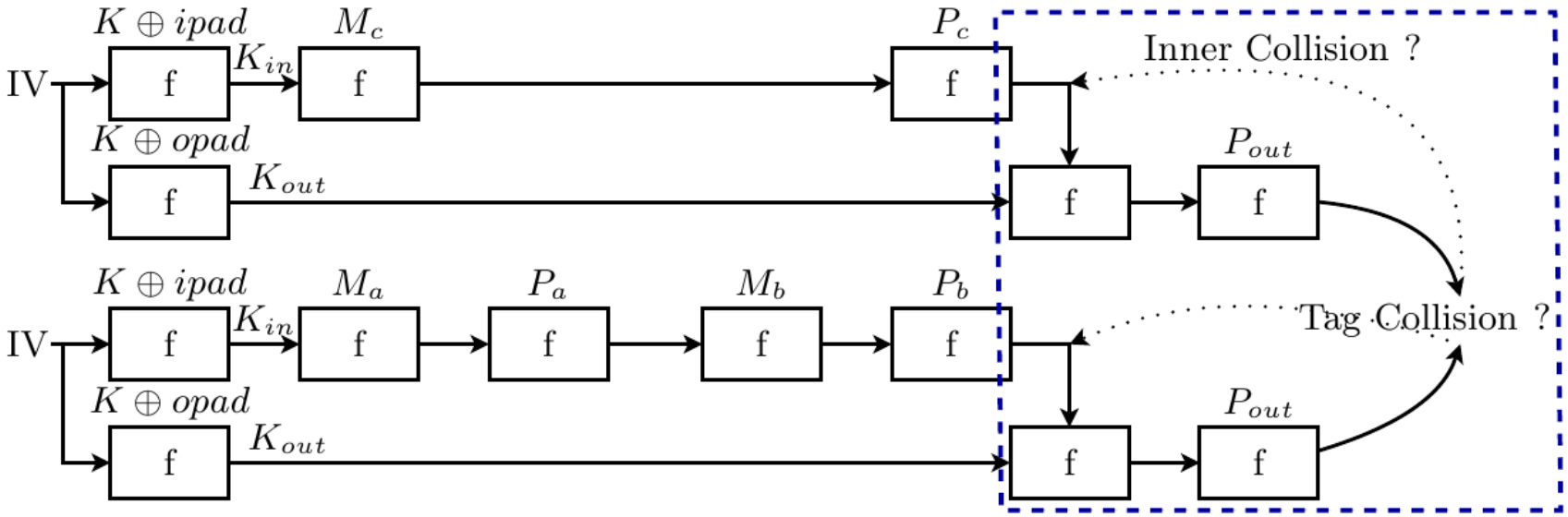
Step 1: Derive P

1. Gaëtan-Peyrin-Wang'13 showed how to derive $h = H(K \oplus ipad \| M_a)$ for some long message M_a of around $2^{n/2}$ blocks.
2. Using unbalanced Meet-in-the-Middle, one can recover $P = H(K \oplus ipad \| M_c \| P_c \| M_d) = f(f(h', P_c), M_d)$, for some 1/2-block M_c, and any M_d with padding satisfied.



Step 1: Derive P

1. Gaëtan-Peyrin-Wang'13 showed how to derive $h = H(K \oplus ipad \| M_a)$ for some long message M_a of around $2^{n/2}$ blocks.
2. Using unbalanced Meet-in-the-Middle, one can recover $P = H(K \oplus ipad \| M_c \| P_c \| M_d) = f(f(h', P_c), M_d)$, for some 1/2-block M_c, and any M_d with padding satisfied.

