Known-Key Distinguisher on Full PRESENT

Céline Blondeau[1] Thomas Peyrin[2] Lei Wang[2,3]

[1]Aalto University, Finland

[2]Nanyang Technological University, Singapore

[3]Shanghai Jiao Tong University, China

CRYPTO 2015

Presented by Pierre Karpman

Outlook

- Introduction
- Our Known-Key Distinguisher
- Application to PRESENT
- Conclusion

Block Cipher

Definition

A block cipher $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ is a family of efficiently invertible permutations on n-bit values, whose index is a k-bit key value.

Applications in Cryptography: a fundamental primitive

- Encryption Scheme: ECB, CBC, CFB, OFB, CTR
- Message Authentication Code: EMAC, CMAC, PMAC
- Authenticated Encryption: GCM, OCB, EAX, CCM
- Hash Function: PGV schemes, MDC-2, MJH, Hirose Scheme

Security Requirement on Block Cipher

A classical security notion: the indistinguishability from an ideal block cipher.

Ideal Block Cipher

Each permutation indexed by a key value is a random permutation. Moreover, any two permutations indexed by distinct key values are completely independent.

Attack Models on Block Cipher

Secret-key Model

- Secret key value
- Impact to Encryption, MAC
- Single-key attack
- Related-key attack

Open-Key Model

- Public key value
- Impact to Hash Function
- Known-key attack
- Chosen-key attack

Attack Models on Block Cipher

- Open-key model is more generous to adversary.
- More rounds are expected to be attacked in open-key model.
- For AES-128 as an example, the number of attacked rounds is

  Secret-key model: 7 rounds [DFJ13];

  Open-key model: 10 (full) rounds [Gilbert14].

Attack Models on Block Cipher

- Open-key model is more generous to adversary.
- More rounds are expected to be attacked in open-key model.
- For AES-128 as an example, the number of attacked rounds is

  Secret-key model: 7 rounds [DFJ13];

  Open-key model: 10 (full) rounds [Gilbert14].

Interestingly the situation for standardized lightweight block cipher PRESENT is rather different, which motivates this research.

PRESENT Cipher

- ISO/IEC standard lightweight block cipher
- Block size is 64 bits; Key size is 80 bits (referred to as PRESENT-80) or 128 bits (referred to as PRESENT-128).
- Composed of 31 rounds:

  Each round consists of a round-key XOR, an Sbox layer and a simple linear bit permutation layer

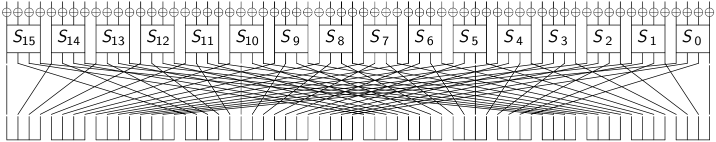


Figure: One round of PRESENT

Previous Analysis Results on PRESENT

- Most scrutinized lightweight cipher.
- Multidimensional linear attack is the most powerful one: easy-to-trace linear trails with large correlations
- Link between differential property and linear correlation in [BN14]:

  A multidimensional linear distinguisher can be converted to a truncated differential distinguisher.

Previous Analysis Results on PRESENT

	#rounds	Version	Attack	Reference
Secret-key Model	16	80	differential	[Wang08]
	19	128	algebraic differential	[AC09]
	19	128	multiple differential	[BN13]
	25	128	linear	[NSZ+09]
	26	80	multidimensional linear	[Cho10]
	26	80	truncated differential	[BN14]
Open-key Model	18	80	differential rebound	[KS+12]
	26	80	linear	[LR15]
	27	128	linear	[LR15]

Our Results on PRESENT

	#rounds	Version	Attack	Reference
Secret-key Model	16	80	differential	[Wang08]
	19	128	algebraic differential	[AC09]
	19	128	multiple differential	[BN13]
	25	128	linear	[NSZ+09]
	26	80	multidimensional linear	[Cho10]
	26	80	truncated differential	[BN14]
Open-key Model	18	80	differential rebound	[KS+12]
	26	80	linear	[LR15]
	27	128	linear	[LR15]
	31 (full)	80/128	truncated differential	Ours

Known-Key Distinguisher

- Key is known to the distinguisher
- Improve estimation of the security margin of block cipher
- Encompass the scenario of block cipher-based hash function
- The goal for an attacker:

  generate input/output pairs with a certain property, such that the complexity for the target block cipher is lower than the generic complexity when dealing with an ideal block cipher

  – target block cipher: open access to internal states to exploit structural weakness;

  – ideal block cipher: black-box access to encryption and decryption oracles

Our Known-Key Distinguisher

Distinguishing property

Find a set of N plaintexts, such that they all have the same value on s pre-determined bits and such that there is a bias on the number of collisions observed on q pre-determined bits of corresponding ciphertexts

