FSE 2011

A Single Key Attack on the Full GOST Block Cipher

Takanori ISOBE

Sony Corporation

Outline

- **Background and Result**
- **GOST Block Cipher**
- **Known Techniques**
  - 3-subset Meet in the Middle Attack
  - Reflection Attack
- **Reflection-MITM attack (R-MITM)**
- **R-MITM attack on the Full GOST Block cipher**
  - **Equivalent-key technique for enhancing the attack**
- **Conclusion**

Background

GOST Block Cipher

- Soviet Encryption Standard "GOST 28147-89".
- Standardized in 1989 as the Russian Encryption Standard. (*Russian DES*).

Implementation Aspect

- Recently, Poschmann et.al. show the 650-GE H/W implementation. @CHES 2010
- Considered as Ultra light weight Block cipher such as KATAN family and PRESENT.
- 650 GE implementation supports only hard-wired fixed key (single key model).

Cryptanalysis

Key Setting	Type of Attack	Round	Complexity	Data	Paper
Single Key	Differential	13	-	2^{51} (CP)	[28]
	Slide	24	2^{63}	$2^{64} - 2^{18}$ (KP)	[2]
	Slide	30	2^{254}	$2^{64} - 2^{18}$ (KP)	[2]
	Reflection	30	2^{224}	2^{32} (KP)	[17]

Single Key (Weak key)	Slide (2^{128} weak keys)	32 (full)	2^{63}	2^{63} (ACP)	[2]
	Reflectction (2^{224} weak keys)	32 (full)	2^{192}	2^{32} (CP)	[17]

Related Key	Differential	21	Not given	2^{56} (CP)	[28]
	Differential	32 (full)	2^{224}	2^{35} (CP)	[19]
	Boomerang	32 (full)	2^{248}	$2^{7.5}$ (CP)	[15]

In spite of considerable efforts, there is no key recovery attack on full GOST in the single (fixed) key model without weak keys.

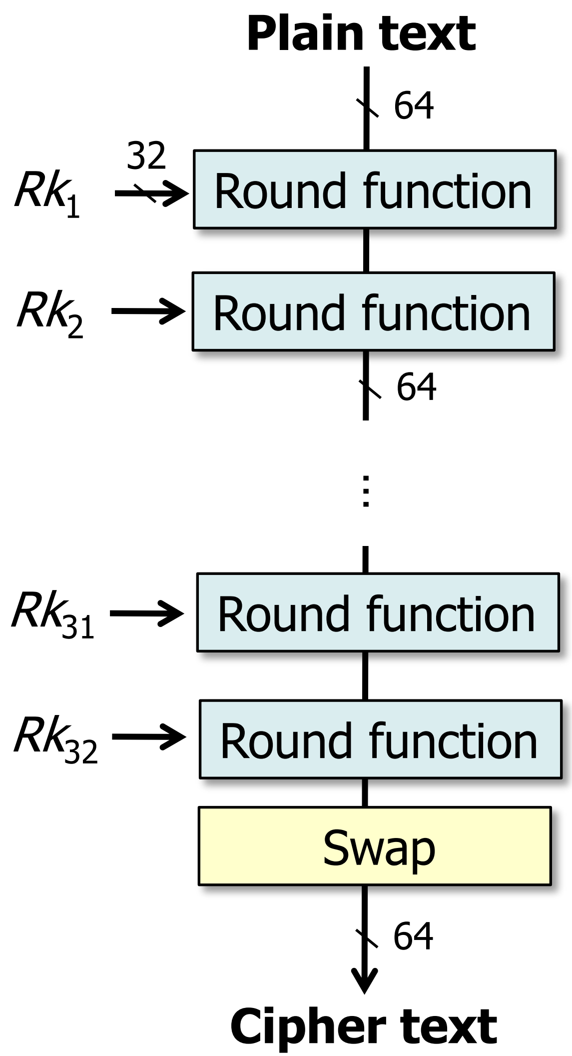
Cryptanalysis

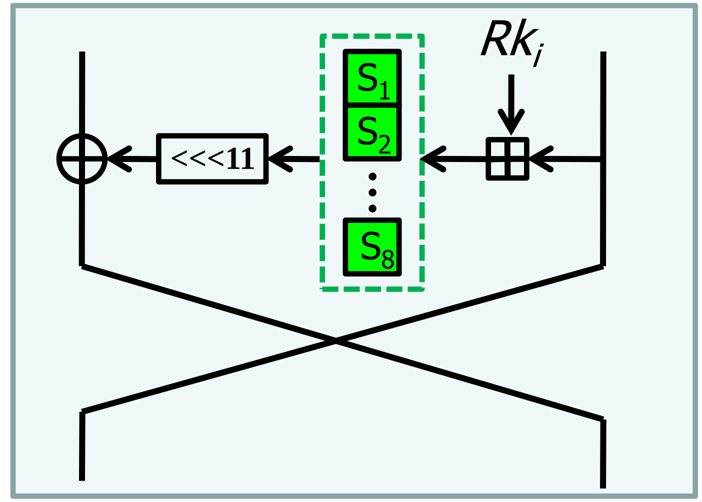
Key Setting	Type of Attack	Round	Complexity	Data	Paper
Single Key	Differential	13	-	2^{51} (CP)	[28]
	Slide	24	2^{63}	$2^{64}-2^{18}$ (KP)	[2]
	Slide	30	2^{254}	$2^{64}-2^{18}$ (KP)	[2]
	Reflection	30	2^{224}	2^{32} (KP)	[17]
	Reflection-MITM	**32 (full)**	**2^{225}**	**2^{32} (KP)**	**Ours**
Single Key (Weak key)	Slide (2^{128} weak keys)	32 (full)	2^{63}	2^{63} (ACP)	[2]
	Reflectction (2^{224} weak keys)	32 (full)	2^{192}	2^{32} (CP)	[17]
Related Key	Differential	21	Not given	2^{56} (CP)	[28]
	Differential	32 (full)	2^{224}	2^{35} (CP)	[19]
	Boomerang	32 (full)	2^{248}	$2^{7.5}$ (CP)	[15]

A first single-key attack on the full GOST block cipher. (work for all key classes)

Structure of GOST

- **32-round Feistel Structure with 64-bit block and 256-bit key**





Key schedule

$$Master\ key = K_1 || K_2 || \ldots || K_8$$

256 bit 32 bit × 8

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Key	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8

Round	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Key	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_8	k_7	k_6	k_5	k_4	k_3	k_2	k_1

Known Techniques

- 3 subset MITM attack
- Reflection attack

3-Subset Meet-in-the-Middle Attack

[General]

- Proposed by Bogdanov and Rechberger @SAC2010.
- Applied to KTANTAN-32/48/64.

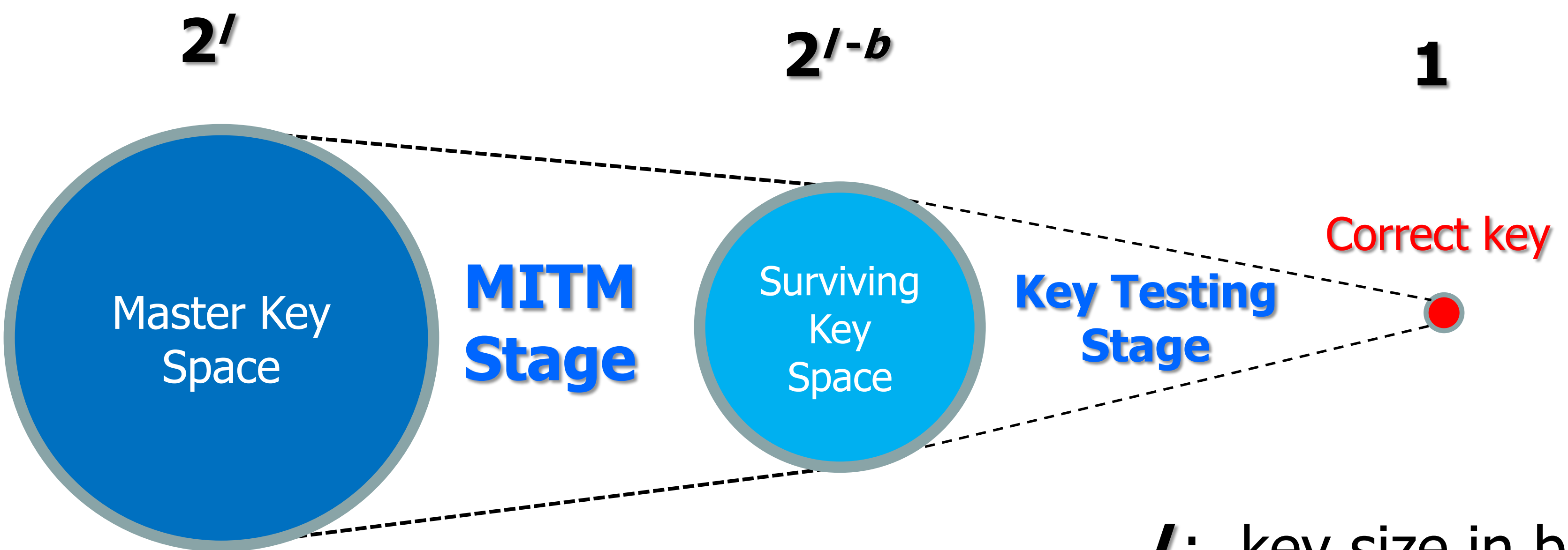
[Technical aspect]

- Construct 3-subsets of key bits to mount the MITM approach.
- Based on recent techniques of preimage attacks of hash functions.

3-Subset Meet-in-the-Middle Attack

- Consists of two stages : MITM stage ⇒ Key testing stage

@ MITM stage : Filter out part of wrong keys by using MITM techniques
@ Key testing stage : Find the correct key in the brute force manner.

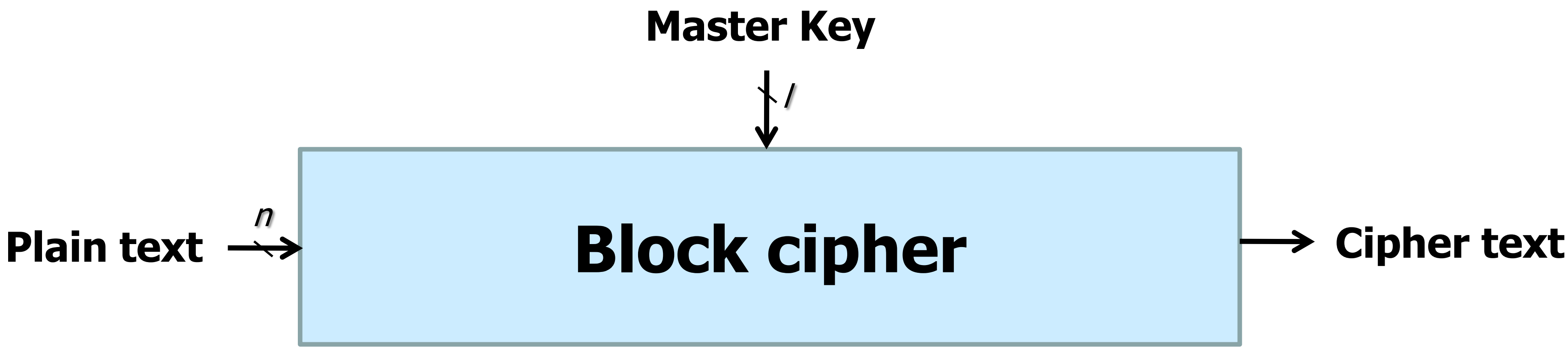


l : key size in bit

b: block size in bit

MITM Stage

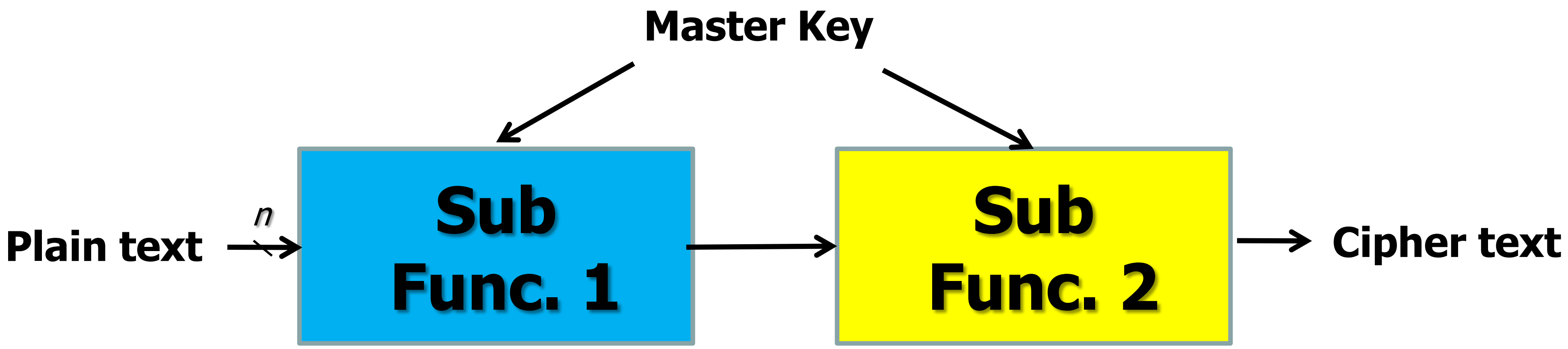
- Divide the Block cipher into 2 sub functions.