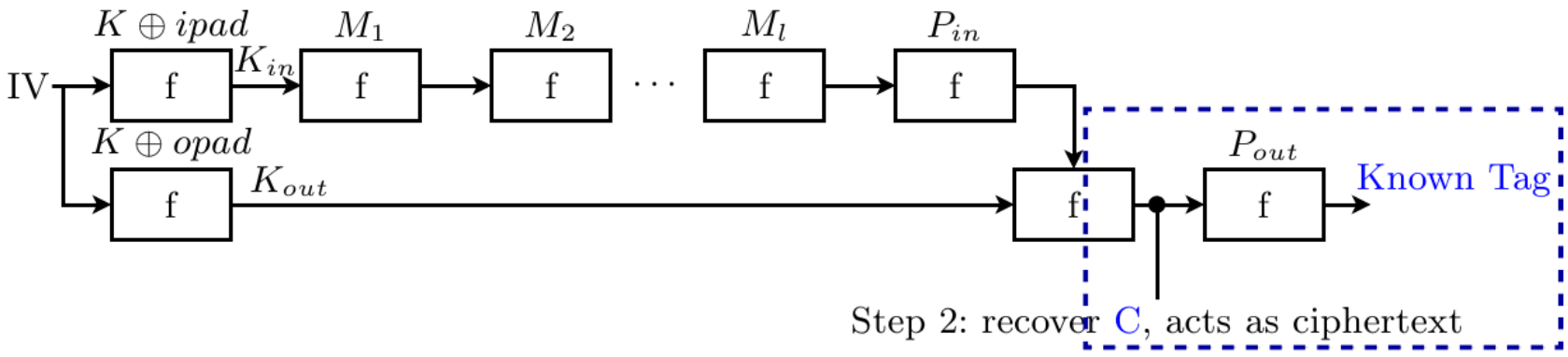


Generic to any f

Step 2: Derive C

The Problem

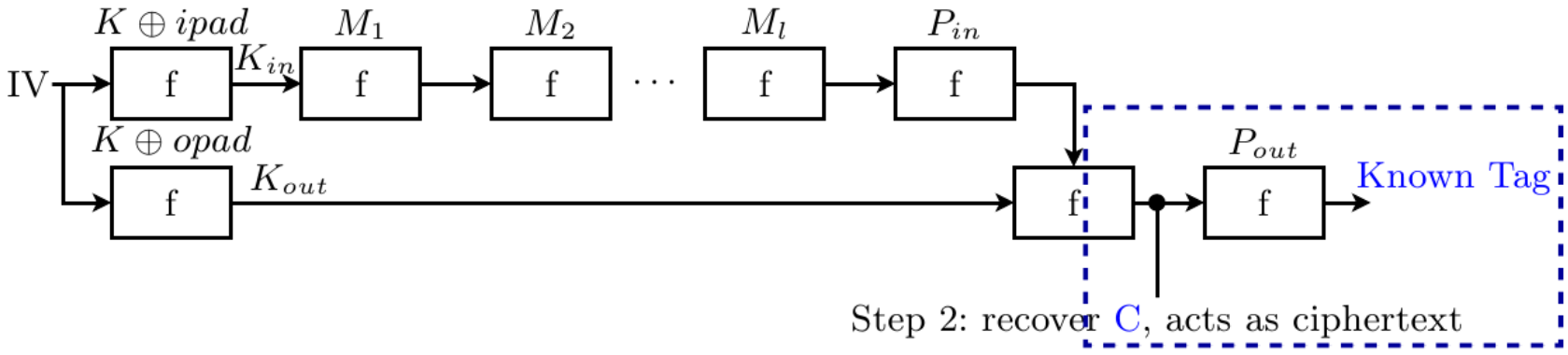
With known Tag value, and fixed message block P_{out}, find input chaining value C.



Step 2: Derive C

The Problem

With known Tag value, and fixed message block P_{out}, find input chaining value C.



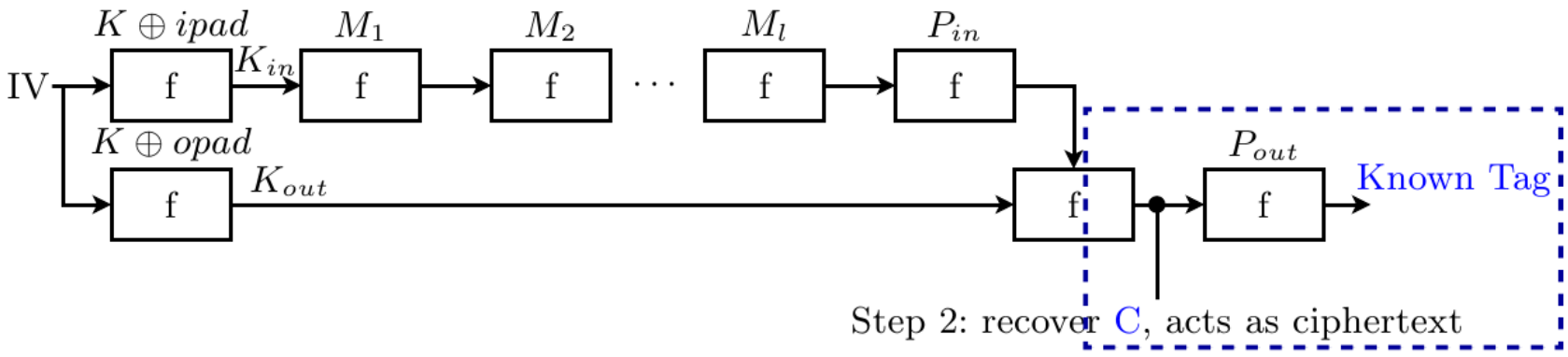
The Solution

Precompute a table $T = f(C, P_{out})$ to obtain many pairs of (C, T)

Step 2: Derive C

The Problem

With known Tag value, and fixed message block P_{out}, find input chaining value C.