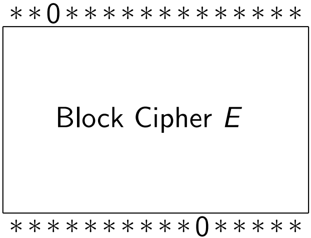


Figure: Our distinguisher model

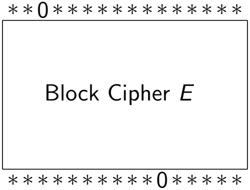
Our Known-Key Distinguisher

Distinguishing property

Find a set of N plaintexts, such that they all have the same value on s pre-determined bits and such that there is a bias on the number of collisions observed on q pre-determined bits of corresponding ciphertexts

Generic attack on an ideal block cipher:

1. Pick N random plaintexts having the same values on s pre-determined bit positions
2. Query them, and count the number of collisions on the q pre-determined bit positions of corresponding ciphertexts



Application to PRESENT

It is important to study known-key distinguishers on PRESENT.

- a natural candidate to build a lightweight hash function
- DM-PRESENT and H-PRESENT in [BL+08]

Application to PRESENT

It is important to study known-key distinguishers on PRESENT.

- a natural candidate to build a lightweight hash function
- DM-PRESENT and H-PRESENT in [BL+08]

We decided to base our distinguisher on truncated differential attacks, because

- it can reach the maximum number of attacked rounds
- it is easier to handle than multidimensional linear attack in the known-key setting

Application to PRESENT

It is important to study known-key distinguishers on PRESENT.

- a natural candidate to build a lightweight hash function
- DM-PRESENT and H-PRESENT in [BL+08]

We decided to base our distinguisher on truncated differential attacks, because

- it can reach the maximum number of attacked rounds
- it is easier to handle than multidimensional linear attack in the known-key setting

On the other hand,

- its statistical bias is small, and a large number of plaintexts is necessary
- pre- and post-adding extra differential characteristics cannot work well, since they reduce #available plaintexts.

Overview of Our Distinguisher on PRESENT

It consists of

- Meet-in-the-middle layer
- Truncated differential layer

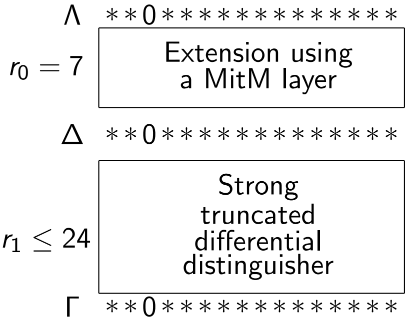


Figure: Overview of our distinguisher

Truncated Differential Layer

- [BN14] studies the link between the probability of a truncated differential and the capacity of a multidimensional linear approximation.

Truncated Differential Layer

- [BN14] studies the link between the probability of a truncated differential and the capacity of a multidimensional linear approximation.
- Truncated differential with strong bias on PRESENT:

  both plaintext and ciphertext have only one Sbox with no difference.

Truncated Differential Layer

- [BN14] studies the link between the probability of a truncated differential and the capacity of a multidimensional linear approximation.
- Truncated differential with strong bias on PRESENT:

  both plaintext and ciphertext have only one Sbox with no difference.
- The truncated differential in our attack:
  - Plaintext: S_{13} has no difference
  - Ciphertext: one of S_5, S_7, S_{13} or S_{15} has no difference

Truncated Differential Layer

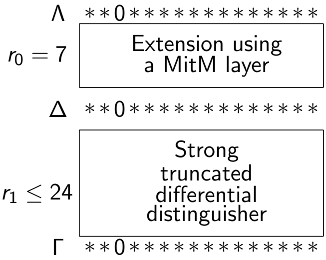
- [BN14] studies the link between the probability of a truncated differential and the capacity of a multidimensional linear approximation.
- Truncated differential with strong bias on PRESENT:

  both plaintext and ciphertext have only one Sbox with no difference.
- The truncated differential in our attack:
  - Plaintext: S_{13} has no difference
  - Ciphertext: one of S_5, S_7, S_{13} or S_{15} has no difference
- Such a truncated differential on 24-round PRESENT:
  - its probability is $2^{-4} + 2^{-62.77}$
  - for an ideal block cipher, the probability is 2^{-4}

Meet-in-the-Middle Layer

It sets constraints only on its input and output, which maintains as many as possible valid inputs to truncated differential layer

- input bit constraints: define the distinguishing property.
- output bit constraints: consistent with truncated differential.