Block cipher : *l* bit master key and *n* bit block size

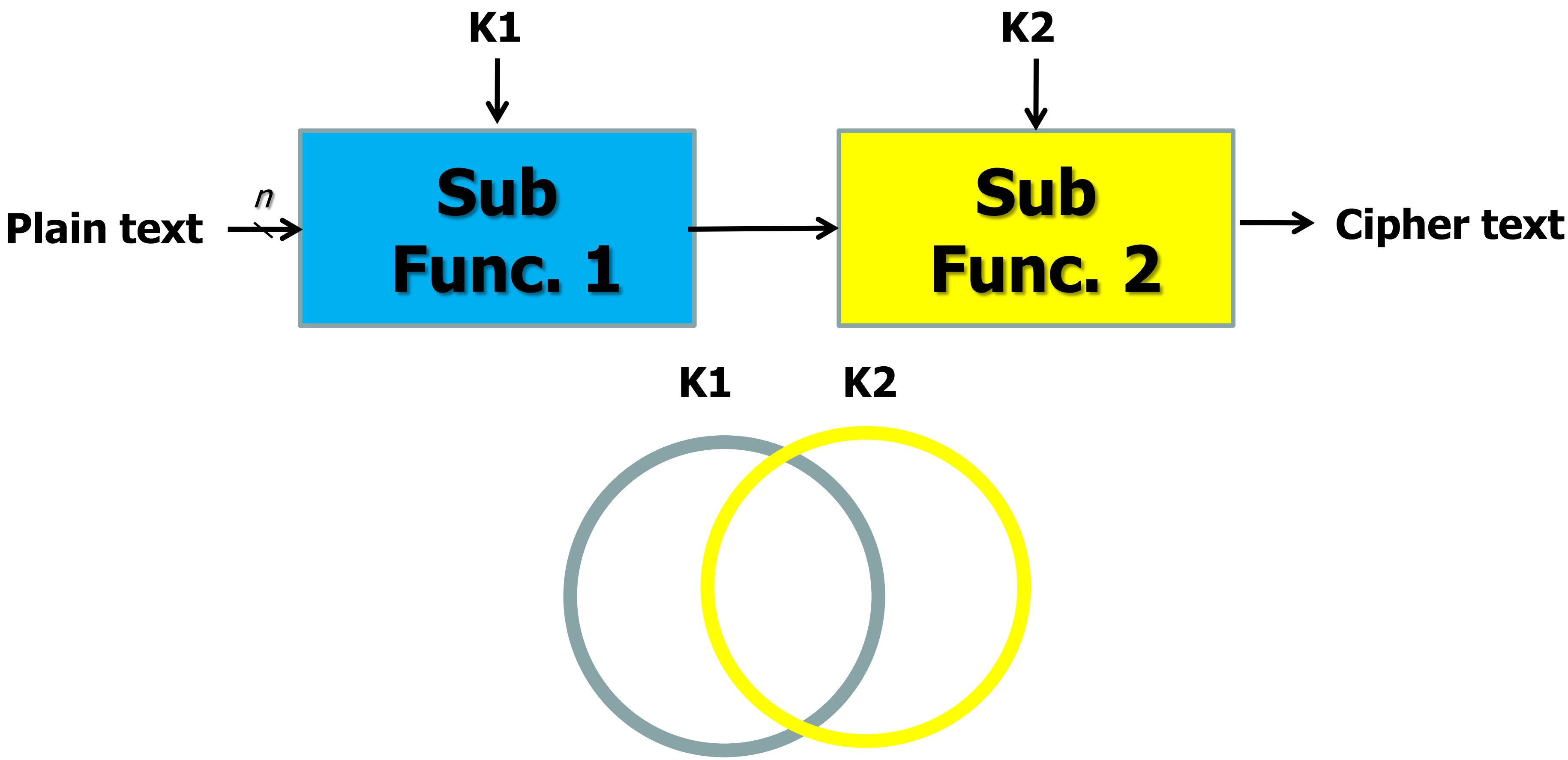
MITM Stage

- Divide the Block cipher into 2 sub functions



MITM Stage

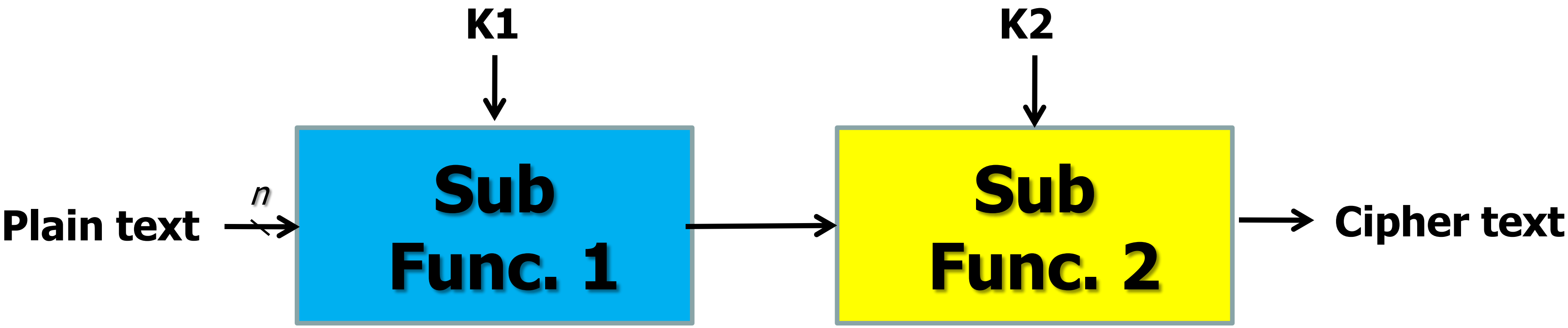
- Divide the Block cipher into 2 sub functions

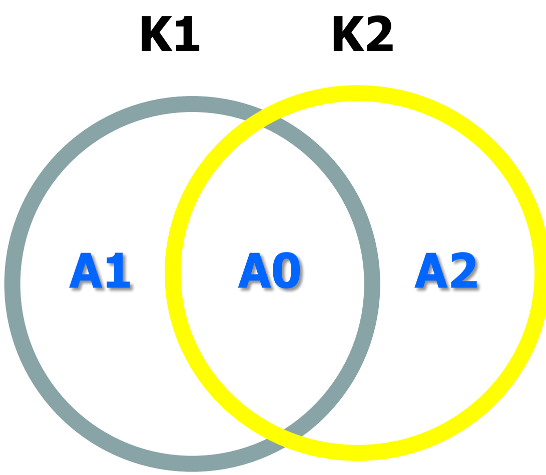


K1: sub set of key bits used in Sub Func. 1.
K2: sub set of key bits used in Sub Func. 2.

MITM Stage

- Construct 3-subset of master key, **A0, A1, A2**



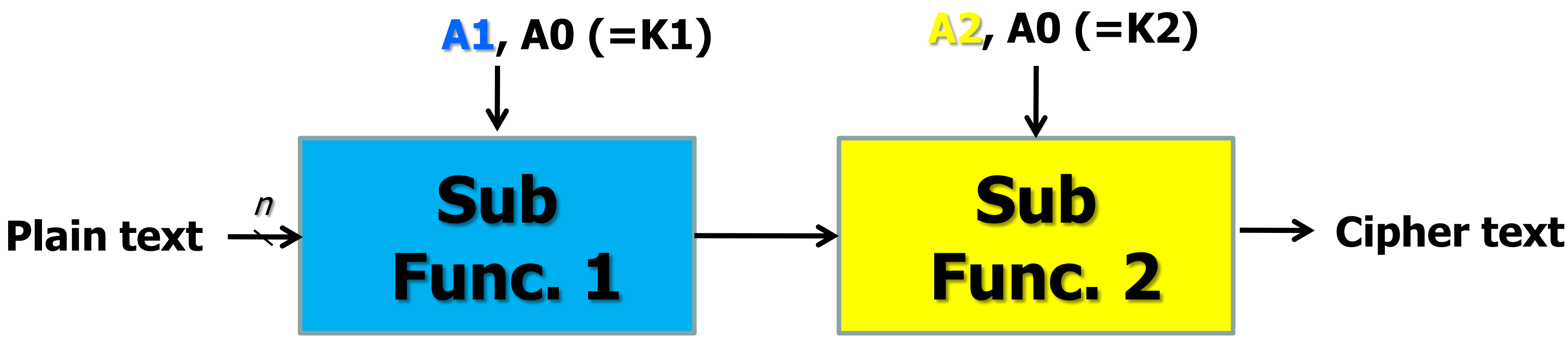


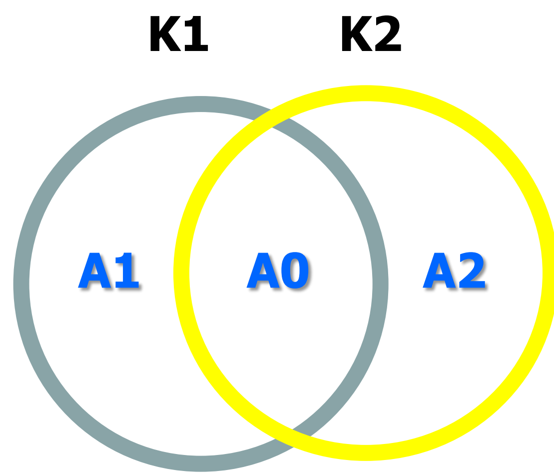
A0 = K1∩K2
A1 = K1/(K1∩K2)
A2 = K2/(K1∩K2)

K1: sub set of key bits used in Sub Func. 1.
K2: sub set of key bits used in Sub Func. 2.

MITM Stage

- Construct 3-subset of master key, **A0, A1, A2**



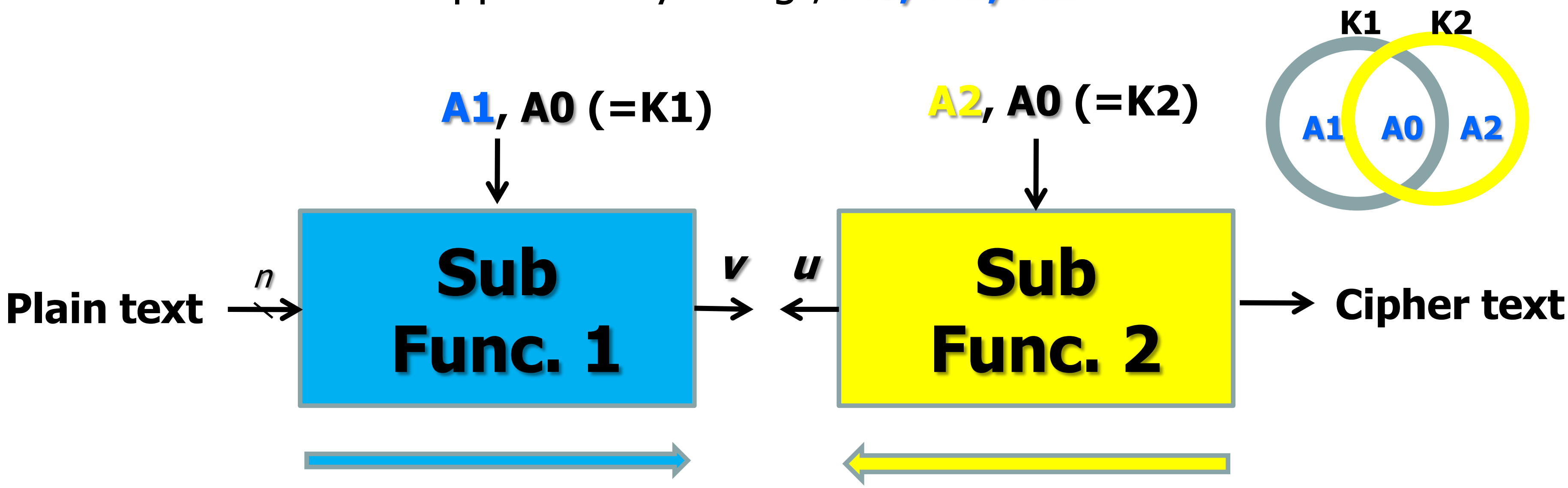


A0 = K1∩K2
A1 = K1/(K1∩K2)
A2 = K2/(K1∩K2)

K1: sub set of key bits used in Sub Func. 1.
K2: sub set of key bits used in Sub Func. 2.

MITM Stage

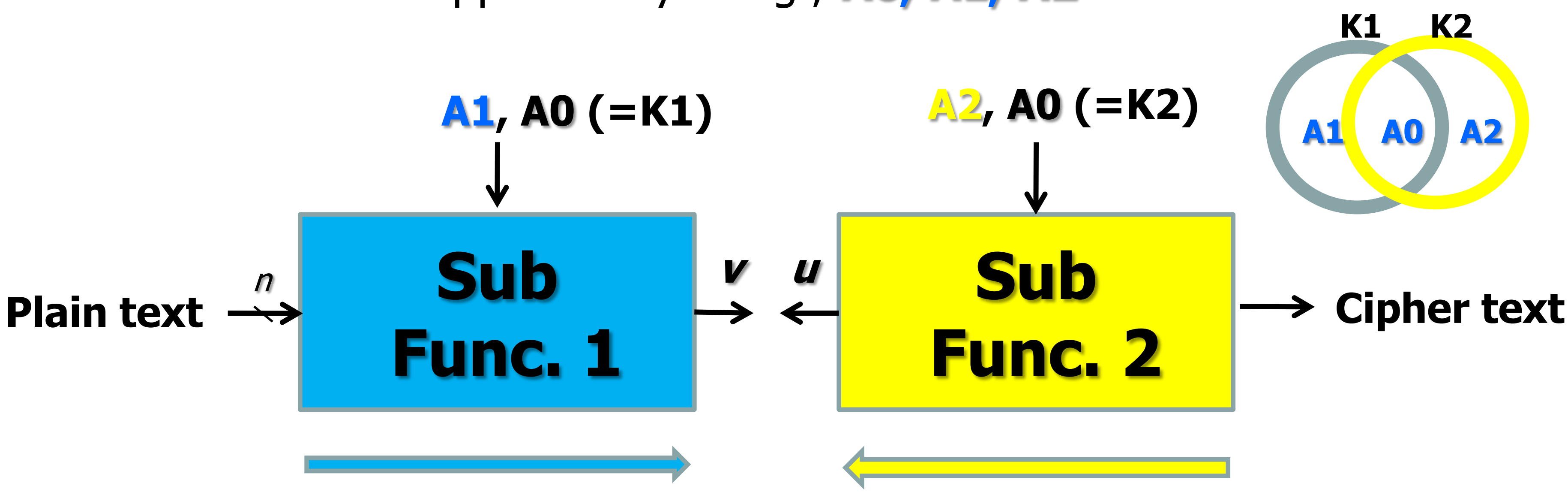
- Mount the MITM approach by using , **A0, A1, A2**



1. Guess the value of **A0**
2. Compute *v* for all value of **A1** and make a table (**A1**, *v*) pairs
3. Compute *u* for all value of **A2**
4. If *v* = *u*, then regard (**A0, A1, A2**) as key candidates
5. Repeat 2-4 with all value of **A0** ($2^{|A0|}$ times)

MITM Stage

- Mount the MITM approach by using , **A0, A1, A2**



1. Guess the value of **A0**
2. Compute v for all value of **A1** and make a table (**A1**, v) pairs
3. Compute u for all value of **A2**
4. If $v = u$, then regard (**A0, A1, A2**) as key candidates
5. Repeat 2-4 with all value of **A0** ($2^{|A0|}$ times)