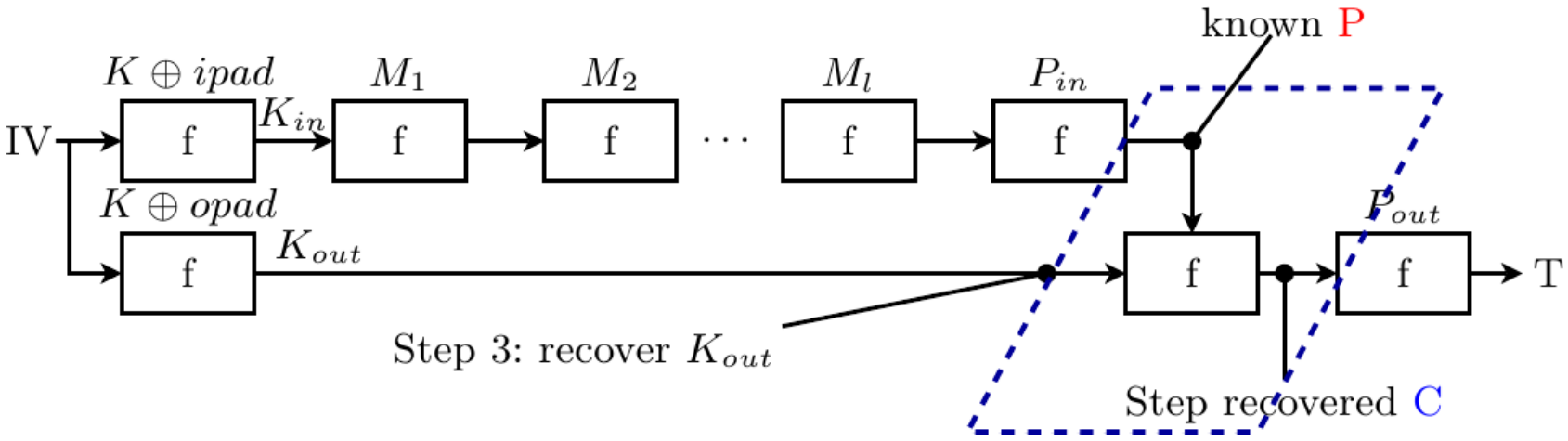


The Solution

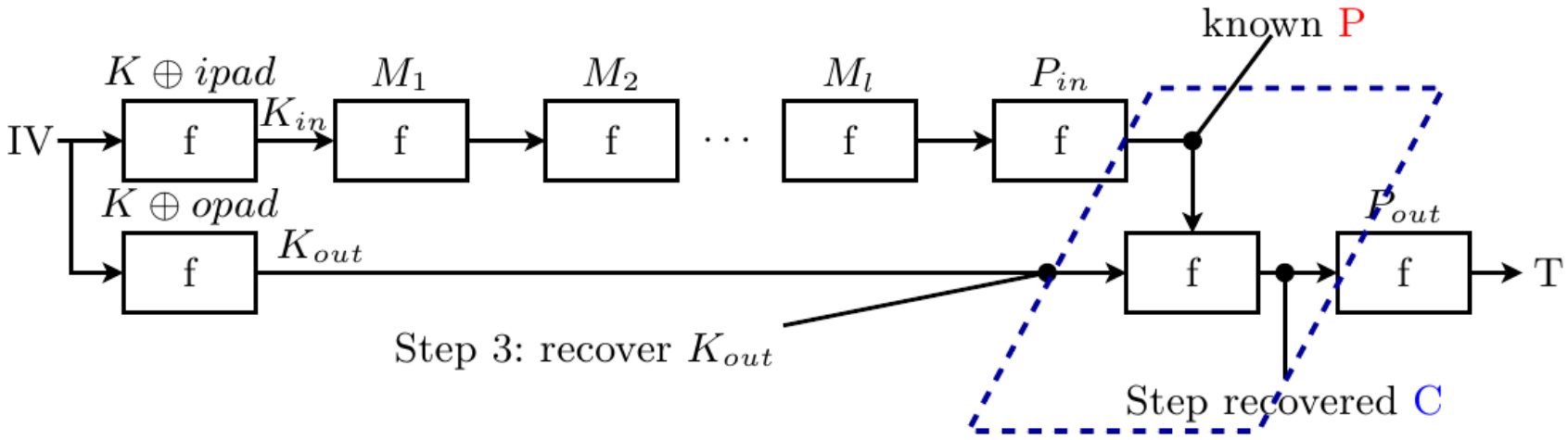
Precompute a table $T = f(C, P_{out})$ to obtain many pairs of (C, T)