Meet-in-the-Middle Layer

Identify all valid plaintexts efficiently:

- meet-in-the-middle approach due to small Sbox and bit-permutation linear layer:

  in two rounds, an input bit interacts with few other bits, and impacts to only partial outputs bits

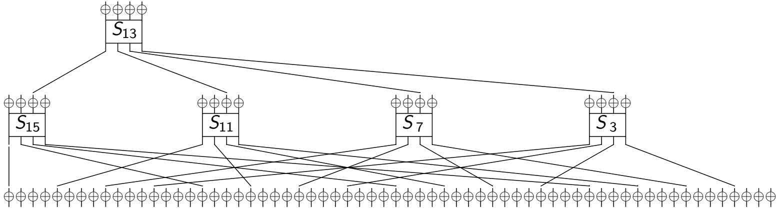
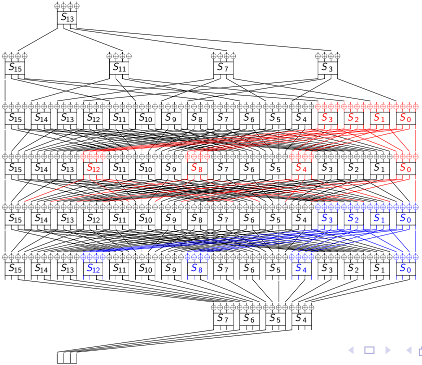


Figure: Propagation for one bit in two rounds

Meet-in-the-Middle Layer

Attack procedure on 7-round PRESENT:

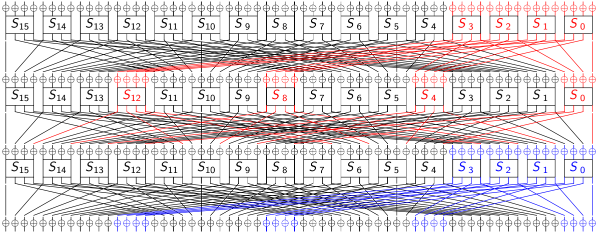
1. Guess and forward compute the first two rounds
2. Guess and backward compute the last one round and half
3. Gradually match the two independent computations through the middle three rounds



Meet-in-the-Middle Layer

Gradually match through the middle three rounds

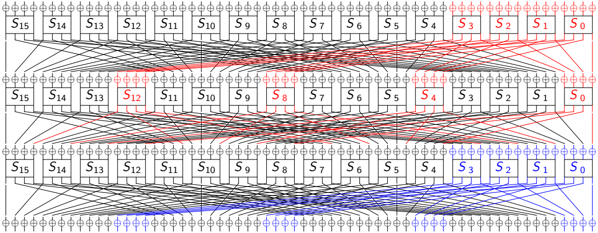
- divide into 4-Sbox groups
- forward: $[S_{4i}, S_{4i+1}, S_{4i+2}, S_{4i+3}]$ as group TF_i
  group TF_0 as an example: red color
- backward: $[S_{4i}, S_{4i+4}, S_{4i+8}, S_{4i+12}]$ as group TB_i
  group TB_0 as an example: blue color



Meet-in-the-Middle Layer

The procedure of gradually matching:

1. guess and compute each group independently
2. merge TF_i and TB_i, and store the results in table T_i
3. merge T_0 and T_1, T_2 and T_3, independently, and store the results in $T_{0,1}$ and $T_{2,3}$ respectively
4. merge $T_{0,1}$ and $T_{2,3}$



Our Results on PRESENT

- #valid messages from MitM layer: 2^{56}
  it contributes to 2^{111} pairs
- Complexity: 2^{56} table lookups and 2^{56} encryptions
- Success probability:

#Rounds	C'_{r-7}	$P_S(2^{111})$	$P_S(2^{109})$
27	$2^{-48.33}$	100%	100%
28	$2^{-50.94}$	99.8%	93.0%
29	$2^{-53.55}$	68.6%	59.5%
30	$2^{-56.16}$	53.2%	51.5%
31	$2^{-58.77}$	50.5%	50.3%

Our Results on PRESENT

- #valid messages from MitM layer: 2^{56}
  it contributes to 2^{111} pairs
- Complexity: 2^{56} table lookups and 2^{56} encryptions
- Success probability:

#Rounds	C'_{r-7}	$P_S(2^{111})$	$P_S(2^{109})$
27	$2^{-48.33}$	100%	100%
28	$2^{-50.94}$	99.8%	93.0%
29	$2^{-53.55}$	68.6%	59.5%
30	$2^{-56.16}$	53.2%	51.5%
31	$2^{-58.77}$	50.5%	50.3%

Overall, with 2^{56} plaintexts and 2^{56} computations, we distinguish PRESENT-80/128 from ideal block cipher with success probability 50.5%.

Conclusion

- a known-key distinguisher on full PRESENT
- the very first non-random property found for full PRESENT
- it is also applicable to DM-PRESENT and H-PRESENT
- our work raises first concerns on the possibility to use PRESENT to build hash functions

Conclusion

- a known-key distinguisher on full PRESENT
- the very first non-random property found for full PRESENT
- it is also applicable to DM-PRESENT and H-PRESENT
- our work raises first concerns on the possibility to use PRESENT to build hash functions

Future work:

- can our attack be simplified or complexity be improved ?
- can we gain something more by choosing the key instead of only knowing it?

Thank you for your attention!