# of surviving key candidates :

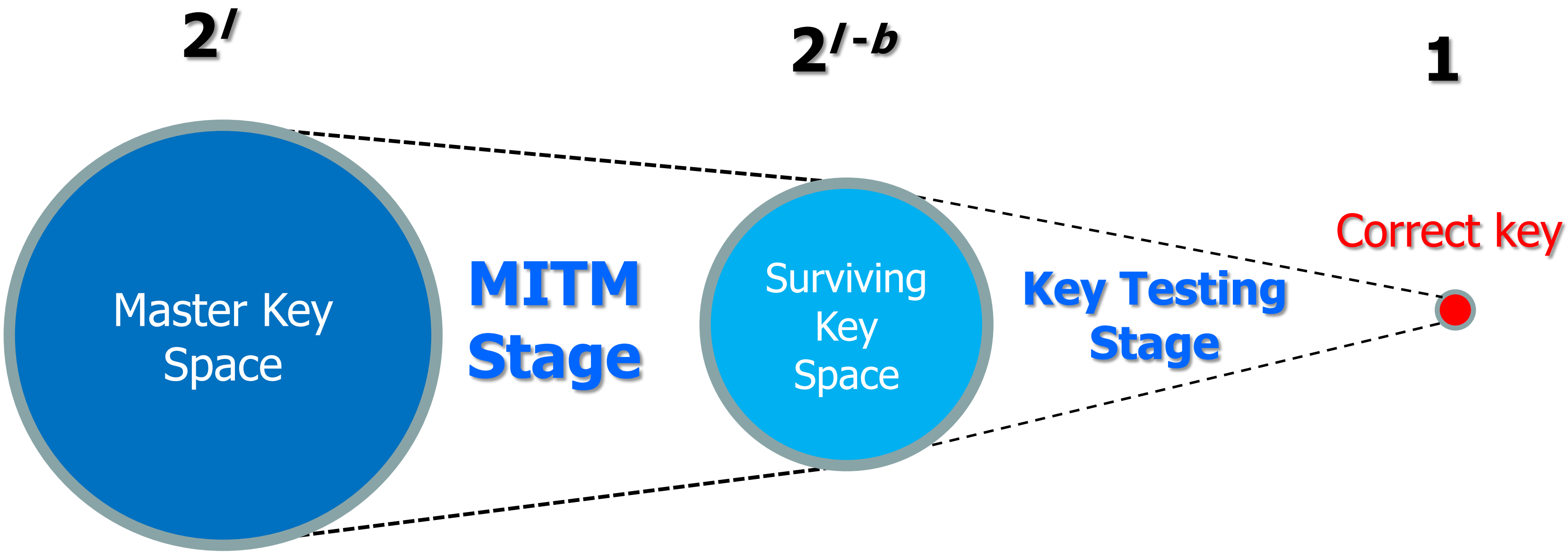
$$2^{|A1|+|A2|} / 2^{b} \times 2^{|A0|} = 2^{l-b}$$

l: key size in bit

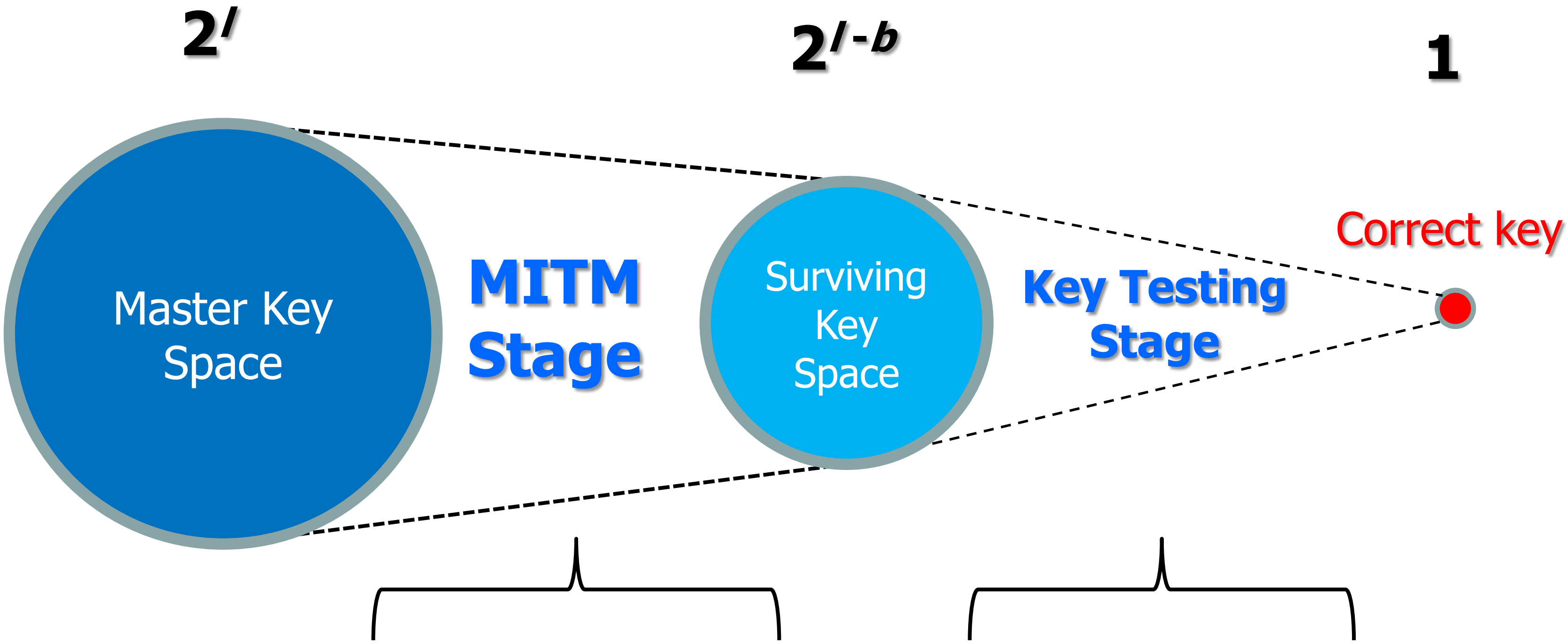
b: block size in bit

Key Testing Stage

- Test surviving keys in brute force manner by using additional data.



Evaluation



Complexity $= 2^{|A0|}(2^{|A1|}+2^{|A2|}) + (2^{l-b}+2^{l-2b}+\ldots)$

Data $= \max(1, l/b)$

- Condition for a successful attack : **$\min(2^{|A1|}, 2^{|A2|}) > 2$**

The Point of the attack :

Find independent sets of master key bit such as A1 and A2

Known Techniques

- 3 subset MITM attack
- Reflection attack

Reflection Attack

[General]

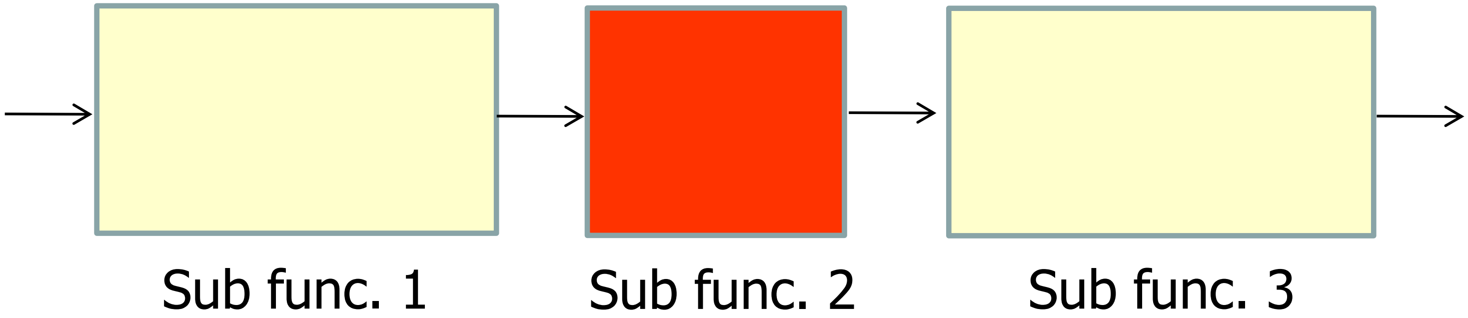
- Introduced by Kara and Manap @ FSE2007.
- Applied to Blowfish, GOST and more, so far.

[Technical Aspect]

- A technique for constructing fixed points .
- Utilize self-similarity of both encryption and decryption round functions.
  (Slide attack uses self-similarity of only encryption round functions)

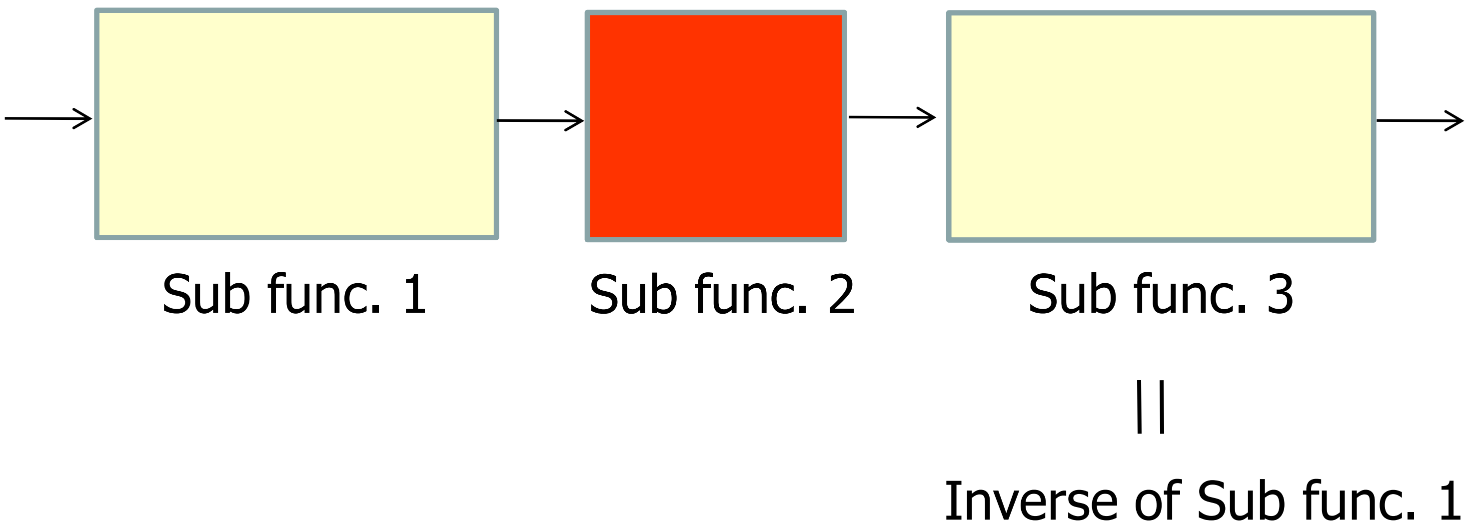
Reflection Attack

- Consider the 3 sub functions.



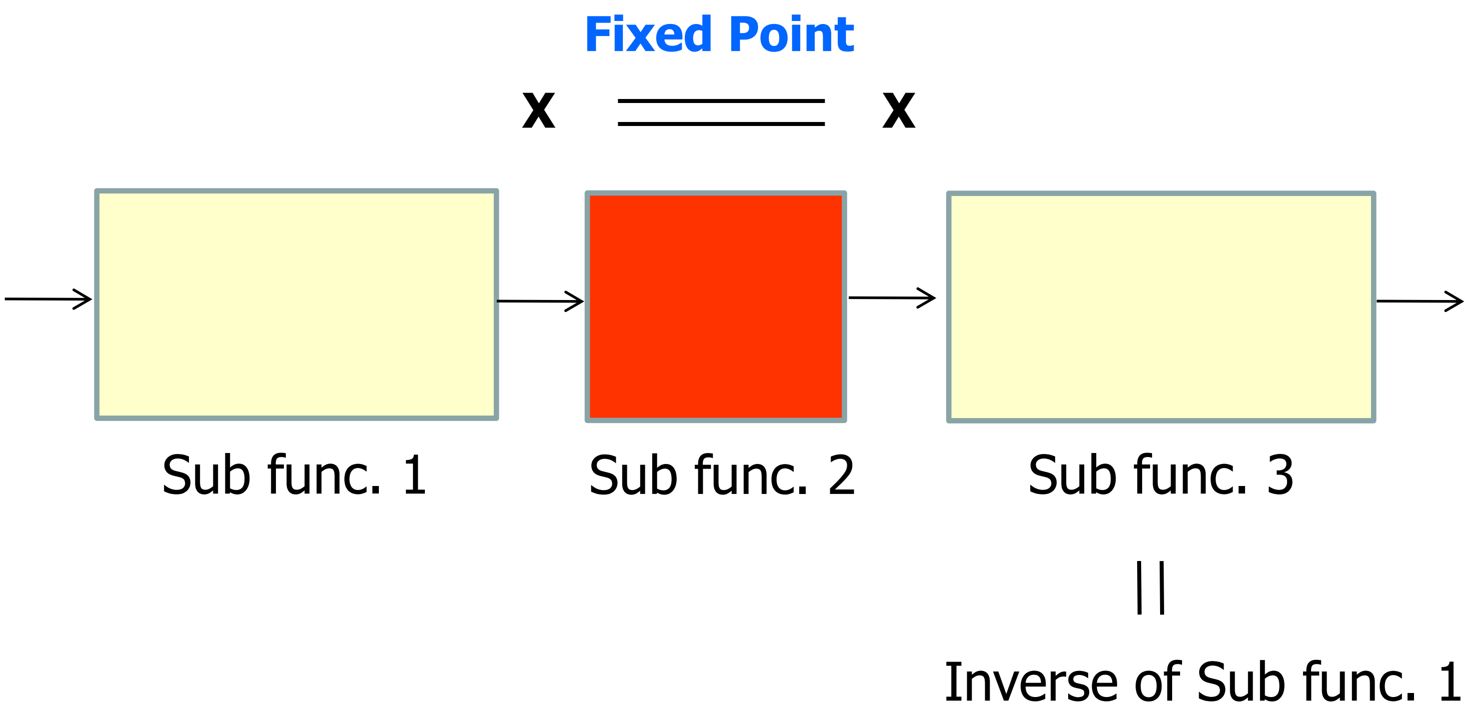
Reflection Attack

- Assume that the Sub func. 3 has involution property.
  i.e., Sub func. 3 is same as the inverse of Sub func. 1.

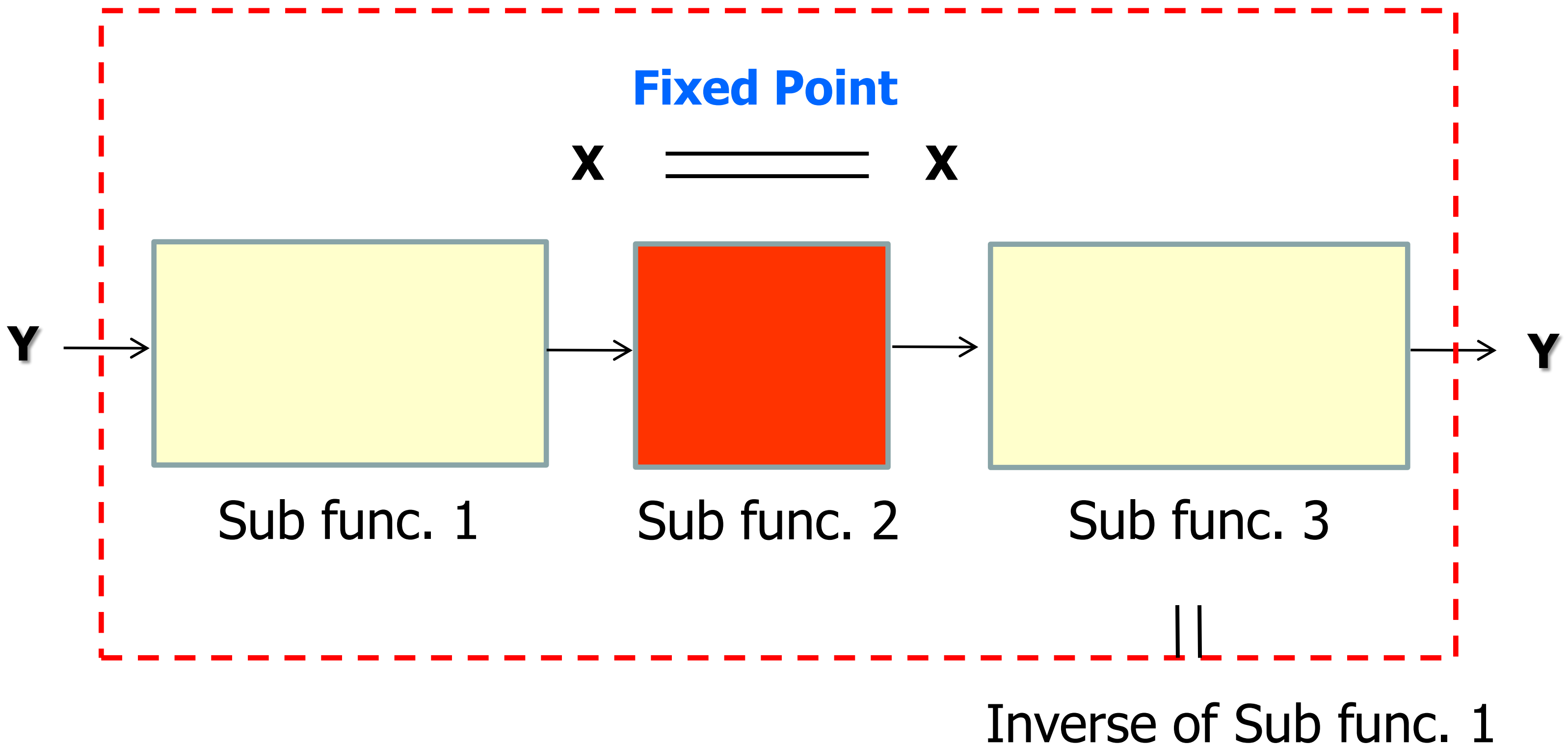


Reflection Attack

- If the Sub func. 2 has fixed points.