Generic to any f

Step 3: Recover K_{out}



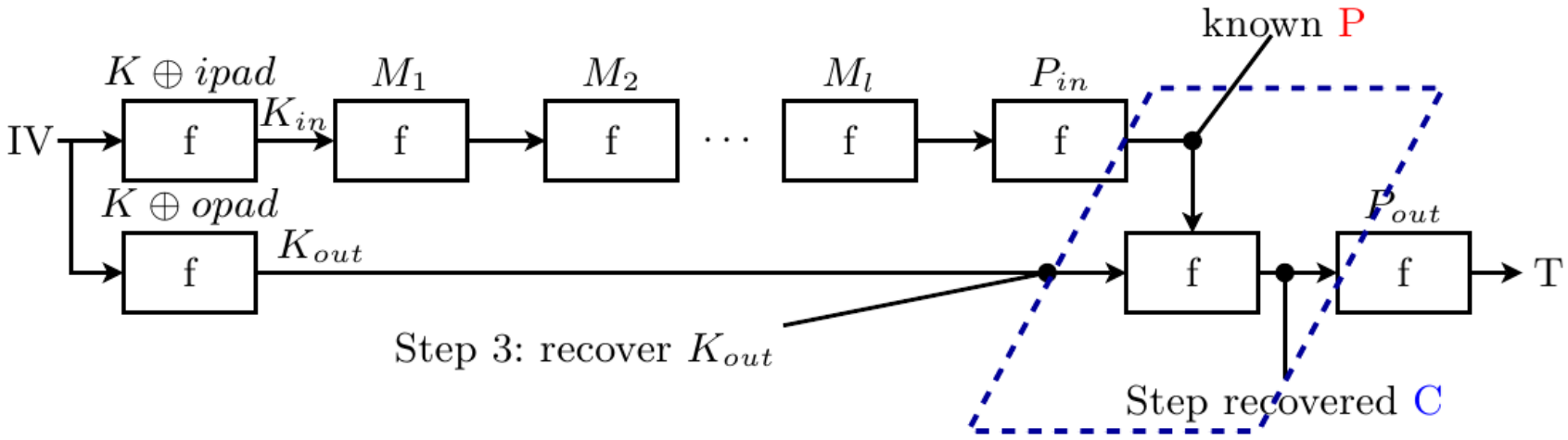
Step 3: Recover K_{out}



$$C = f(K_{out}, P) = E_{K_{out}}(P) \oplus P$$

denote the underlying block cipher as *Whirlpool-BC*

Step 3: Recover K_{out}

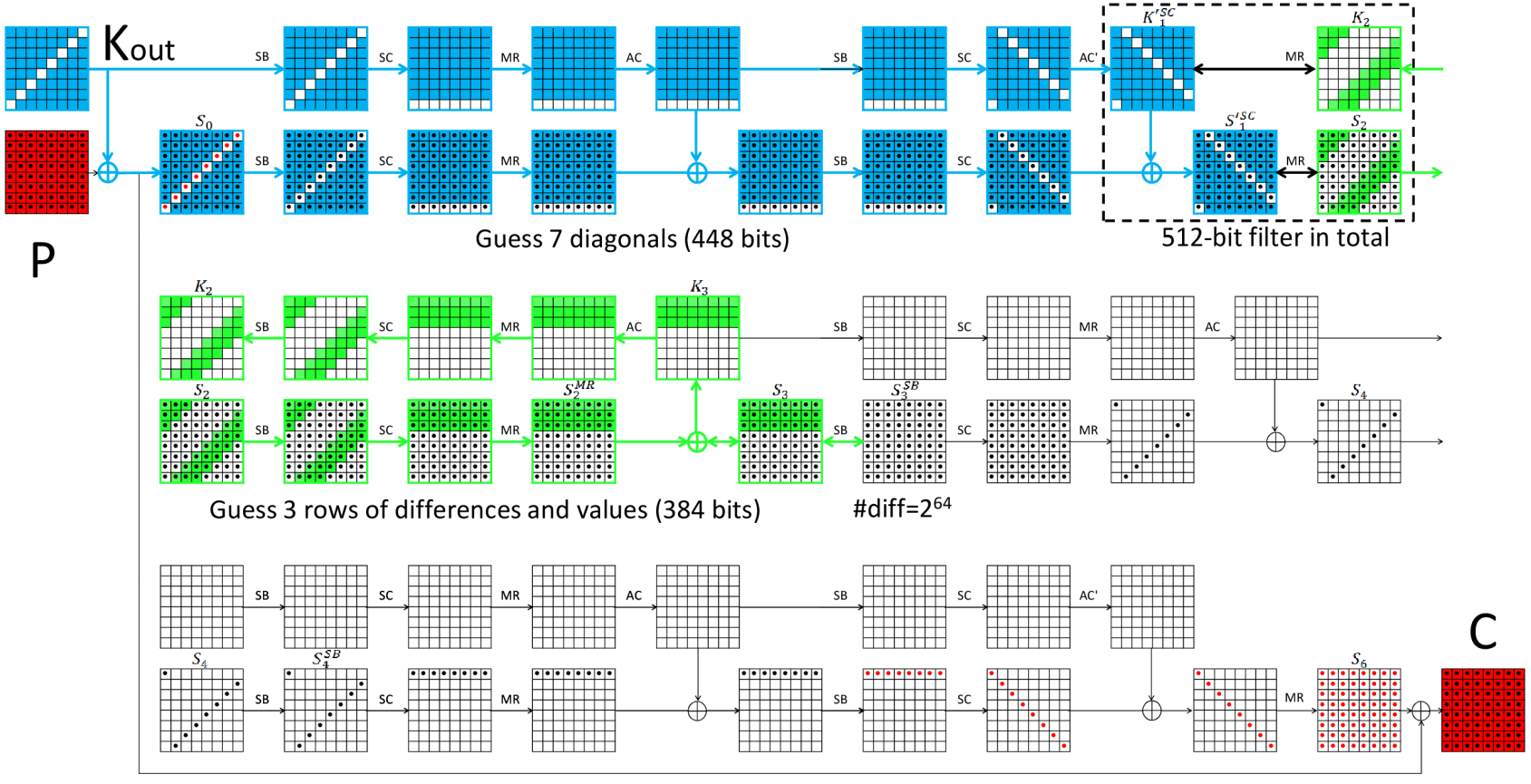


$$C = f(K_{out}, P) = E_{K_{out}}(P) \oplus P$$

denote the underlying block cipher as *Whirlpool-BC*

Known-Plaintext Key Recovery Problem

Step 3: 6-Round Known-Plaintext Attack against Whirlpool-BC

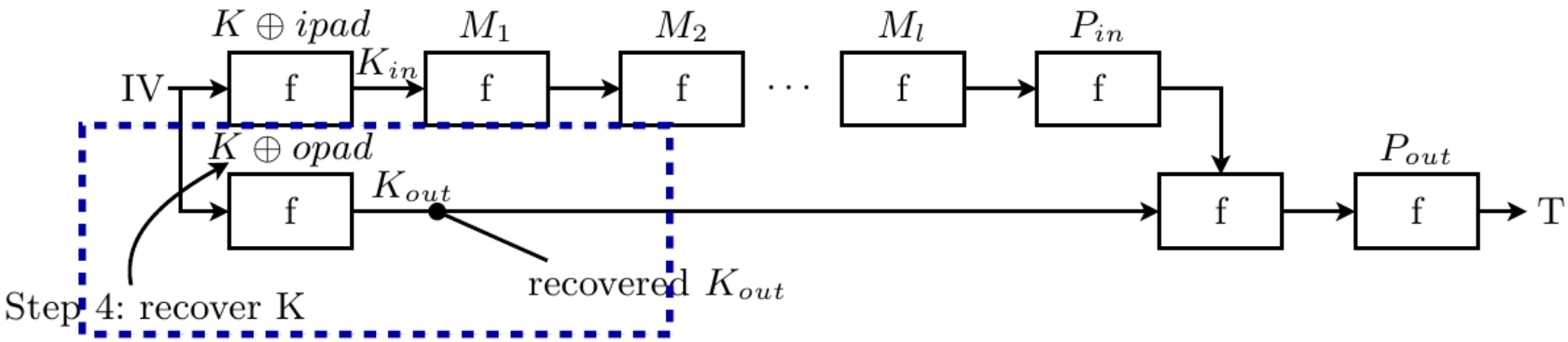


Given many (P, C) pairs, filter for 3-collision with strctured difference in diagonal of $V = MR^{-1}(P \oplus C)$.

Step 4: Recover K

The Problem

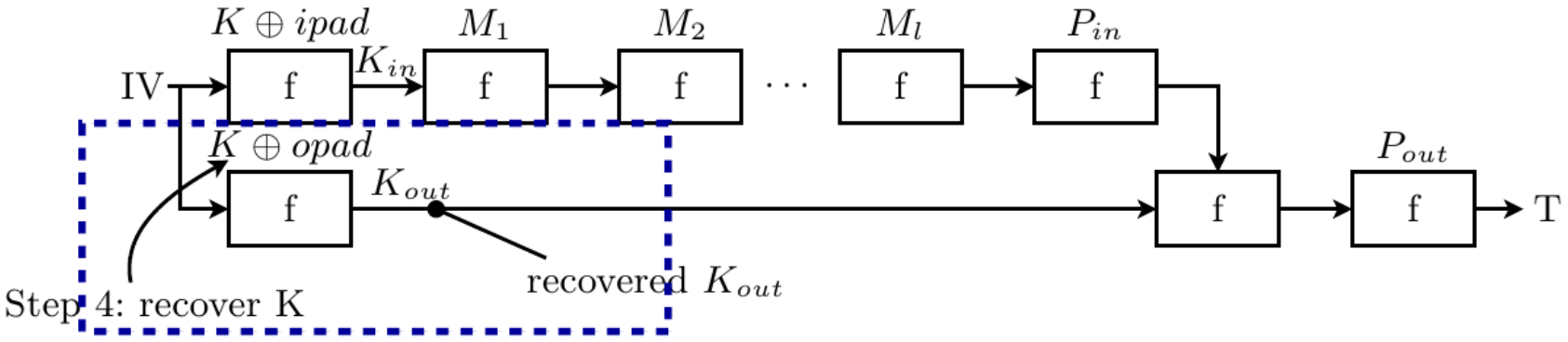
With input chaining IV, output chaining K_{out}, recover K.



Step 4: Recover K

The Problem

With input chaining IV, output chaining K_{out}, recover K.