Reflection Attack

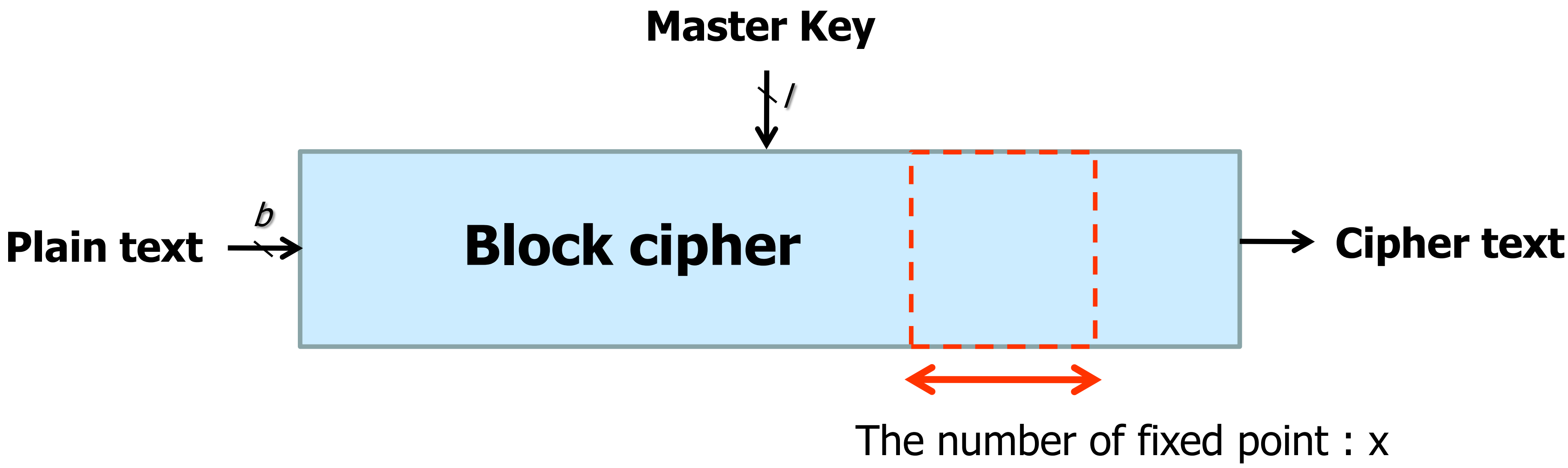


Local fixed Point of Sub func. 2 is expanded into previous and next rounds.

Reflection-MITM attack

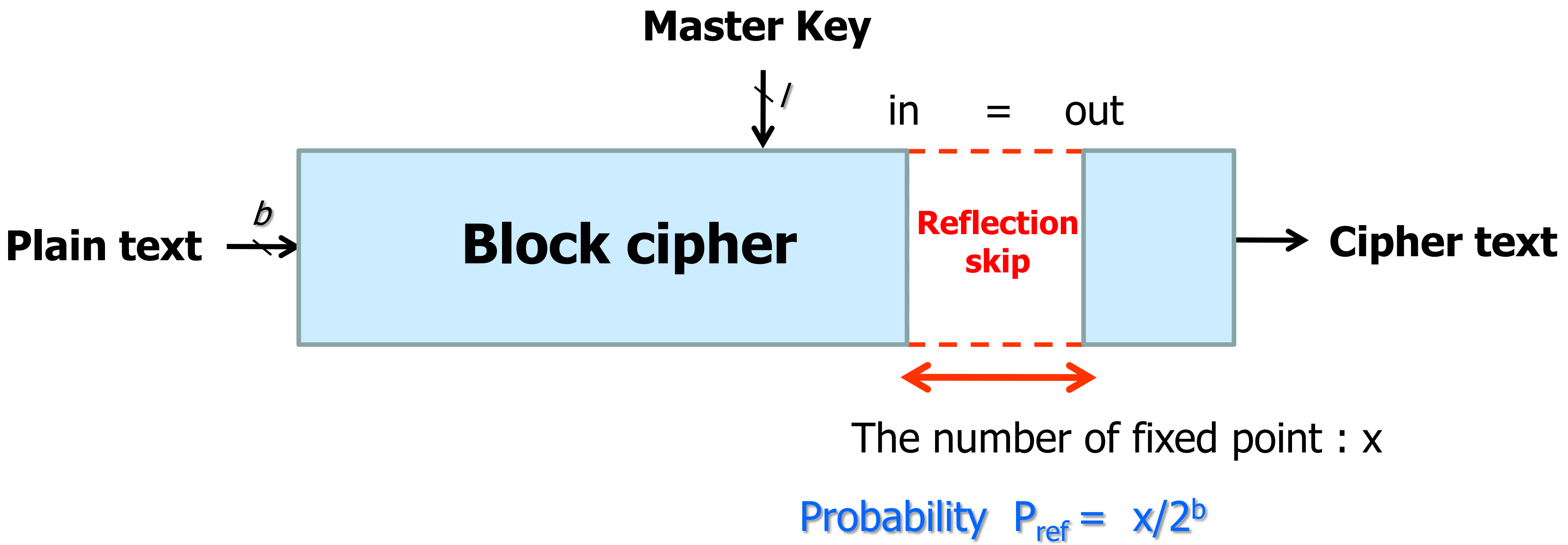
Core Idea of the R-MITM Attack

- Skip some round functions by using the fixed points of the Reflection attack



Core Idea of the R-MITM Attack

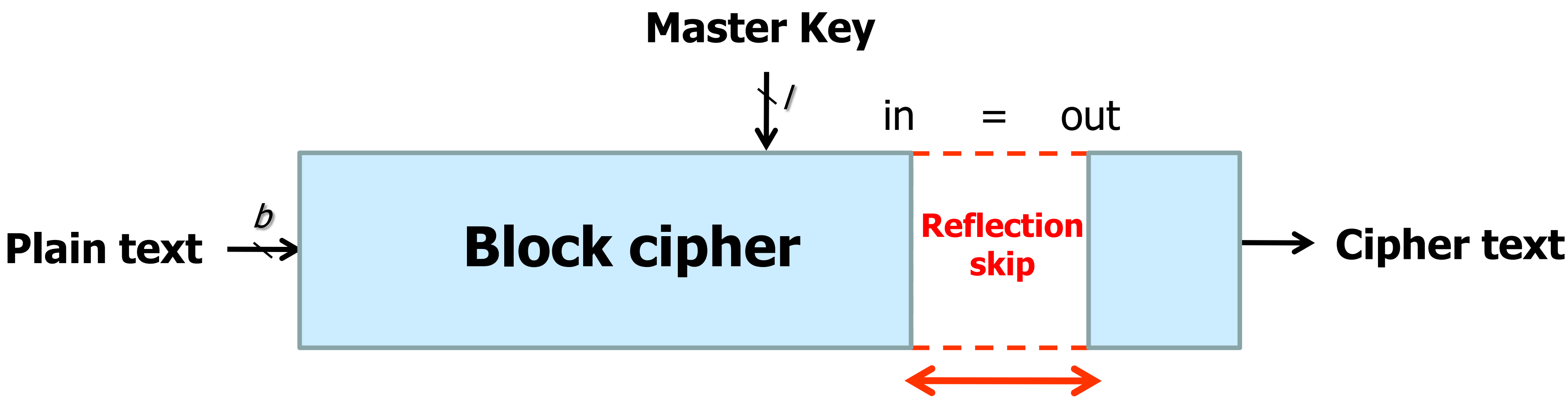
- Skip some round functions by using the fixed points of the Reflection attack



In one of P_{ref}^{-1} Plaintext/Ciphertext pairs,
the reflection skip occurs

Stages of the R-MITM Attack

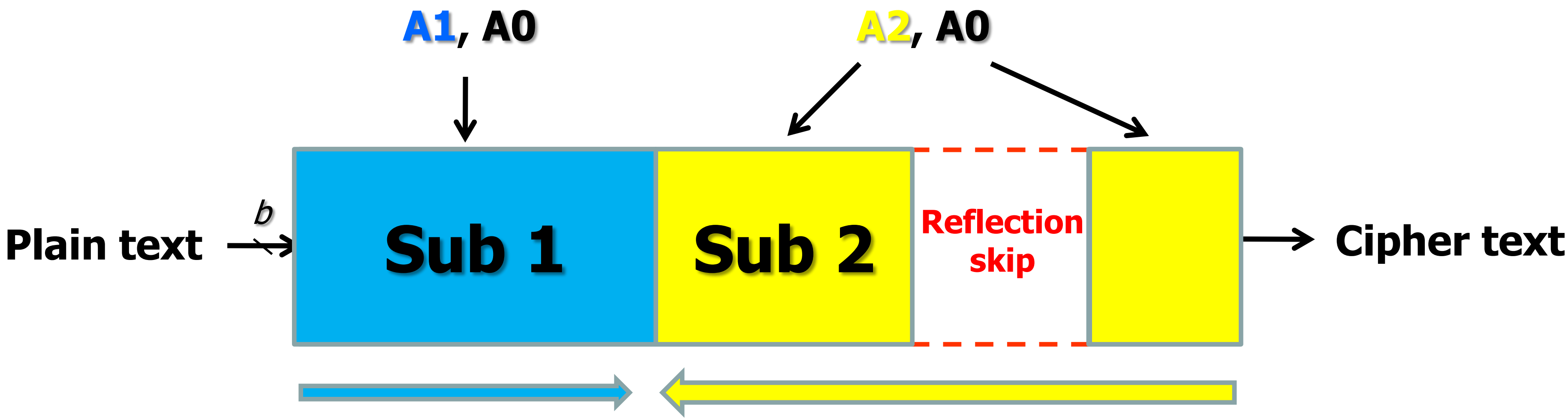
- **Data Collection stage**
  - Collect P_{ref}^{-1} Plaintext/Ciphertext pairs.
- **MITM stage and Key testing stage**
  - Mount all collected pair.
  - Assume that reflection skip occurs.



Stages of the R-MITM Attack

- **Data Collection stage**
  - Collect P_{ref}^{-1} Plaintext/Ciphertext pairs.
- **MITM stage and Key testing stage**

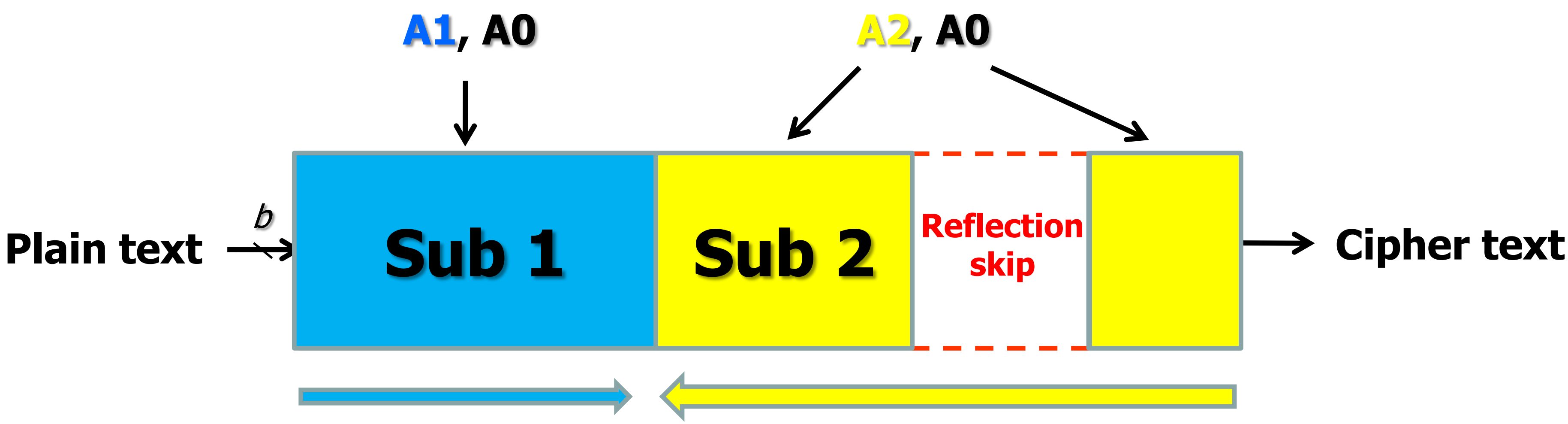
Assume that the reflection skip occurs in used P/C pair



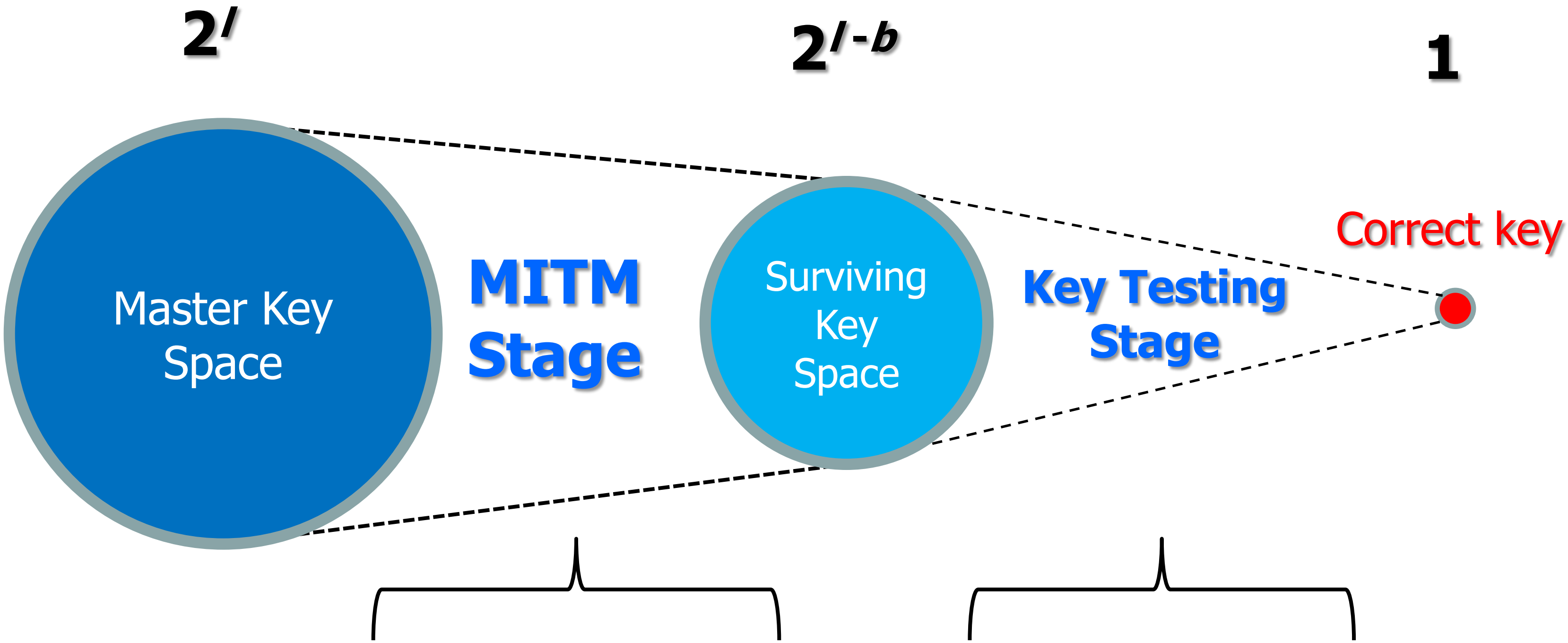
Stages of the R-MITM Attack

Advantage of R-MITM attack over 3-subset MITM attack

The key bits involved in skipped round can be disregarded!
=> it become easier to construct independent key sets.



Evaluation



Complexity $= \left(2^{|A0|}(2^{|A1|}+2^{|A2|}) + (2^{l-b}+2^{l-2b}+\ldots) \right) \times P_{ref}^{-1}$

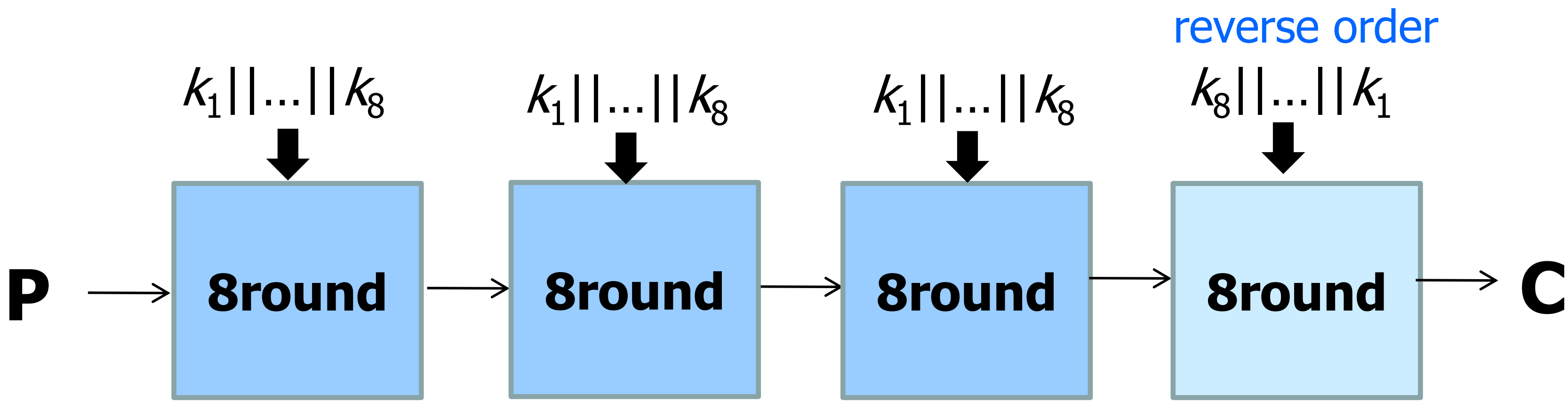
Data $= \max \left(1 , l/b , P_{ref}^{-1} \right)$

- Condition for a successful attack : **$\min (2^{|A1|}, 2^{|A2|}) > P_{ref}^{-1}$**

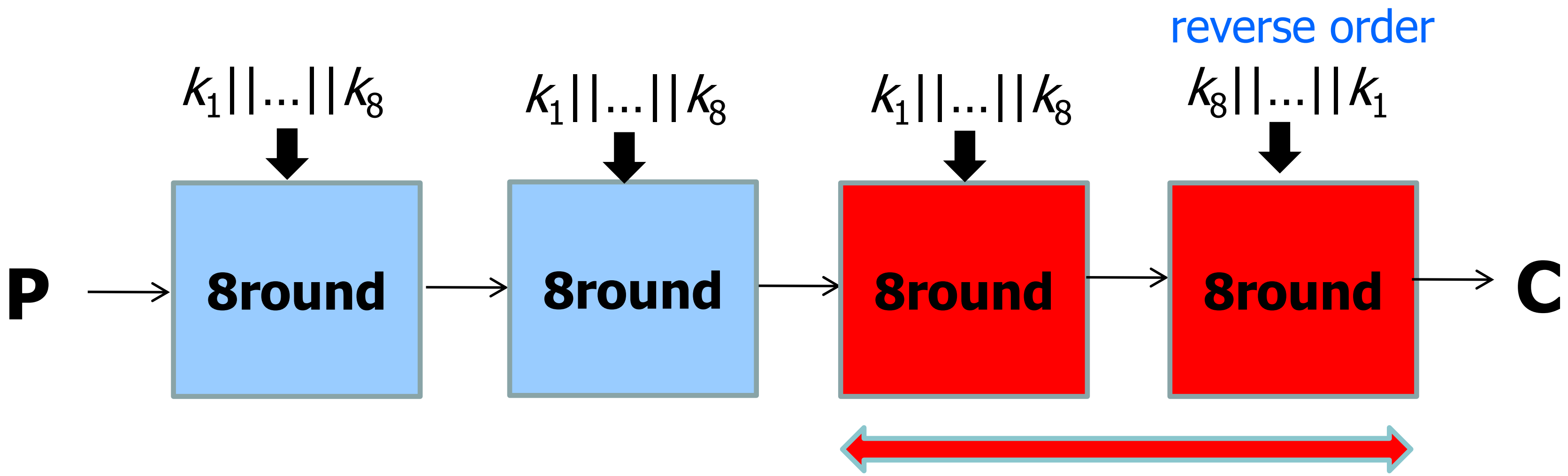
We need to construct large set of independent keys.

R-MITM Attack on the Full GOST

Application to Full GOST



Reflection Property

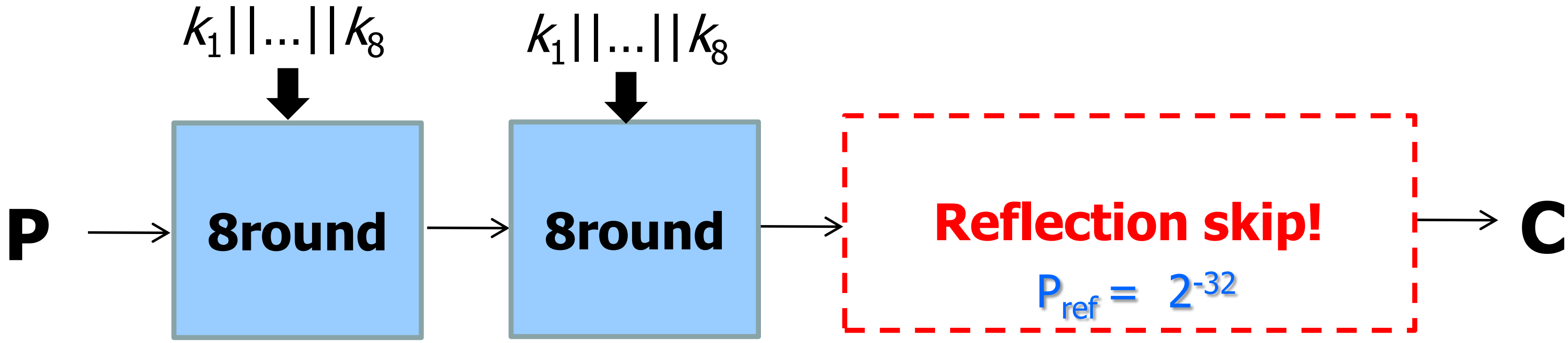


Reflection property was shown by Kara.

- # of fixed points is 2^{32}

- Probability $P_{ref} = 2^{-32}$ ($>> 2^{-64}$)

Reflection Skip

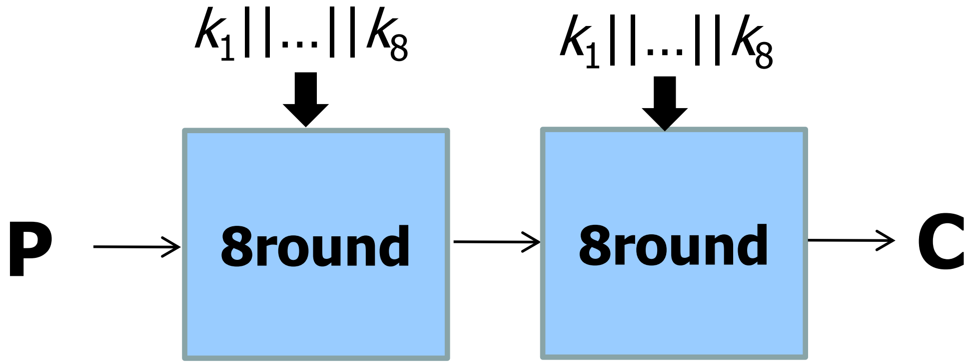


@Data collection stage:

Collect 2^{32} known plaintext/ciphertext pairs

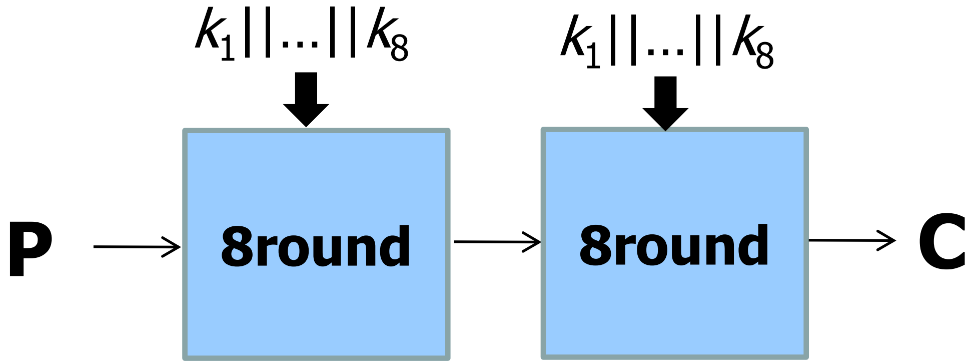
R-MITM Stage

- Assume that the reflection skip occurs ($P_{ref} = 2^{-32}$) for each pair.

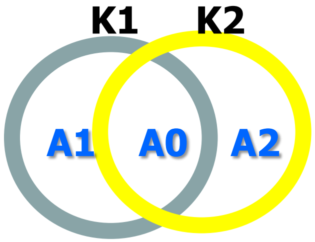


R-MITM Stage

- Assume that the reflection skip occurs ($P_{ref} = 2^{-32}$) for each pair.