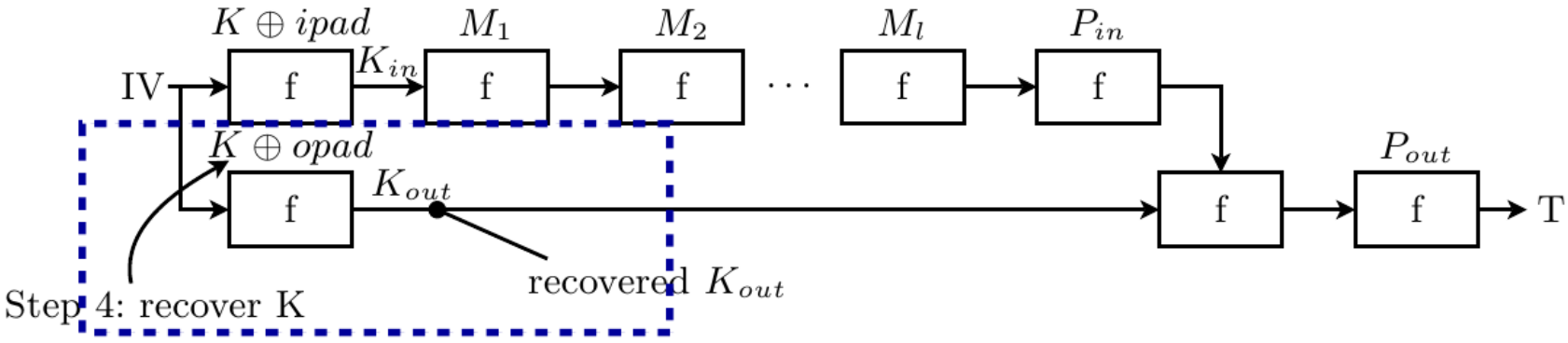
The Solution

Preimage attack by Sasaki et al. ASIACRYPT 2012, working for 6 rounds.

Step 4: Recover K

The Problem

With input chaining IV, output chaining K_{out}, recover K.



The Solution

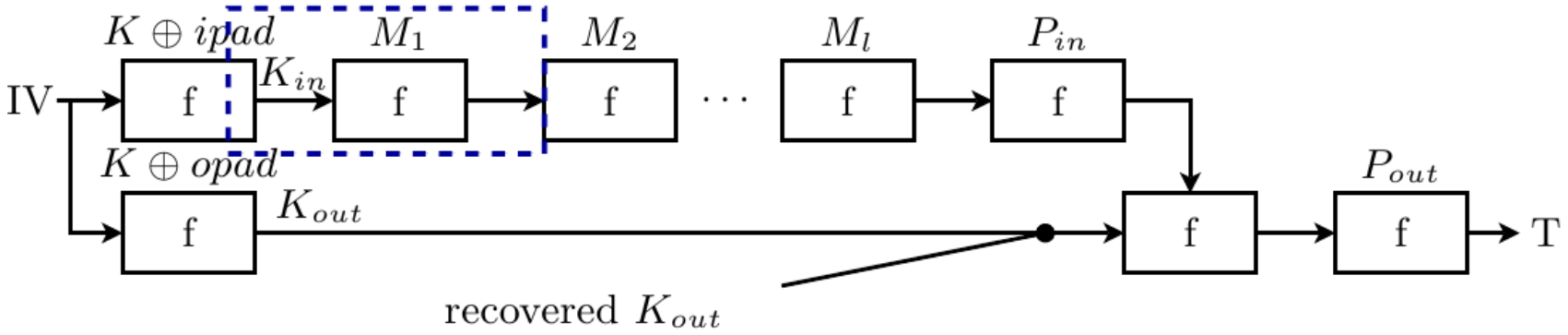
Preimage attack by Sasaki et al. ASIACRYPT 2012, working for 6 rounds.

non-generic, no known result on 7-round preimage attack

Step 5: Recover K_{in}

The Problem

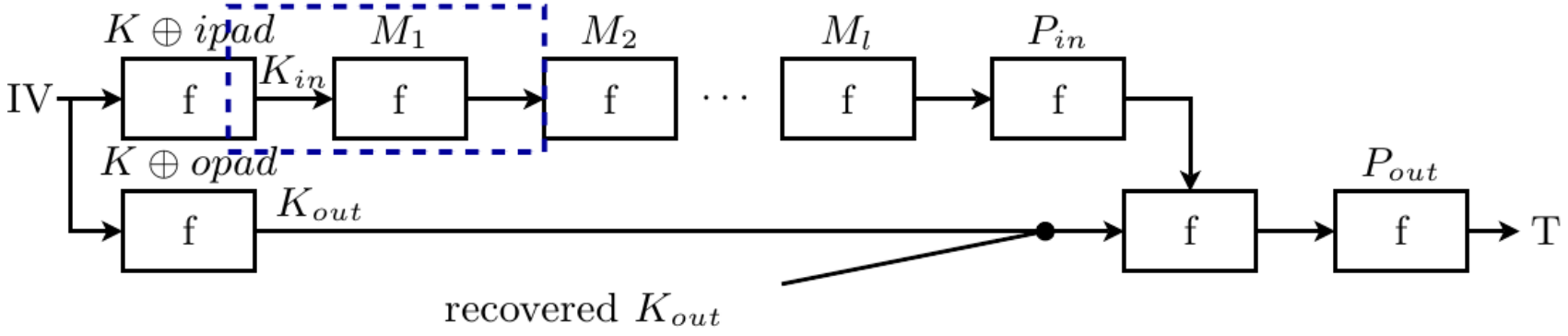
With known K_{out}, chosen M_1, recover K_{in}.