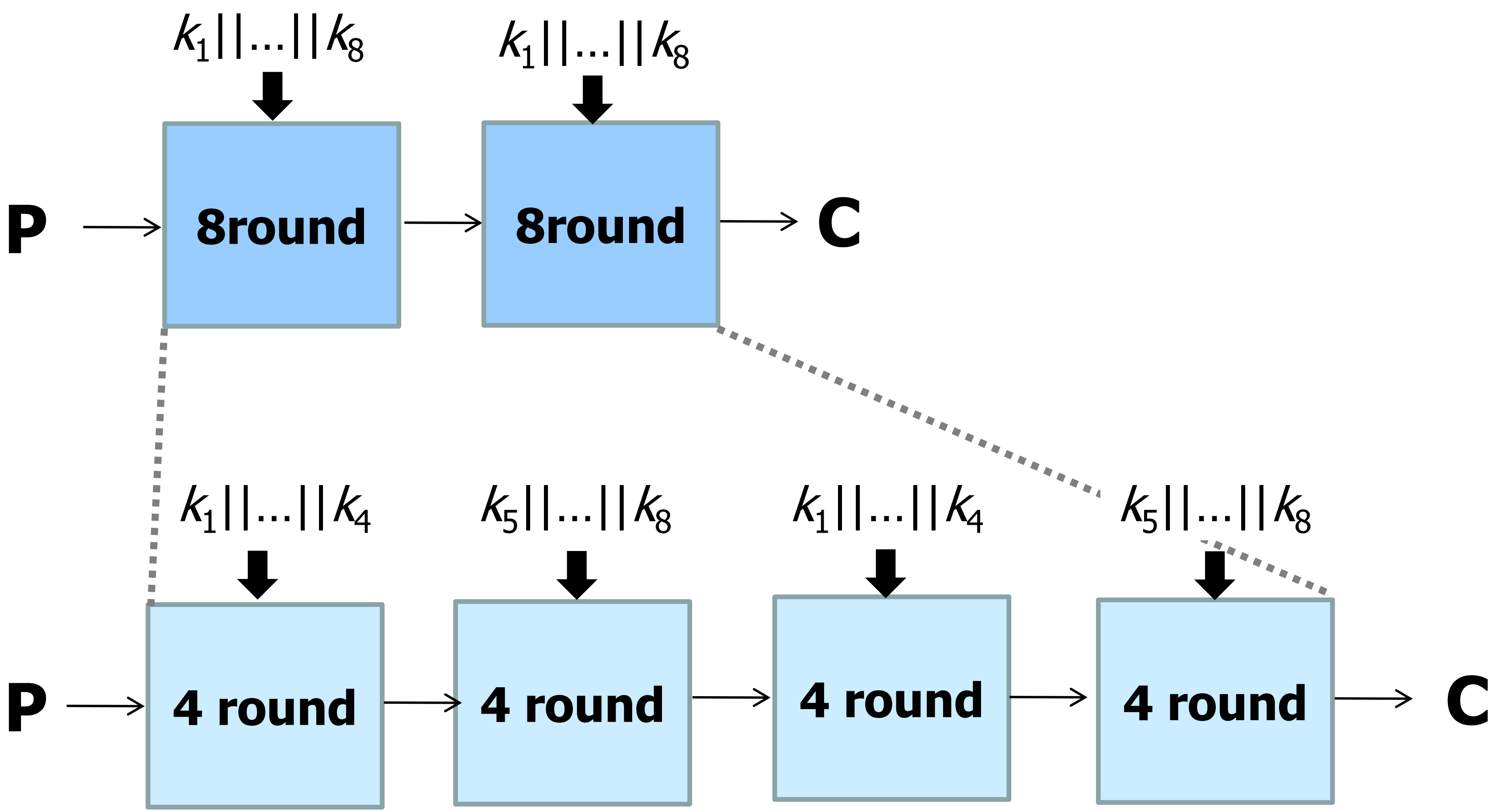


Condition for a successful attack :
$\min(2^{|A1|}, 2^{|A2|}) > 2^{32}$

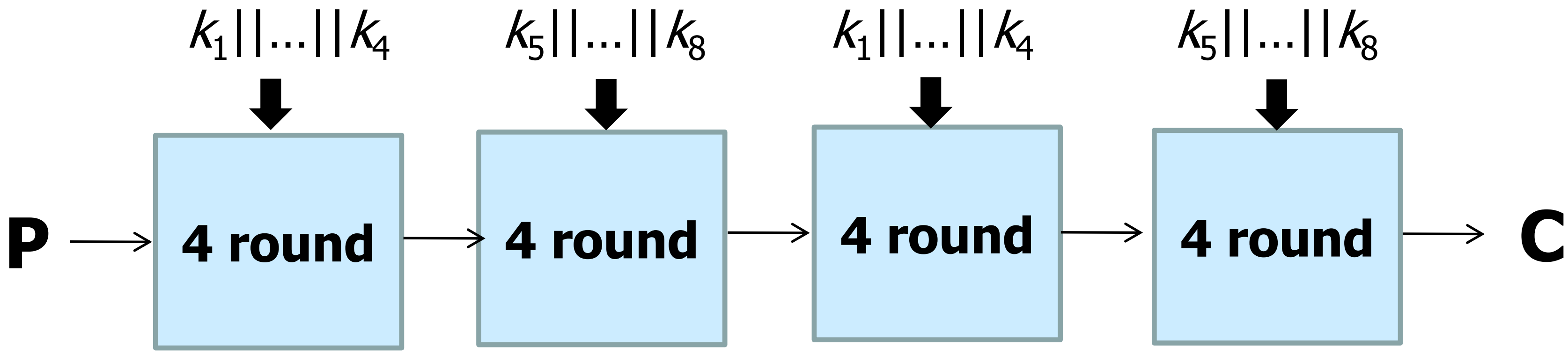


R-MITM Stage

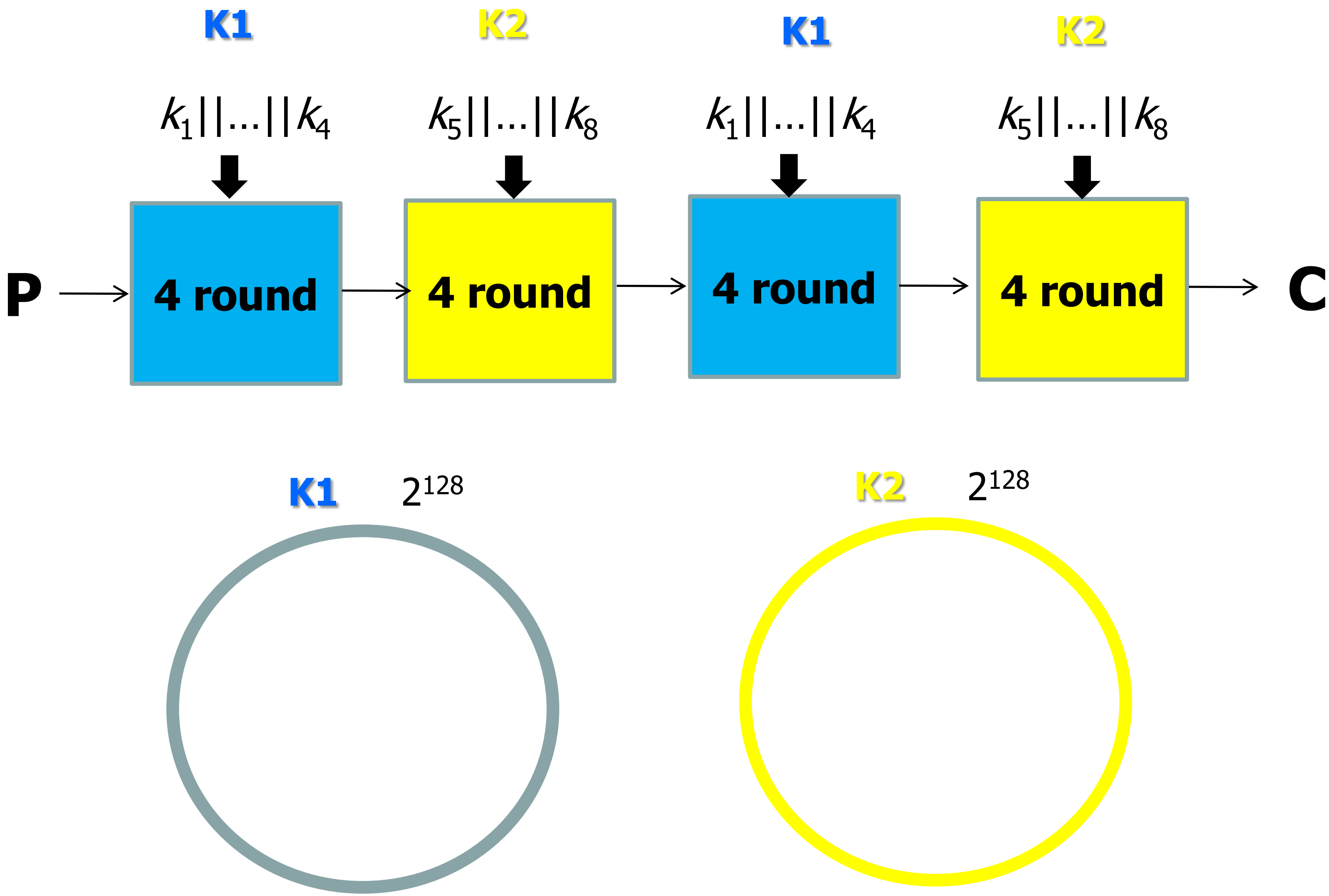
- Divide 4-round units.



MITM Stage

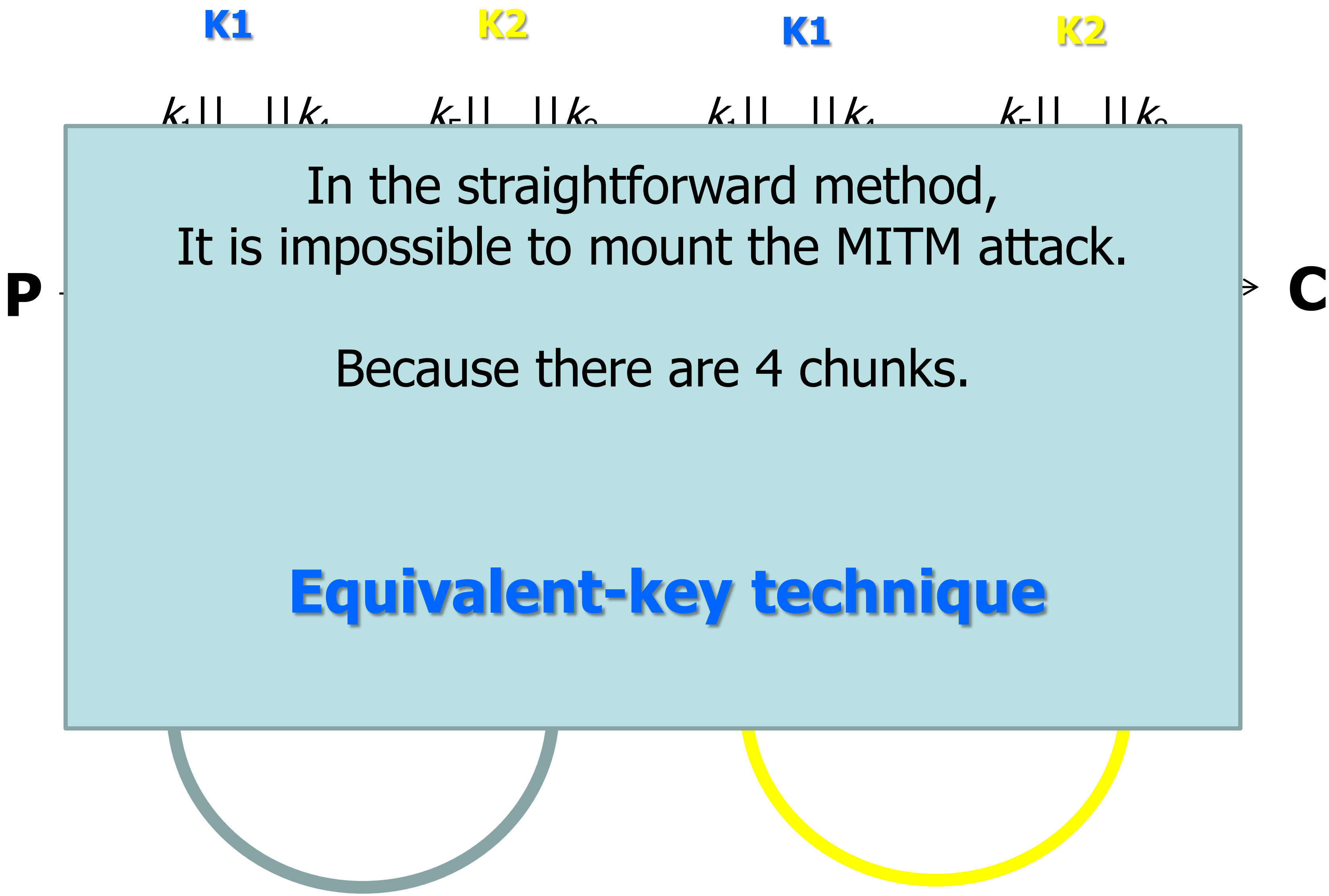


MITM Stage



K1
K2
K1
K2
$k_1||\ldots||k_4$
$k_5||\ldots||k_8$
$k_1||\ldots||k_4$
$k_5||\ldots||k_8$
P
4 round
4 round
4 round
4 round
C
K1
2^{128}
K2
2^{128}

MITM Stage



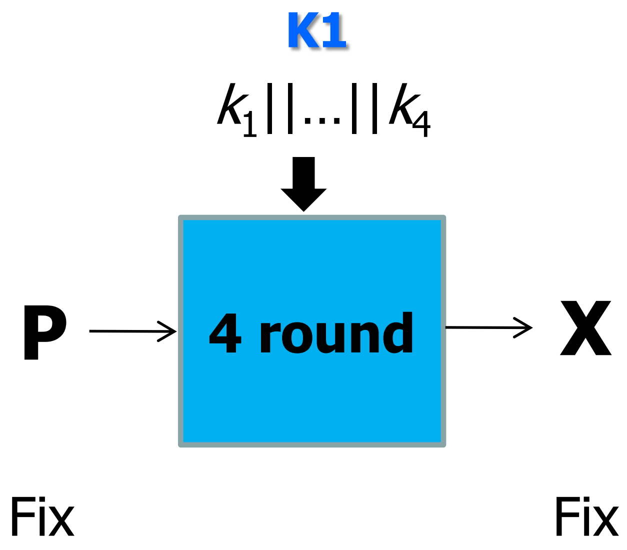
In the straightforward method,
It is impossible to mount the MITM attack.

Because there are 4 chunks.

Equivalent-key technique

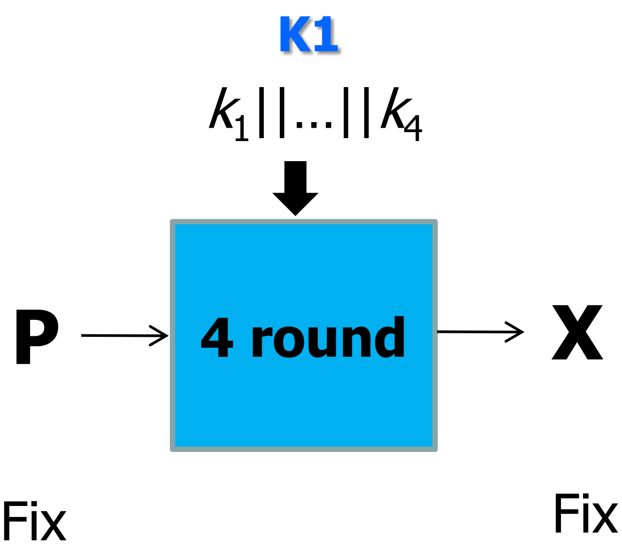
Equivalent Keys

- Define Equivalent keys used for our attack as
  "a set of keys that transforms P to X for 4-round unit"

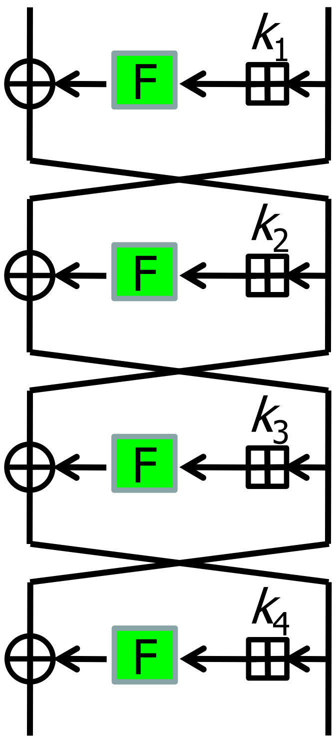


Equivalent Keys

- Define Equivalent keys used for our attack as
  "a set of keys that transforms P to X for 4-round unit"



Given the values of P and X,
It is easy to find such set.



k_1 : Guess
32 bit

k_2 : Guess
32 bit

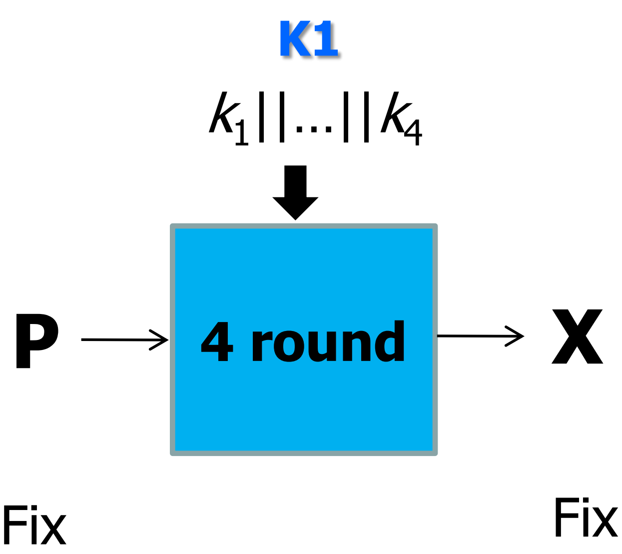
k_3 : Determine
32 bit

k_4 : Determine
32 bit

Equivalent Keys

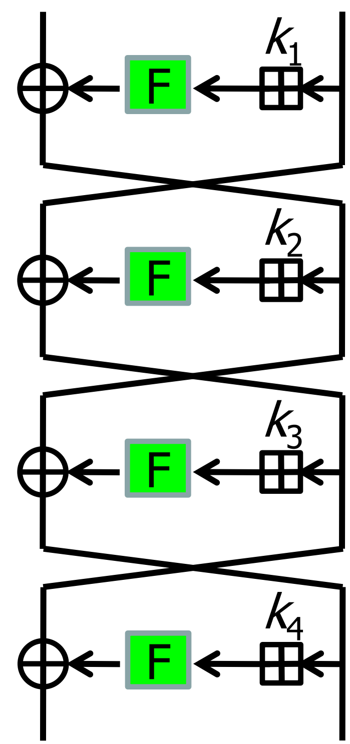
- Define Equivalent keys used for our attack as

"a set of keys that transforms P to X for 4-round unit"



For each (P, X) pair,
there are 2^{64} such equivalent keys

Given the values of P and X,
It is easy to find such set.



k_1 : Guess 32 bit

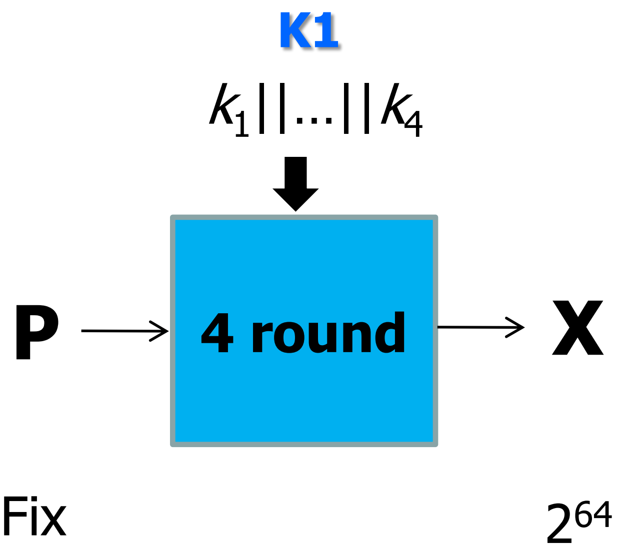
k_2 : Guess 32 bit

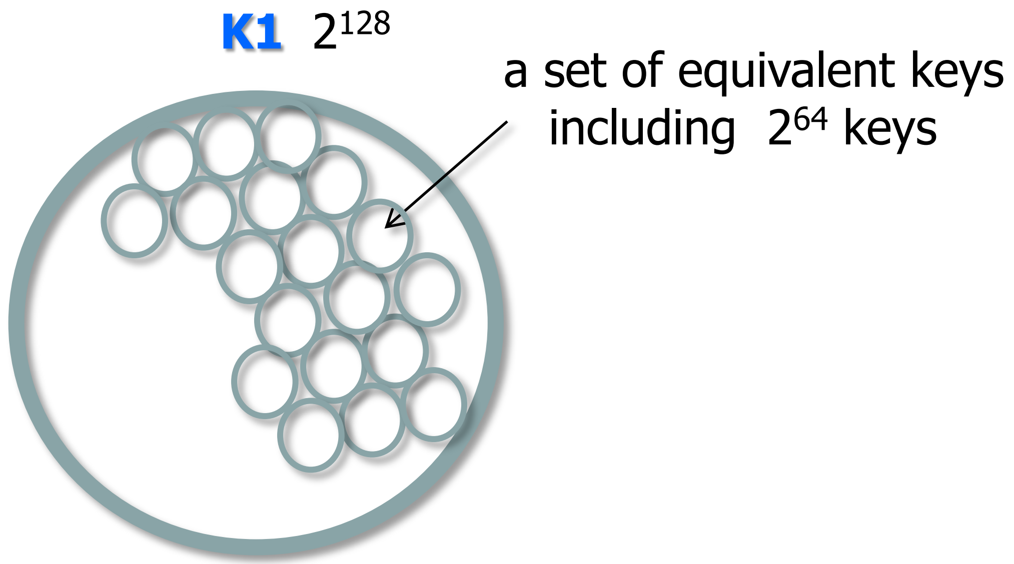
k_3 : Determine 32 bit

k_4 : Determine 32 bit

Equivalent Keys

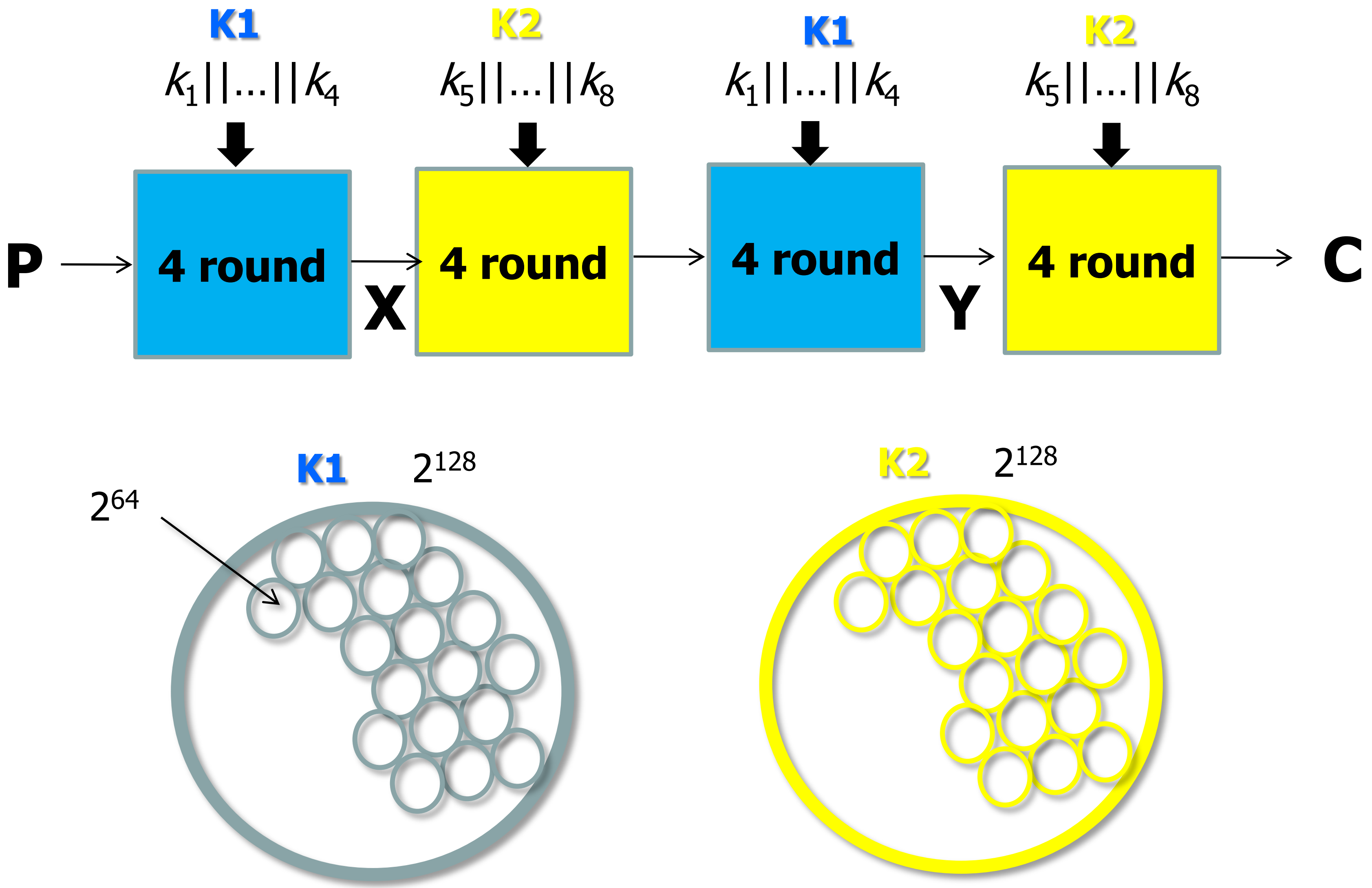
- Categorize K1 into sets of equivalent keys depending on values of X





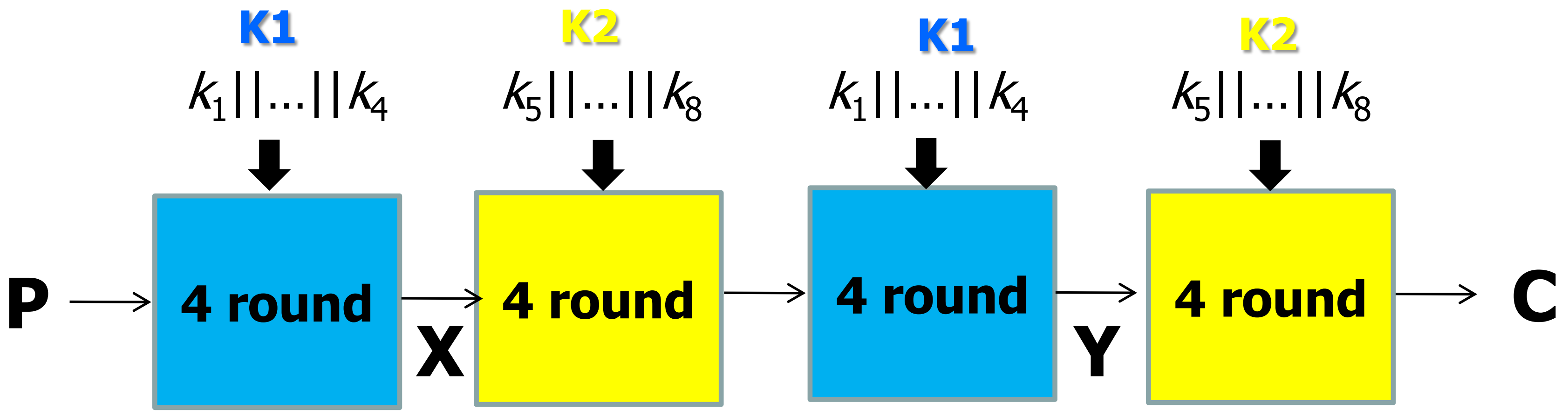
2^{64} sets of 2^{64} keys
(cover 2^{128} key space)