Step 5: Recover K_{in}

The Problem

With known K_{out}, chosen M_1, recover K_{in}.



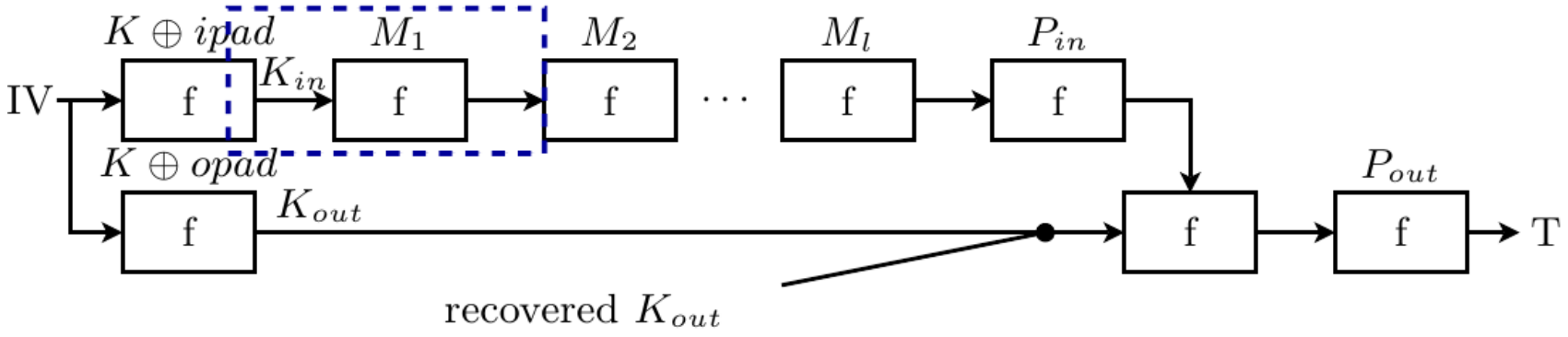
The Solution

Exactly the same procedure as recovering K_{out}.

Step 5: Recover K_{in}

The Problem

With known K_{out}, chosen M_1, recover K_{in}.



The Solution

Exactly the same procedure as recovering K_{out}.

Same number of rounds can be attacked as in Step 3

7-Round Attack using MITM techniques

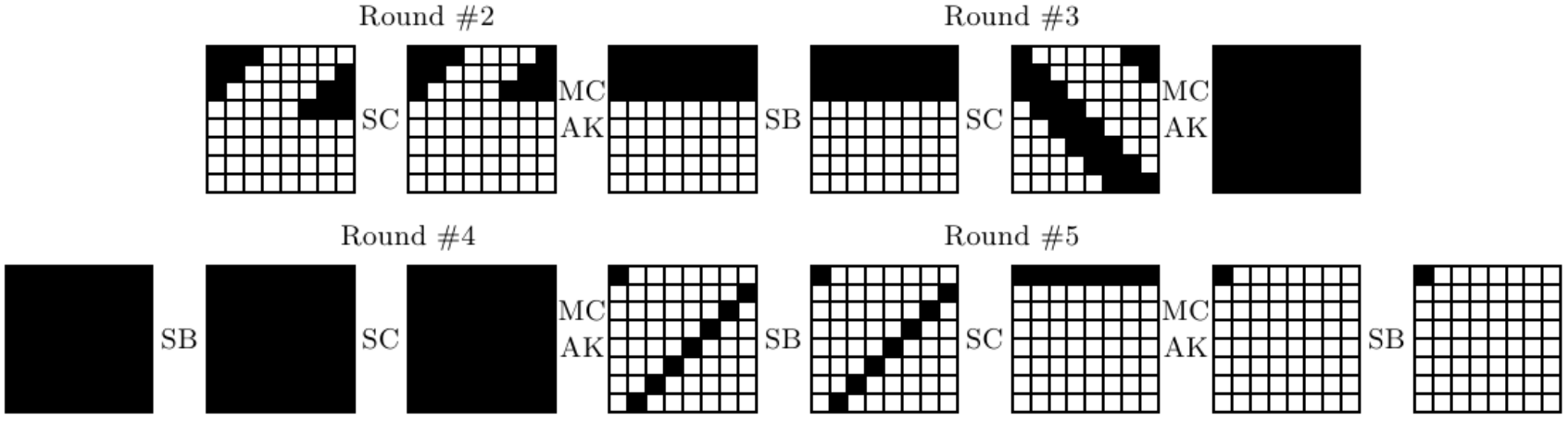
Overview of the 7-Round Attack

- **Idea**: MITM attack, with 4-round distinguisher in the middle + 1 round in front + 2 rounds at the back.