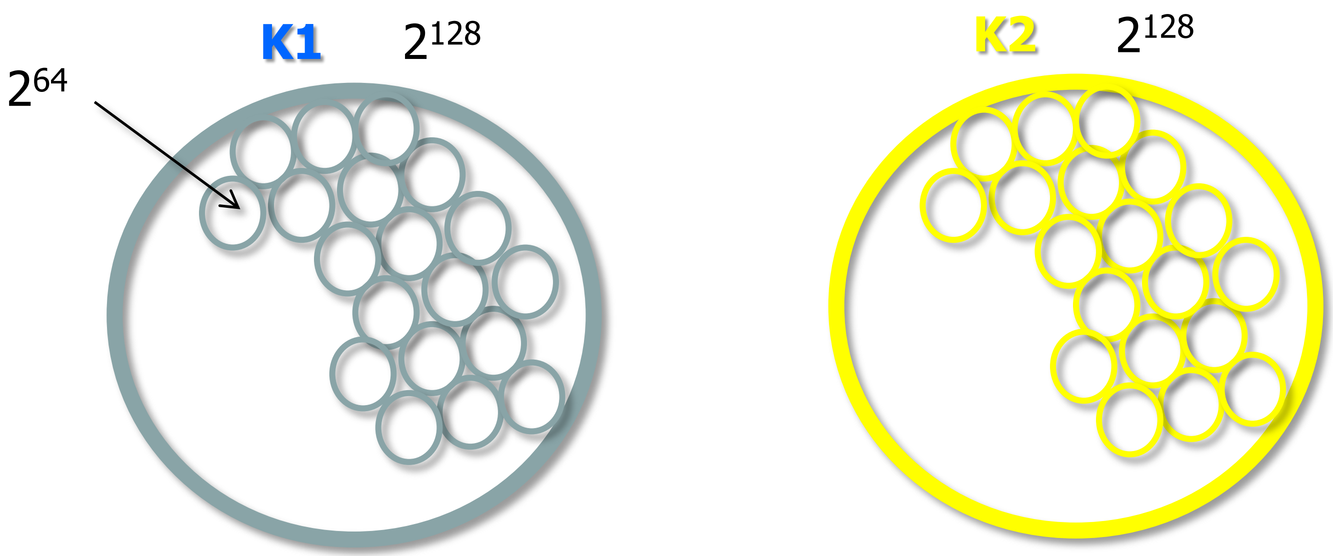
Equivalent Keys



Effective MITM approach

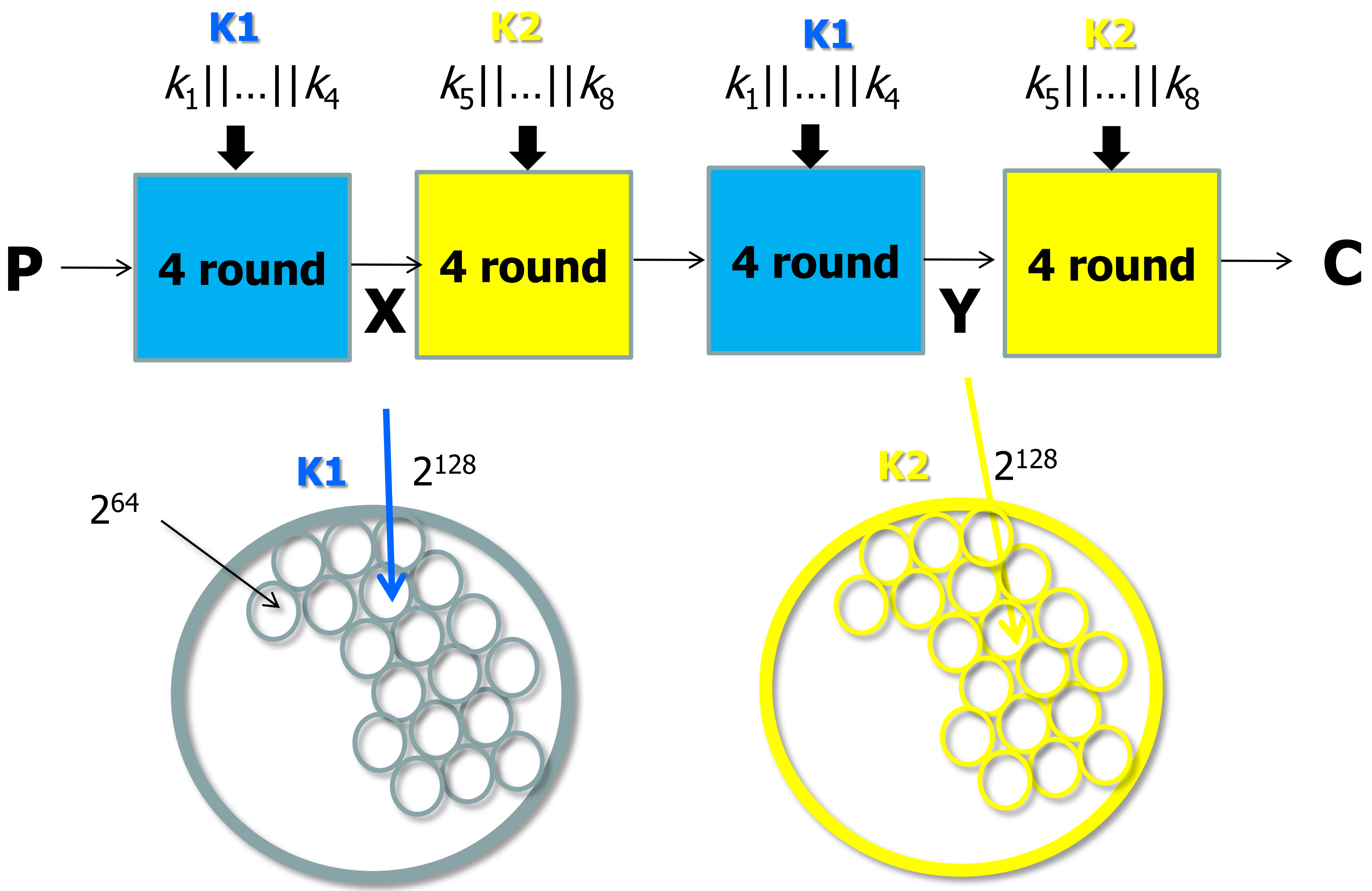
- **Guess values of X and Y.**





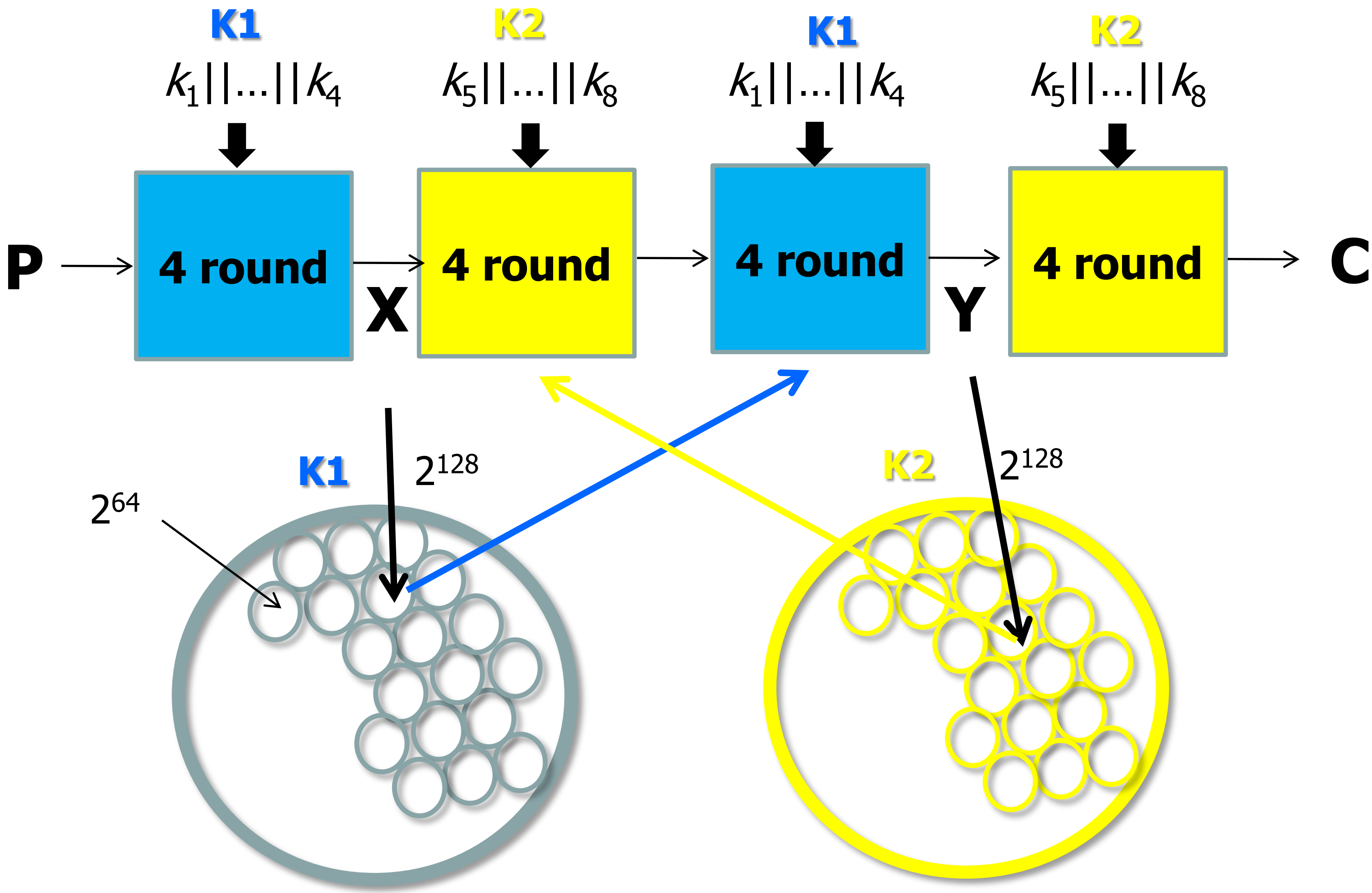
Effective MITM approach

- **Choose two set from K1 and K2, which transform X and Y**



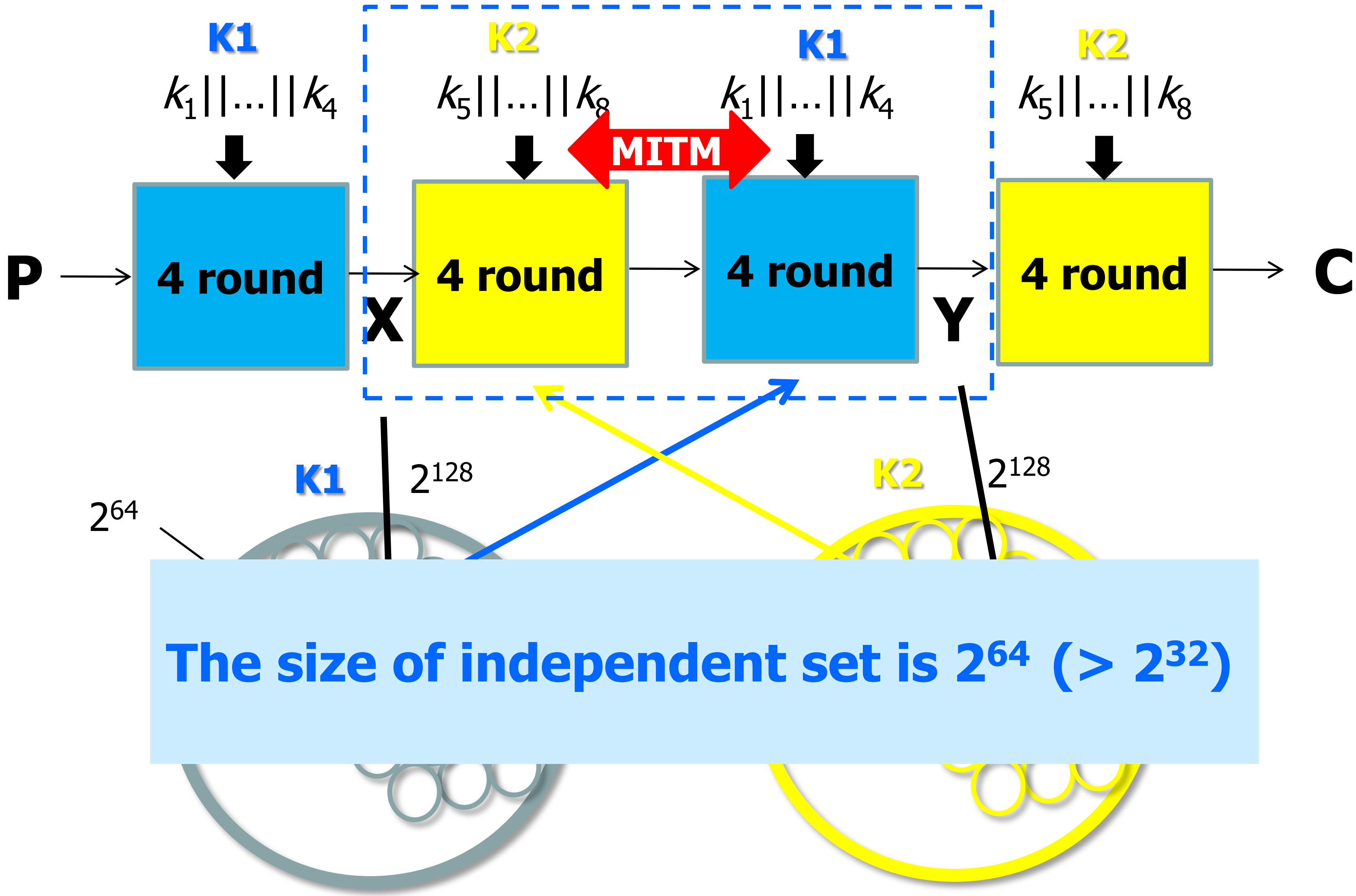
Effective MITM approach

- Mount the MITM approach in only intermediate 8 round.



Effective MITM approach

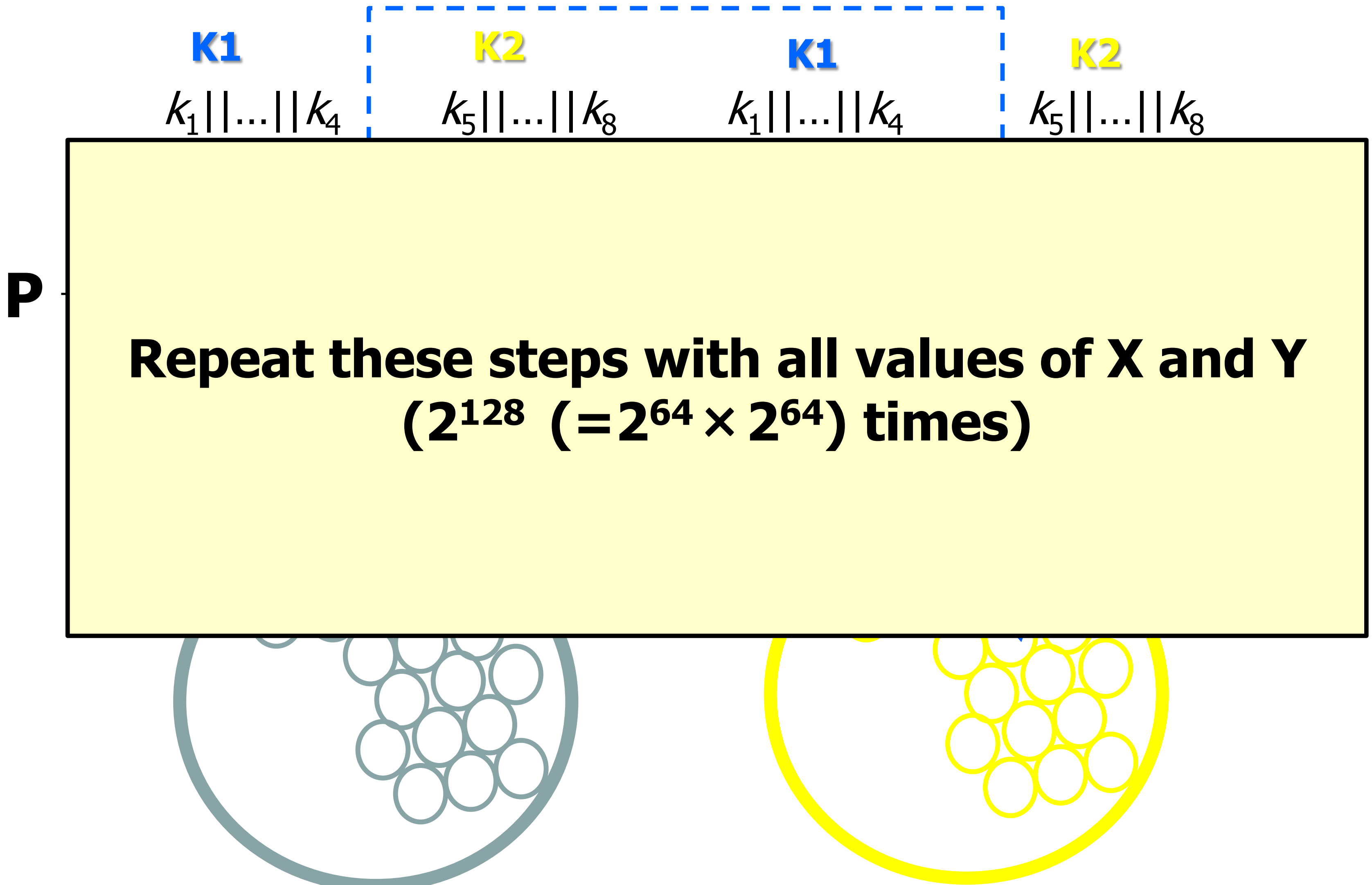
- Mount the MITM approach in only intermediate 8 round.



The size of independent set is 2^{64} ($> 2^{32}$)

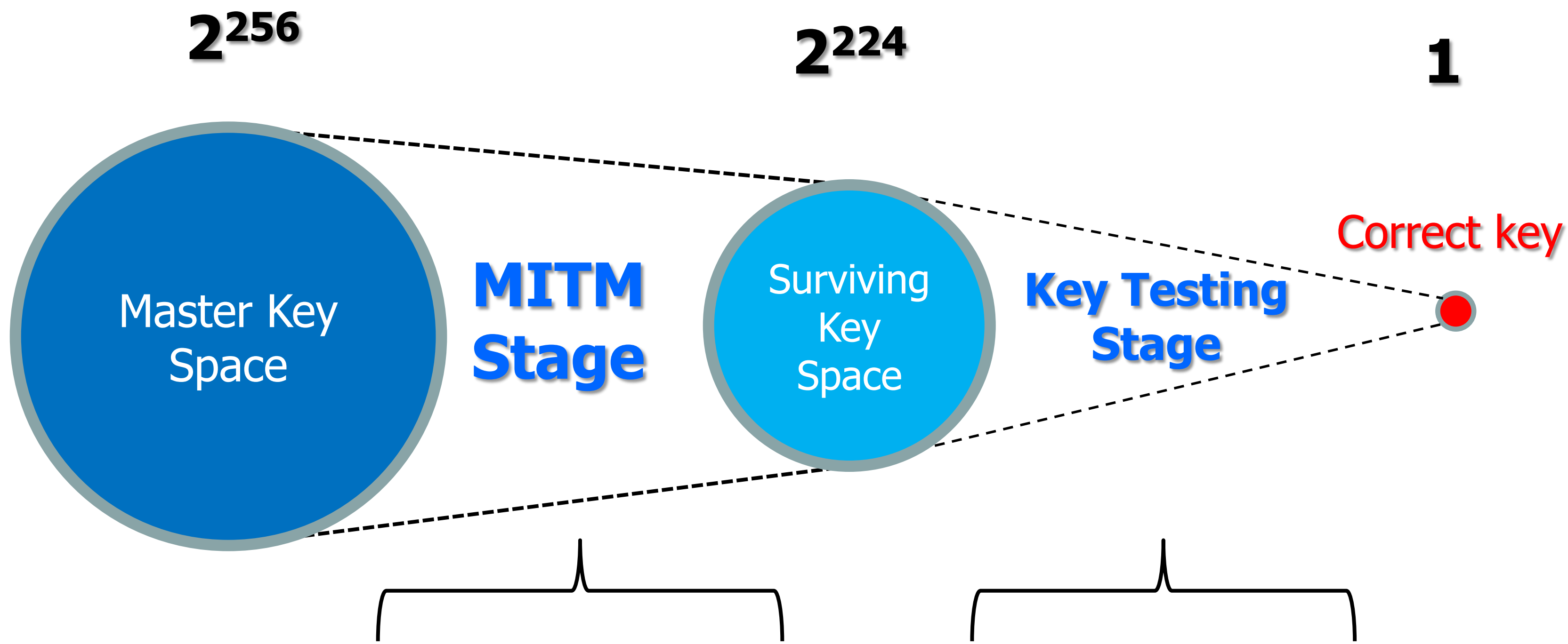
Effective MITM approach

- Mount the MITM approach in only intermediate 8 round.



Repeat these steps with all values of X and Y
(2^{128} ($=2^{64}\times 2^{64}$) times)

Evaluation



Complexity $= 2^{128}(2^{64}+2^{64}) + (2^{256-32}+2^{256-64}+\dots) = 2^{225}$

Data $= \max(2^{32}, 8) = 2^{32}$

It is faster than brute force attack (2^{256})

Result

Key Setting	Type of Attack	Round	Complexity	Data	Paper
Single Key	Differential	13	-	2^{51} (CP)	[28]
	Slide	24	2^{63}	$2^{64} - 2^{18}$ (KP)	[2]
	Slide	30	2^{254}	$2^{64} - 2^{18}$ (KP)	[2]
	Reflection	30	2^{224}	2^{32} (KP)	[17]
	Reflection-MITM	**32 (full)**	**2^{225}**	**2^{32} (KP)**	**Ours**

Key Setting	Type of Attack	Round	Complexity	Data	Paper
Single Key (Weak key)	Slide (2^{128} weak keys)	32 (full)	2^{63}	2^{63} (ACP)	[2]
	Reflectction (2^{224} weak keys)	32 (full)	2^{192}	2^{32} (CP)	[17]

Key Setting	Type of Attack	Round	Complexity	Data	Paper
Related Key	Differential	21	Not given	2^{56} (CP)	[28]
	Differential	32 (full)	2^{224}	2^{35} (CP)	[19]
	Boomerang	32 (full)	2^{248}	$2^{7.5}$ (CP)	[15]

A first single-key attack on the full GOST block cipher.
(work for all key classes)

Conclusion

- New attack framework "R-MITM attack"
  - Utilize fixed points to remove some rounds.
- Applied "R-MITM" to GOST block cipher
  - As a result, succeeded in constructing **first single key recovery attack.**
- Future Works and Remarks
  - Applied it to other block ciphers.
  - Other property may be used as skip technique instead of fixed points .

Thank You For Your Attention