Overview of the 7-Round Attack

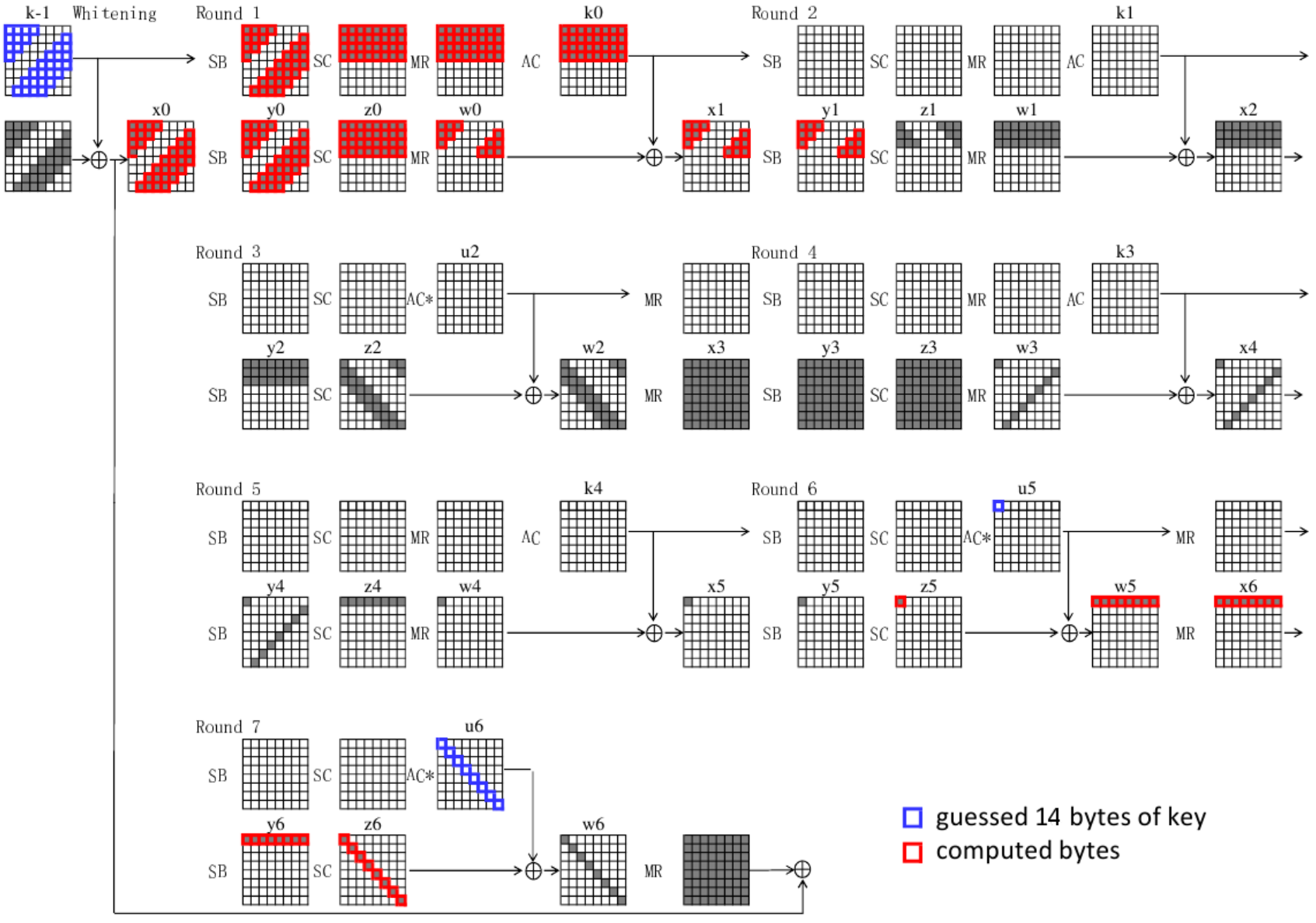
- **Idea**: MITM attack, with 4-round distinguisher in the middle + 1 round in front + 2 rounds at the back.
- **Key problem**: how to convert the current MITM attack on AES-like block cipher in chosen-plaintext model, to that against Whirlpool-BC in known-plaintext model. Simple plaintext filtering does not work anymore ...

The 4-round Distinguisher and Lookup table



- given a pair of input/output pairs $\Delta_{in} = I \oplus I'$ and $\Delta_{out} = O \oplus O'$, path can be uniquely determined by 24+8 byte values
- with help of (I, I'), one can compute the active bytes of the output for any I^* following the input difference.
- precompute a table of the mapping between input and output values, indexed by 32-byte intermediate values + 12-byte Δ_{in} + 1-byte Δ_{out}.

The 7-round Attack I



The 7-round Attack II

1. group the known plaintext-ciphertext pairs (P, C) according to the structures of P.
2. filter all pairs in each structure by w_6.
3. for each pair left, guess 12-byte key values, partially encrypt the plaintext by one round and decrypt the ciphertext by 2 rounds.
4. do lookup against the precomputed table, filter out the wrong guesses by other values in the structure.

Conclusion

Key results:

- Provided a framework to attack HMAC/NMAC
- Original key recovery against 6-round HMAC-Whirlpool
- Equivalent key recovery against 7-round HMAC-Whirlpool

Thank you!

Questions?